

```

1 /*
2 <span id="gsh" data-title="GShell" data-author="sato@its-more.jp">
3 <meta charset="UTF-8">
4 <meta name="viewport" content="width=device-width, initial-scale=1.0">
5 <link rel="icon" id="GshFaviconURL" href=""><!-- place holder -->
6 <span id="GshVersion" hidden="">gsh--0.7.0--2020-10-16--SatoxITS</span>
7 <title id="GshTitle">GShell-0.7.0 by SatoxITS</title>
8
9 <div id="GshHeading">
10 <div id="GshTopbar" class="MetaWindow"></div>
11 <div id="GshPerfMon"></div>
12 <div id="GshSidebar" class="SbSidebar" style=""><div id="GshIndexer" style=""></div></div>
13 </div>
14 <div id="GshMain">
15 <div id="GshBanner" height="100px" onclick="shiftBG();">
16 <div align="right"><note><a href="http://archive.gshell.org">GShell</a> version 0.7.0 // 2020-10-16 // SatoxITS</note></div>
17 </div>
18 */
19
20 <!-- ----- Work { ----- -->
21 <span id="Topbar_WorkCodeSpan">
22 /*
23 <details><summary>Topbar</summary>
24 <!-- ----- Topbar // 2020-1008 SatoxITS { -->
25 <h2>Topbar</h2>
26 <input id="Topbar_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
27 <input id="Topbar_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
28 <input id="Topbar_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
29 <span id="Topbar_WorkCodeView"></span>
30 </details>
31
32 <style>
33 #GshHeading {
34   display:inline;
35   overflow:visible;
36 }
37 .ConfigIcon {
38   position:absolute;
39   top:-6px;
40   left:92%;
41   width:32px;
42   height:32px;
43 }
44 .MetaWindow {
45   z-index:1000;
46   position:relative;
47   display:block;
48   overflow:visible !important;
49   width:99.9%;
50   height:22px;
51   top:-22px;
52   border:1px solid #22a;
53   margin:0px;
54   left:0.0%;
55   line-height:1.0;
56   font-family:Georgia;
57   color:#fff;
58   font-size:12pt;
59   text-align:center;
60   vertical-align:middle;
61   padding:4px;
62   xxbackground-color:rgba(0,8,170,0.8);
63   background-color:#3a4861;xxx-PBlue;
64   vertical-align:middle;
65 }
66 .MetaWindow:hover {
67   color:#000;
68   border:1px solid #22a;
69   background-color:rgba(255,255,255,1.0);
70 }
71 #GshBanner {
72   overflow:visible;
73   display:block;
74   width:100%;
75   height:100px;
76   left:inherit !important;
77 }
78 </style>
79 <script>
80 function Topbar_openWorkCodeView(){
81   function Topbar_showWorkCode(){
82     showHtmlCode(Topbar_WorkCodeView,Topbar_WorkCodeSpan);
83   }
84   Topbar_WorkCodeViewOpen.addEventListener('click',Topbar_showWorkCode);
85 }
86 Topbar_openWorkCodeView();
87 function ConfigClick(){
88   if( 0 <= AffView.style.zIndex ){
89     AffView.style.saved_zIndex = AffView.style.zIndex;
90     AffView.style.zIndex = -1000;
91     GshSidebar.style.zIndex = -1;
92     GshPerfMon.style.zIndex = -1;
93   }else{
94     //AffView.style.zIndex = AffView.style.saved_zIndex;
95     AffView.style.zIndex = 1;
96     GshSidebar.style.zIndex = 1;
97     GshPerfMon.style.zIndex = 1;
98     GMenu.style.zIndex = 100000000;
99   }
100   console.log('AffZidex='+AffView.style.zIndex);
101 }
102 function Gshell_initTopbar(){
103   GshTopbar.innerHTML = GshTitle.innerHTML;
104   <!--<img id="ConfigIcon" class="ConfigIcon">
105   if( true ){
106     cfigi = document.createElement('img');
107     cfigi.id = 'ConfigIcon';
108     cfigi.setAttribute('class','ConfigIcon');
109     GshTopbar.appendChild(cfigi);
110     cfigi.src = ConfigICON_DATA;
111
112     //cfigi.style.zIndex = 100000000000;
113     //cfigi.addEventListener('click',ConfigClick);
114     GshTopbar.addEventListener('click',ConfigClick);
115   }
116 </script>
117 <!-- Topbar_WorkCodeSpan } -->
118 */ </span>
119 <!-- ----- Work } ----- -->
120 <!-- ----- Work { ----- -->
121 <span id="Indexer_WorkCodeSpan">

```

```

125 /*
126 <details><summary>Indexer</summary>
127 <!-- ----- Indexer // 2020-1007 SatoxITS { -->
128 <h2>Indexer</h2>
129 <input id="Indexer_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
130 <input id="Indexer_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
131 <input id="Indexer_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
132 <span id="Indexer_WorkCodeView"></span>
133 </details>
134 <style id="SidebarIndex">
135 #gsh {
136     display:block;
137     xxxoverflow:scroll !important;
138 }
139 #GshMain {
140     z-index:1;
141     position:relative;
142     display:block;
143     width:80% !important;
144     left:19.5% !important;
145 }
146 #GshSidebar {
147     z-index:0;
148     position:relative !important;
149     overflow:auto;
150     resize:both !important;
151     xxxoverflow-y:hidden !important;
152     xxxheight:100px !important;
153     xxxdisplay:inline !important;
154     left:0px;
155     top:0px;
156     width:19.5%;
157     min-width:80px;
158     xxxheight:100% !important;
159     height:0px;
160     color:#f00;
161     xxxbackground-color:rgba(64,64,64,0.5);
162     xxxbackground-color:#DFE3EB;xxx-PBlue;
163     background-color:#eeeeee;xxx-PBlue;
164 }
165 #GshPerfMon {
166     position:relative;
167     display:block;
168     overflow:visible;
169     z-index:0 !important;
170     xxheight:12pt;
171     font-family:monospace, Courier New !important;
172     font-size:9pt !important;
173     color:#f84;
174     top:-20px;
175 }
176 #GshPerfMon:hover {
177     z-index:3 !important;
178 }
179 #GshSidebar:hover {
180     z-index:2;
181     overflow-x:visible !important;
182     background-color:rgba(255,255,255,0.7);
183     width:50%;
184 }
185 #GshIndexer {
186     z-index:0;
187     position:relative;
188     resize:both !important;
189     height:100%;
190     left:0px;
191     top:0px;
192     scroll-behavior: overflow !important;
193     padding-left:4pt;
194     font-size:0.5em;
195     white-space:nowrap;
196     xxx-background-color:rgba(64,160,64,0.6) !important;
197     color:#794c6;xxx-PBlue;
198     xxxbackground-color:#DFE3EB;xxx-PBlue;
199     background-color:#eeeeee;xxx-PBlue;
200 }
201 #GshIndexer:hover {
202     z-index:10000000;
203     overflow-x:visible !important;
204     color:#000000 !important;xxx-PBlue;
205     xxxbackground-color:#FFFFFF;xxx-PBlue;
206     background-color:rgba(255,255,255,0.7);
207     padding-right:0px;
208     width:80%;
209 }
210 #GshIndexer:select {
211     color:#000000 !important;xxx-PBlue;
212     background-color:#FFFFFF;xxx-PBlue;
213 }
214 .IndexLine {
215     font-size:8pt !important;
216     font-family:Georgia;
217     display:block;
218     xxx-color:#fff;
219     xxx-color:#efff5;xxx-PBlue;
220     xxx-color:#41516d;xxx-PBlue;
221     xxx-color:#7794c6;xxx-PBlue;
222     padding-right:4pt;
223 }
224 .IndexLine:hover {
225     font-size:10pt !important;
226     xxx-color:#228;
227     xxx-background-color:#fff;
228     xxxcolor:#fff;xxx-PBlue;
229     color:#516487;xxx-PBlue;
230     background-color:rgba(220,220,255,1.0);xxx-PBlue;
231     xxxtext-shadow:1px 1px #3f3;
232     text-shadow:1px 1px #eee;
233     xxxbackground-color:#516487;xxx-PBlue;
234     xxxtext-decoration:underline !important;
235 }
236 </style>
237
238 <script id="Indexer_WorkScript">
239 function Indexer_openWorkCodeView(){
240     function Indexer_showWorkCode(){
241         showHtmlCode(Indexer_WorkCodeView,Indexer_WorkCodeSpan);
242     }
243     Indexer_WorkCodeViewOpen.addEventListener('click',Indexer_showWorkCode);
244 }
245 //Indexer_WorkCodeViewOpen.addEventListener('click',Indexer_openWorkCodeView);
246 Indexer_openWorkCodeView();
247
248 var startPerfDate = new Date();

```

```

249 var prevPerfDate = startPerfDate;
250 function ShowResourceUsage(){
251     d = new Date();
252     perf = '';
253     perf += '<'+font color="gray">UA: ' + window.navigator.userAgent + '<'+font><br>\n';
254     perf += DateShort0(startPerfDate) + '<br>\n';
255     perf += DateShort() + '<br>\n';
256     elps = d.getTime() - startPerfDate.getTime();
257     itvl = d.getTime() - prevPerfDate.getTime();
258     perf += 'Elapsed: ' + elps/1000 + ' s<br>\n';
259     perf += 'Skew: ' + (itvl-1000) + ' ms<br>\n';
260     prevPerfDate = d;
261
262     if( performance.memory !== undefined ){
263         m0 = performance.memory;
264         mu0 = (m0.usedJSHeapSize / 1000000.0); //.toFixed(6);
265         perf += 'Memory: '+mu0+' MB<br>\n';
266     }
267     perf += '<br>\n';
268
269     //GshSidebar.innerHTML = perf;
270     GshPerfMon.innerHTML = perf;
271     //GshIndexer.innerHTML = 'Memory: '+mu0+' MB';
272     //console.log('-- PerfMon heap: '+mu0+'/'+m0.totalHeapSize+'/'+m0.jsHeapSizeLimit);
273     if( true ){
274         GshSidebar.style.zIndex = 1000;
275         GshIndexer.style.zIndex = 0;
276         GshPerfMon.style.zIndex = 1;
277         //GshSidebar.appendChild(GshPerfMon);
278         if( document.getElementById('primary') == null ){ // not in WordPress
279             // GshPerfMon.style.position = 'absolute';
280         }
281         GshPerfMon.style.display = 'block';
282         GshPerfMon.style.marginLeft = '4px';
283         //GshPerfMon.style.top = '45px';
284         GshPerfMon.style.position = 'relative';
285         //GshPerfMon.style.position = 'absolute';
286         //topy = GshTopbar.getBoundingClientRect().top;
287         //topy = parseInt(topy) + 40;
288         //GshPerfMon.style.top = topy + 'px';
289         GshPerfMon.style.left = '0px';
290
291         GshMain.style.top = -GshPerfMon.getBoundingClientRect().height;
292     }
293 }
294 function ResetPerfMon(){
295     GshPerfMon.removeAttribute('style');
296     GshSidebar.removeAttribute('style');
297 }
298
299 var iserno = 0;
300 var GeneratedId = 0;
301 function generateIndex(ni,e,chv,nch,ht){
302     // https://developer.mozilla.org/en-US/docs/Web/API/Element
303     c = '';
304     if( e.classList != null ){
305         c = e.classList.value;
306     }
307     //console.log('-- <'+e.nodeName+'> #' + e.id + ' .'+c+' '+e.attributes);
308     if( e.nodeName == '#text' ){ return ''; }
309     if( e.nodeName == '#comment' ){ return ''; }
310     if( e.nodeName == 'H2' || e.nodeName == 'H3' ){
311         id = e.innerHTML;
312         GeneratedId += 1;
313         eid = 'GeneratedId-'+GeneratedId;
314         e.id = eid;
315     }else
316     if( e.nodeName == 'SUMMARY' ){
317         id = e.innerHTML;
318         GeneratedId += 1;
319         eid = 'GeneratedId-'+GeneratedId;
320         e.id = eid;
321     }else
322     if( and(e.nodeName == 'DIV',c=='xxxxxxxxxxxxxxxxentry-content') ){
323         console.log('-- DIV entry-content begin');
324         id = e.innerHTML;
325         GeneratedId += 1;
326         eid = 'GeneratedId-'+GeneratedId;
327         e.id = eid;
328         console.log('-- DIV entry-content end hash-child=',e.hasChildNodes());
329     }else
330     if( e.id == '' || e.id == 'undefined' ){
331         return '';
332     }else{
333         id = '#' + e.id;
334         eid = e.id;
335     }
336     iserno += 1;
337     ht = '<'+div id="GeneratedEref '+iserno+' " class="IndexLine" href="'+eid+'">'
338     + iserno+' '+ni+' '+e.nodeName + ' ' + id;
339     if( e.id == '' || e.id == 'undefined' ){ return ht + '<'+div>'; }
340     if( !e.hasChildNodes() ){ return ht + '<'+div>'; }
341     chv = e.childNodes;
342     nch = e.childNodes.length;
343     if( chv != null ){ nch = chv.length; }
344     ht += ' ('+nch+')' + '<'+div>';
345     for( let i = 0; i < chv.length; i++ ){
346         sec = ni+'.'+i;
347         if( ni == '' ){ sec = i; }
348         ht += generateIndex(sec,chv[i],null,0);
349     }
350     return ht;
351 }
352 function onClickIndex(e){
353     tid = e.target.id;
354     tge = document.getElementById(tid);
355     eid = tge.getAttribute('href');
356     rx = tge.getBoundingClientRect().left.toFixed(0)
357     ry = tge.getBoundingClientRect().top.toFixed(0)
358     if( false ){
359         alert('index clicked mouse(x='+e.x+', y='+e.y+')'
360             + '\ntid=#'+ tid + ' rx='+rx + ',ry='+ry
361             + '\neid= ' + eid + '\n'
362             + '\nhtml=' + tge.outerHTML);
363     }
364     ee = document.getElementById(eid);
365     sx = 'NaN';
366     sy = ee.getBoundingClientRect().top;
367     console.log('sx='+sx+',sy='+sy);
368     ee.scrollIntoView();
369     window.scrollTo(sx,sy)
370     //window.scrollTo({left:'Non',top:sy,behavior:'smooth'});
371 }
372 function Indexer_afterLoaded(){

```

```

373 sideindex = document.getElementById('GshIndexer');
374 ht = '<'+h3>G-Index<+'/h3>';
375 ht += generateIndex(document.getElementById('gsh'),null,0,'');
376 if( (pri = document.getElementById('primary')) != null ){
377     ht += generateIndex("",pri,null,0,'');
378 }
379 ht += '<'+br>';
380 ht += '<'+br>';
381 ht += '<'+br>';
382 ht += '<'+br>';
383 sideindex.innerHTML = ht;
384 sideindex.addEventListener('click',onClickIndex);
385
386 if( (pri = document.getElementById('primary')) != null ){
387     console.log('-- Seems in WordPress');
388     pri.style.zIndex = 2000;
389
390     GshSidebar.style.setProperty('position','relative','important');
391     GshSidebar.style.top = '-1400px';
392     //GshSidebar.style.setProperty('position','absolute','important');
393     //GshSidebar.style.top = '0px';
394
395     GshSidebar.style.setProperty('width','200px','important');
396     GshSidebar.style.setProperty('overflow','scroll','important');
397     GshSidebar.style.resize = 'both';
398
399     GshSidebar.style.left = '-100px';
400     GshIndexer.style.left = '100px';
401     GshIndexer.style.height = '1400px';
402     gsh.appendChild(GshSidebar); // change parent
403 }else{
404     console.log('-- Seems not in WordPress');
405     GshSidebar.style.setProperty('position','fixed','important');
406 }
407 }
408 //document.addEventListener('load',Indexer_afterLoaded);
409
410 DestroyIndexBar = function(){
411     sideindex = document.getElementById('GshIndexer');
412     sideindex.innerHTML = "";
413     sideindex.style = "";
414 }
415 </script>
416
417 <!-- Indexer_WorkCodeSpan -->
418 *//</span>
419 //<!-- ----- Work } ----- -->
420
421 /*
422 <h2>GShell // a General purpose Shell built on the top of Golang</h2>
423 <p>
424 <note>
425 It is a shell for myself, by myself, of myself. --SatoxITS(^-^ )
426 <a href="gsh-0.6.2.go.html">prev.</a>
427 </note>
428 </p>
429 <div id="GJFactory_x"></div>
430
431 <div>
432 <span id="GshMenu">
433 <span class="GshMenu" id="GshMenuEdit" onclick="html_edit();">Edit</span>
434 <span class="GshMenu" id="GshMenuSave" onclick="html_save();">Save</span>
435 <span class="GshMenu" id="GshMenuLoad" onclick="html_load();">Load</span>
436 <span class="GshMenu" id="GshMenuVers" onclick="html_ver();">Vers</span>
437 <span id="gsh-WinId" onclick="win_jump('0.1');">0</span>
438 <span class="GshMenu" id="gsh-menu-exit" onclick="html_close();"></span>
439 <span class="GshMenu" id="gsh-menu-fork" onclick="html_fork();">Fork</span>
440 <span class="GshMenu" id="GshMenuStop" onclick="html_stop(this,true);">Stop</span>
441 <span class="GshMenu" id="GshMenuFold" onclick="html_fold(this);">Unfold</span>
442 <span class="GshMenu" id="gsh-menu-cksum" onclick="html_digest();">Digest</span>
443 <span class="GshMenu" id="GshMenuSign" onclick="html_sign(this);" style="">Source</span>
444 <!-- | <span id="gsh-menu-pure" onclick="html_pure(this);">Pure</span> -->
445 </span>
446 </div>
447 *//
448 /*
449
450 <details id="GshStatement" class="gsh-document"><summary>Statement</summary>
451 <h3>Fun to create a shell</h3>
452 <p>For a programmer, it must be far easy and fun to create his own simple shell
453 rightly fitting to his favor and necessities, than learning existing shells with
454 complex full features that he never use.
455 I, as one of programmers, am writing this tiny shell for my own real needs,
456 totally from scratch, with fun.
457 </p><p>
458 For a programmer, it is fun to learn new computer languages. For long years before
459 writing this software, I had been specialized to C and early HTML2 :-).
460 Now writing this software, I'm learning Go language, HTML5, JavaScript and CSS
461 on demand as a novice of these, with fun.
462 </p><p>
463 This single file "gsh.go", that is executable by Go, contains all of the code written
464 in Go. Also it can be displayed as "gsh.go.html" by browsers. It is a standalone
465 HTML file that works as the viewer of the code of itself, and as the "home page" of
466 this software.
467 </p><p>
468 Because this HTML file is a Go program, you may run it as a real shell program
469 on your computer.
470 But you must be aware that this program is written under situation like above.
471 Needless to say, there is no warranty for this program in any means.
472 </p>
473 <address>Aug 2020, SatoxITS (sato@its-more.jp)</address>
474 </details>
475 *//
476 /*
477 <details id="GshFeatures" class="gsh-document"><summary>Features</summary><p>
478 </p>
479 <h3>Cross-browser communication</h3>
480 <p>
481 ... to be written ...
482 </p>
483 <h3>Vi compatible command line editor</h3>
484 <p>
485 The command line of GShell can be edited with commands compatible with
486 <a href="https://www.washington.edu/computing/unix/vi.html"><b>vi</b></a>.
487 As in vi, you can enter <b>command mode</b> by <b>ESC</b> key,
488 then move around in the history by <b>code>j k / ? n N</code>,
489 or within the current line by <b>code>l h f w b O $ %</code> or so.
490 </p>
491 </details>
492 *//
493 /*
494 <details id="gsh-gindex">
495 <summary>Index</summary><div class="gsh-src">

```

```

497 Documents
498 <span class="gsh-link" onclick="jumpto_JavaScriptView();">Command summary</span>
499 Go lang part<span class="gsh-src" onclick="document.getElementById('gsh-gocode').open=true;">
500 Package structures
501 <a href="#import">import</a>
502 <a href="#struct">struct</a>
503 Main functions
504 <a href="#comexpansion">str-expansion</a> // macro processor
505 <a href="#finder">finder</a> // builtin find + du
506 <a href="#grep">grep</a> // builtin grep + wc + cksum + ...
507 <a href="#plugin">plugin</a> // plugin commands
508 <a href="#ex-commands">system</a> // external commands
509 <a href="#builtin">builtin</a> // builtin commands
510 <a href="#network">network</a> // socket handler
511 <a href="#remote-sh">remote-sh</a> // remote shell
512 <a href="#redirect">redirect</a> // StdIn/Out redirection
513 <a href="#history">history</a> // command history
514 <a href="#rusage">rusage</a> // resource usage
515 <a href="#encode">encode</a> // encode / decode
516 <a href="#IME">IME</a> // command line IME
517 <a href="#getline">getline</a> // line editor
518 <a href="#scanf">scanf</a> // string decomposer
519 <a href="#interpreter">interpreter</a> // command interpreter
520 <a href="#main">main</a>
521 </span>
522 JavaScript part
523 <a href="#script-src-view" class="gsh-link" onclick="jumpto_JavaScriptView();">Source</a>
524 <a href="#gsh-data-frame" class="gsh-link" onclick="jumpto_DataView();">Builtin data</a>
525 CSS part
526 <a href="#style-src-view" class="gsh-link" onclick="jumpto_StyleView();">Source</a>
527 References
528 <a href="#" class="gsh-link" onclick="jumpto_WholeView();">Internal</a>
529 <a href="#gsh-reference" class="gsh-link" onclick="jumpto_ReferenceView();">External</a>
530 Whole parts
531 <a href="#whole-src-view" class="gsh-link" onclick="jumpto_WholeView();">Source</a>
532 <a href="#whole-src-view" class="gsh-link" onclick="jumpto_WholeView();">Download</a>
533 <a href="#whole-src-view" class="gsh-link" onclick="jumpto_WholeView();">Dump</a>
534
535 </div>
536 </details>
537 */
538 <<details id="gsh-gocode">
539 <<summary>Go Source</summary><div class="gsh-src" onclick="document.getElementById('gsh-gocode').open=false;">
540 // gsh - Go lang based Shell
541 // (c) 2020 ITS more Co., Ltd.
542 // 2020-0807 created by SatoxITS (sato@its-more.jp)
543
544 package main // gsh main
545
546 // <a name="import">Imported packages</a> // <a href="https://golang.org/pkg/">Packages</a>
547 import (
548     "fmt" // <a href="https://golang.org/pkg/fmt/">fmt</a>
549     "strings" // <a href="https://golang.org/pkg/strings/">strings</a>
550     "strconv" // <a href="https://golang.org/pkg/strconv/">strconv</a>
551     "sort" // <a href="https://golang.org/pkg/sort/">sort</a>
552     "time" // <a href="https://golang.org/pkg/time/">time</a>
553     "bufio" // <a href="https://golang.org/pkg/bufio/">bufio</a>
554     "io/ioutil" // <a href="https://golang.org/pkg/io/ioutil/">ioutil</a>
555     "os" // <a href="https://golang.org/pkg/os/">os</a>
556     "syscall" // <a href="https://golang.org/pkg/syscall/">syscall</a>
557     "plugin" // <a href="https://golang.org/pkg/plugin/">plugin</a>
558     "net" // <a href="https://golang.org/pkg/net/">net</a>
559     "net/http" // <a href="https://golang.org/pkg/net/http/">http</a>
560     "html" // <a href="https://golang.org/pkg/html/">html</a>
561     "path/filepath" // <a href="https://golang.org/pkg/path/filepath/">filepath</a>
562     "go/types" // <a href="https://golang.org/pkg/go/types/">types</a>
563     "go/token" // <a href="https://golang.org/pkg/go/token/">token</a>
564     "encoding/base64" // <a href="https://golang.org/pkg/encoding/base64/">base64</a>
565     "unicode/utf8" // <a href="https://golang.org/pkg/unicode/utf8/">utf8</a>
566     // "gshdata" // gshell's logo and source code
567     "hash/crc32" // <a href="https://golang.org/pkg/unicode/hash/crc32/">crc32</a>
568     "golang.org/x/net/websocket"
569 )
570
571 // // 2020-0906 added,
572 // // <a href="https://golang.org/cmd/cgo/">CGO</a>
573 // #include "poll.h" // <poll.h> // </poll.h> to be closed as HTML tag :-p
574 // typedef struct { struct pollfd fdv[8]; } pollFdv;
575 // int pollx(pollFdv *fdv, int nfd, int timeout){
576 //     return poll(fdv->fdv,nfds,timeout);
577 // }
578 import "C"
579
580 // // 2020-0906 added,
581 func CPollIn1(fp*os.File, timeoutUs int)(ready uintptr){
582     var fdv = C.pollFdv{}
583     var nfd = 1
584     var timeout = timeoutUs/1000
585
586     fdv.fdv[0].fd = C.int(fp.Fd())
587     fdv.fdv[0].events = C.POLLIN
588     if (0 < EventRecvFd) {
589         fdv.fdv[1].fd = C.int(EventRecvFd)
590         fdv.fdv[1].events = C.POLLIN
591         nfd += 1
592     }
593     r := C.pollx(&fdv,C.int(nfd),C.int(timeout))
594     if (r <= 0){
595         return 0
596     }
597     if (int(fdv.fdv[1].revents) & int(C.POLLIN)) != 0 {
598         //fprintf(stderr,"--De-- got Event\n");
599         return uintptr(EventFdOffset + fdv.fdv[1].fd)
600     }
601     if (int(fdv.fdv[0].revents) & int(C.POLLIN)) != 0 {
602         return uintptr(NormalFdOffset + fdv.fdv[0].fd)
603     }
604     return 0
605 }
606
607 const (
608     NAME = "gsh"
609     VERSION = "0.7.0"
610     DATE = "2020-10-16"
611     AUTHOR = "SatoxITS(^-^)"
612 )
613 var (
614     GSH_HOME = ".gsh" // under home directory
615     GSH_PORT = 9999
616     MaxStreamSize = int64(128*1024*1024*1024) // 128GiB is too large?
617     PROMPT = ">"
618     LINESIZE = (8*1024)
619     PATHSEP = ";" // should be ";" in Windows
620     DIRSEP = "/" // canbe \ in Windows

```

```

621 )
622 // -xX logging control
623 // --A-- all
624 // --I-- info.
625 // --D-- debug
626 // --T-- time and resource usage
627 // --W-- warning
628 // --E-- error
629 // --F-- fatal error
630 // --Xn- network
631 //
632
633 // <a name="struct">Structures</a>
634 type GCommandHistory struct {
635     StartAt    time.Time // command line execution started at
636     EndAt      time.Time // command line execution ended at
637     ResCode    int       // exit code of (external command)
638     CmdError   error     // error string
639     OutData    *os.File // output of the command
640     FoundFile  []string // output - result of ufind
641     Rusagev    [2]syscall.Rusage // Resource consumption, CPU time or so
642     CmdId      int       // maybe with identified with arguments or impact
643     // redirection commands should not be the CmdId
644     WorkDir    string   // working directory at start
645     WorkDirX   int     // index in ChdirHistory
646     CmdLine    string   // command line
647 }
648 type GChdirHistory struct {
649     Dir        string
650     MovedAt   time.Time
651     CmdIndex   int
652 }
653 type CmdMode struct {
654     BackGround bool
655 }
656 type Event struct {
657     when        time.Time
658     event       int
659     evarg       int64
660     CmdIndex    int
661 }
662 var CmdIndex int
663 var Events []Event
664 type PluginInfo struct {
665     Spec        *plugin.Plugin
666     Addr        plugin.Symbol
667     Name        string // maybe relative
668     Path        string // this is in Plugin but hidden
669 }
670 type GServer struct {
671     host        string
672     port        string
673 }
674
675 // <a href="https://tools.ietf.org/html/rfc3230">Digest</a>
676 const ( // SumType
677     SUM_ITEMS = 0x000001 // items count
678     SUM_SIZE  = 0x000002 // data length (simply added)
679     SUM_SIZEHASH = 0x000004 // data length (hashed sequence)
680     SUM_DATEHASH = 0x000008 // date of data (hashed sequence)
681     // also envelope attributes like time stamp can be a part of digest
682     // hashed value of sizes or mod-date of files will be useful to detect changes
683
684     SUM_WORDS = 0x000010 // word count is a kind of digest
685     SUM_LINES = 0x000020 // line count is a kind of digest
686     SUM_SUM64 = 0x000040 // simple add of bytes, useful for human too
687
688     SUM_SUM32_BITS = 0x000100 // the number of true bits
689     SUM_SUM32_2BYTE = 0x000200 // 16bits words
690     SUM_SUM32_4BYTE = 0x000400 // 32bits words
691     SUM_SUM32_8BYTE = 0x000800 // 64bits words
692
693     SUM_SUM16_BSD = 0x001000 // UNIXsum -sum -bsd
694     SUM_SUM16_SYSV = 0x002000 // UNIXsum -sum -sysv
695     SUM_UNIXFILE = 0x004000
696     SUM_CRCIEEE = 0x008000
697 )
698 type CheckSum struct {
699     Files    int64 // the number of files (or data)
700     Size     int64 // content size
701     Words    int64 // word count
702     Lines    int64 // line count
703     SumType  int
704     Sum64    uint64
705     Crc32Table crc32.Table
706     Crc32Val  uint32
707     Sum16     int
708     Ctime     time.Time
709     Atime     time.Time
710     Mtime     time.Time
711     Start     time.Time
712     Done      time.Time
713     RusgAtStart [2]syscall.Rusage
714     RusgAtEnd   [2]syscall.Rusage
715 }
716 type ValueStack [][]string
717 type GshContext struct {
718     StartDir    string // the current directory at the start
719     GetLine     string // gsh-getline command as a input line editor
720     ChdirHistory []GChdirHistory // the 1st entry is wd at the start
721     gshPA       syscall.ProcAttr
722     CommandHistory []GCommandHistory
723     CmdCurrent   GCommandHistory
724     BackGround   bool
725     BackGroundJobs []int
726     LastRusage    syscall.Rusage
727     GshHomeDir    string
728     TerminalId    int
729     CmdTrace      bool // should be [map]
730     CmdTime       bool // should be [map]
731     PluginFuncs  []PluginInfo
732     iValues       []string
733     iDelimiter    string // field sepearater of print out
734     iFormat       string // default print format (of integer)
735     iValStack     ValueStack
736     LastServer    GServer
737     RSERV         string // [gsh://]host[:port]
738     RWD           string // remote (target, there) working directory
739     lastCheckSum CheckSum
740 }
741
742 func nsleep(ns time.Duration){
743     time.Sleep(ns)
744 }

```

```

745 func usleep(ns time.Duration){
746     nsleep(ns*1000)
747 }
748 func msleep(ns time.Duration){
749     nsleep(ns*1000000)
750 }
751 func sleep(ns time.Duration){
752     nsleep(ns*1000000000)
753 }
754
755 func strBegins(str, pat string)(bool){
756     if len(pat) <= len(str){
757         yes := str[0:len(pat)] == pat
758         //fmt.Printf("--D-- strBegins(%v,%v)=%v\n",str,pat, yes)
759         return yes
760     }
761     //fmt.Printf("--D-- strBegins(%v,%v)=%v\n",str,pat,false)
762     return false
763 }
764 func isin(what string, list []string) bool {
765     for _, v := range list {
766         if v == what {
767             return true
768         }
769     }
770     return false
771 }
772 func isinX(what string,list[]string)(int){
773     for i,v := range list {
774         if v == what {
775             return i
776         }
777     }
778     return -1
779 }
780
781 func env(opts []string) {
782     env := os.Environ()
783     if isin("-", opts){
784         sort.Slice(env, func(i,j int) bool {
785             return env[i] < env[j]
786         })
787     }
788     for _, v := range env {
789         fmt.Printf("%v\n",v)
790     }
791 }
792
793 // - rewriting should be context dependent
794 // - should postpone until the real point of evaluation
795 // - should rewrite only known notation of symobl
796 func scanInt(str string)(val int,leng int){
797     leng = -1
798     for i,ch := range str {
799         if '0' <= ch && ch <= '9' {
800             leng = i+1
801         }else{
802             break
803         }
804     }
805     if 0 < leng {
806         ival, _ := strconv.Atoi(str[0:leng])
807         return ival,leng
808     }else{
809         return 0,0
810     }
811 }
812 func substHistory(gshCtx *GshContext,str string,i int,rstr string)(leng int,rst string){
813     if len(str[i+1:]) == 0 {
814         return 0,rstr
815     }
816     hi := 0
817     histlen := len(gshCtx.CommandHistory)
818     if str[i+1] == '|' {
819         hi = histlen - 1
820         leng = 1
821     }else{
822         hi,leng = scanInt(str[i+1:])
823         if leng == 0 {
824             return 0,rstr
825         }
826         if hi < 0 {
827             hi = histlen + hi
828         }
829     }
830     if 0 <= hi && hi < histlen {
831         var ext byte
832         if 1 < len(str[i+leng:]) {
833             ext = str[i+leng:][1]
834         }
835         //fmt.Printf("--D-- %v(%c)\n",str[i+leng:],str[i+leng])
836         if ext == 'f' {
837             leng += 1
838             xlist := []string{}
839             list := gshCtx.CommandHistory[hi].FoundFile
840             for _,v := range list {
841                 //list[i] = escapeWhiteSP(v)
842                 xlist = append(xlist,escapeWhiteSP(v))
843             }
844             //rstr += strings.Join(list," ")
845             rstr += strings.Join(xlist," ")
846         }else
847         if ext == '@' || ext == 'd' {
848             // !N@ .. workdir at the start of the command
849             leng += 1
850             rstr += gshCtx.CommandHistory[hi].WorkDir
851         }else{
852             rstr += gshCtx.CommandHistory[hi].CmdLine
853         }
854     }else{
855         leng = 0
856     }
857     return leng,rstr
858 }
859 func escapeWhiteSP(str string)(string){
860     if len(str) == 0 {
861         return "\\z" // empty, to be ignored
862     }
863     rstr := ""
864     for _,ch := range str {
865         switch ch {
866             case '\\': rstr += "\\\\"
867             case ' ': rstr += "\\s"
868             case '\t': rstr += "\\t"

```

```

869         case 'r': rstr += "\\r"
870         case 'n': rstr += "\\n"
871         default: rstr += string(ch)
872     }
873 }
874 return rstr
875 }
876 func unescapeWhiteSP(str string)(string){ // strip original escapes
877     rstr := ""
878     for i := 0; i < len(str); i++ {
879         ch := str[i]
880         if ch == '\\' {
881             if i+1 < len(str) {
882                 switch str[i+1] {
883                     case 'z':
884                         continue;
885                 }
886             }
887         }
888         rstr += string(ch)
889     }
890     return rstr
891 }
892 func unescapeWhiteSPV(strv []string)([]string){ // strip original escapes
893     ustrv := []string{}
894     for _,v := range strv {
895         ustrv = append(ustrv,unescapeWhiteSP(v))
896     }
897     return ustrv
898 }
899
900 // <a name="comexpansion">str-expansion</a>
901 // - this should be a macro processor
902 func strsubst(gshCtx *GshContext,str string,histonly bool) string {
903     rbuff := []byte{}
904     if false {
905         //@@@ Unicode should be cared as a character
906         return str
907     }
908     //rstr := ""
909     inEsc := 0 // escape characer mode
910     for i := 0; i < len(str); i++ {
911         //fmt.Printf("--D--Subst %v:%v\n",i,str[i:])
912         ch := str[i]
913         if inEsc == 0 {
914             if ch == '|' {
915                 //leng,xrstr := substHistory(gshCtx,str,i,rstr)
916                 leng,rs := substHistory(gshCtx,str,i,"")
917                 if 0 < leng {
918                     //_,rs := substHistory(gshCtx,str,i,"")
919                     rbuff = append(rbuff,[]byte(rs)...)
920                     i += leng
921                     //rstr = xrstr
922                     continue
923                 }
924             }
925             switch ch {
926                 case '\\': inEsc = '\\'; continue
927                 //case '&': inEsc = '&'; continue
928                 case '$':
929             }
930         }
931         switch inEsc {
932             case '\\':
933                 switch ch {
934                     case '\\': ch = '\\'
935                     case 's': ch = ' '
936                     case 't': ch = '\t'
937                     case 'r': ch = '\r'
938                     case 'n': ch = '\n'
939                     case 'z': inEsc = 0; continue // empty, to be ignored
940                 }
941             case '&':
942                 inEsc = 0
943                 case '$':
944                     switch {
945                         case ch == '&': ch = '%'
946                         case ch == 'T':
947                             //rstr = rstr + time.Now().Format(time.Stamp)
948                             rs := time.Now().Format(time.Stamp)
949                             rbuff = append(rbuff,[]byte(rs)...)
950                             inEsc = 0
951                             continue;
952                         default:
953                             // postpone the interpretation
954                             //rstr = rstr + "&" + string(ch)
955                             rbuff = append(rbuff,ch)
956                             inEsc = 0
957                             continue;
958                     }
959             case 0
960         }
961         //rstr = rstr + string(ch)
962         rbuff = append(rbuff,ch)
963     }
964     //fmt.Printf("--D--subst(%s)(%s)\n",str,string(rbuff))
965     return string(rbuff)
966     //return rstr
967 }
968 func showFileInfo(path string, opts []string) {
969     if isin("-l",opts) || isin("-ls",opts) {
970         fi, err := os.Stat(path)
971         if err != nil {
972             fmt.Printf("----- ((%v))",err)
973         }else{
974             mod := fi.ModTime()
975             date := mod.Format(time.Stamp)
976             fmt.Printf("%v %v %s ",fi.Mode(),fi.Size(),date)
977         }
978     }
979     fmt.Printf("%s",path)
980     if isin("-sp",opts) {
981         fmt.Printf(" ")
982     }else
983     if ! isin("-n",opts) {
984         fmt.Printf("\n")
985     }
986 }
987 func userHomeDir()(string,bool){
988     /*
989     homedir,_ = os.UserHomeDir() // not implemented in older Golang
990     */
991     homedir,found := os.LookupEnv("HOME")
992     //fmt.Printf("--I-- HOME=%v(%v)\n",homedir,found)
993     if !found {

```



```

993     return "/tmp",found
994 }
995     return homedir,found
996 }
997
998 func toFullpath(path string) (fullpath string) {
999     if path[0] == '/' {
1000         return path
1001     }
1002     pathv := strings.Split(path,DIRSEP)
1003     switch {
1004     case pathv[0] == ".":
1005         pathv[0], _ = os.Getwd()
1006     case pathv[0] == "..": // all ones should be interpreted
1007         cwd, _ := os.Getwd()
1008         ppathv := strings.Split(cwd,DIRSEP)
1009         pathv[0] = strings.Join(ppathv,DIRSEP)
1010     case pathv[0] == "~":
1011         pathv[0],_ = userHomeDir()
1012     default:
1013         cwd, _ := os.Getwd()
1014         pathv[0] = cwd + DIRSEP + pathv[0]
1015     }
1016     return strings.Join(pathv,DIRSEP)
1017 }
1018
1019 func IsRegFile(path string)(bool){
1020     fi, err := os.Stat(path)
1021     if err == nil {
1022         fm := fi.Mode()
1023         return fm.IsRegular();
1024     }
1025     return false
1026 }
1027
1028 // <a name="encode">Encode / Decode</a>
1029 // <a href="https://golang.org/pkg/encoding/base64/#example_NewEncoder">Encoder</a>
1030 func (gshCtx *GshContext)Enc(argv[]string){
1031     file := os.Stdin
1032     buff := make([]byte,LINESIZE)
1033     li := 0
1034     encoder := base64.NewEncoder(base64.StdEncoding,os.Stdout)
1035     for li = 0; ; li++ {
1036         count, err := file.Read(buff)
1037         if count <= 0 {
1038             break
1039         }
1040         if err != nil {
1041             break
1042         }
1043         encoder.Write(buff[0:count])
1044     }
1045     encoder.Close()
1046 }
1047 func (gshCtx *GshContext)Dec(argv[]string){
1048     decoder := base64.NewDecoder(base64.StdEncoding,os.Stdin)
1049     li := 0
1050     buff := make([]byte,LINESIZE)
1051     for li = 0; ; li++ {
1052         count, err := decoder.Read(buff)
1053         if count <= 0 {
1054             break
1055         }
1056         if err != nil {
1057             break
1058         }
1059         os.Stdout.Write(buff[0:count])
1060     }
1061 }
1062 // lnspl [N] [-crlf][-C \\\]
1063 func (gshCtx *GshContext)SplitLine(argv[]string){
1064     strRep := isin("-str",argv) // ". "+
1065     reader := bufio.NewReaderSize(os.Stdin,64*1024)
1066     ni := 0
1067     toi := 0
1068     for ni = 0; ; ni++ {
1069         line, err := reader.ReadString('\n')
1070         if len(line) <= 0 {
1071             if err != nil {
1072                 fmt.Fprintf(os.Stderr,"--I-- lnspl %d to %d (%v)\n",ni,toi,err)
1073                 break
1074             }
1075         }
1076         off := 0
1077         ilen := len(line)
1078         remlen := len(line)
1079         if strRep { os.Stdout.Write([]byte("\n")) }
1080         for oi := 0; 0 < remlen; oi++ {
1081             olen := remlen
1082             addnl := false
1083             if 72 < olen {
1084                 olen = 72
1085                 addnl = true
1086             }
1087             fmt.Fprintf(os.Stderr,"--D-- write %d [%d.%d] %d %d/%d/%d\n",
1088                 toi,ni,oi,off,olen,remlen,ilen)
1089             toi += 1
1090             os.Stdout.Write([]byte(line[0:olen]))
1091             if addnl {
1092                 if strRep {
1093                     os.Stdout.Write([]byte("\n\n"))
1094                 }else{
1095                     //os.Stdout.Write([]byte("\r\n"))
1096                     os.Stdout.Write([]byte("\n"))
1097                     os.Stdout.Write([]byte("\n"))
1098                 }
1099             }
1100             line = line[olen:]
1101             off += olen
1102             remlen -= olen
1103         }
1104         if strRep { os.Stdout.Write([]byte("\n\n")) }
1105     }
1106     fmt.Fprintf(os.Stderr,"--I-- lnspl %d to %d\n",ni,toi)
1107 }
1108
1109 // CRC32 <a href="http://golang.jp/pkg/hash-crc32">crc32</a>
1110 // 1 0000 0100 1100 0001 0001 1101 1011 0111
1111 var CRC32UNIX uint32 = uint32(0x04c11db7) // Unix cksum
1112 var CRC32IEEE uint32 = uint32(0xedb88320)
1113 func byteCRC32add(crc uint32,str[]byte,len uint64)(uint32){
1114     var oi uint64
1115     for oi = 0; oi < len; oi++ {
1116         var oct = str[oi]

```

```

1117     for bi := 0; bi < 8; bi++ {
1118         //fprintf(stderr, "--CRC32 %d %X (%d.%d)\n", crc, oct, oi, bi)
1119         ovf1 := (crc & 0x80000000) != 0
1120         ovf2 := (oct & 0x80) != 0
1121         ovf := (ovf1 && !ovf2) || (!ovf1 && ovf2)
1122         oct <<= 1
1123         crc <<= 1
1124         if ovf { crc ^= CRC32UNIX }
1125     }
1126 }
1127 //fprintf(stderr, "--CRC32 return %d %d\n", crc, len)
1128 return crc;
1129 }
1130 func byteCRC32end(crc uint32, len uint64)(uint32){
1131     var slen = make([]byte, 4)
1132     var li = 0
1133     for li = 0; li < 4; {
1134         slen[li] = byte(len)
1135         li += 1
1136         len >>= 8
1137         if( len == 0 ){
1138             break
1139         }
1140     }
1141     crc = byteCRC32add(crc, slen, uint64(li))
1142     crc ^= 0xFFFFFFFF
1143     return crc
1144 }
1145 func strCRC32(str string, len uint64)(crc uint32){
1146     crc = byteCRC32add(0, []byte(str), len)
1147     crc = byteCRC32end(crc, len)
1148     //fprintf(stderr, "--CRC32 %d %d\n", crc, len)
1149     return crc
1150 }
1151 func CRC32Finish(crc uint32, table *crc32.Table, len uint64)(uint32){
1152     var slen = make([]byte, 4)
1153     var li = 0
1154     for li = 0; li < 4; {
1155         slen[li] = byte(len & 0xFF)
1156         li += 1
1157         len >>= 8
1158         if( len == 0 ){
1159             break
1160         }
1161     }
1162     crc = crc32.Update(crc, table, slen)
1163     crc ^= 0xFFFFFFFF
1164     return crc
1165 }
1166 }
1167 func (gsh*GshContext)xChecksum(path string, argv []string, sum*Checksum)(int64){
1168     if isin("-type/f", argv) && !IsRegFile(path){
1169         return 0
1170     }
1171     if isin("-type/d", argv) && IsRegFile(path){
1172         return 0
1173     }
1174     file, err := os.OpenFile(path, os.O_RDONLY, 0)
1175     if err != nil {
1176         fmt.Printf("--E-- cksum %v (%v)\n", path, err)
1177         return -1
1178     }
1179     defer file.Close()
1180     if gsh.CmdTrace { fmt.Printf("--I-- cksum %v %v\n", path, argv) }
1181 }
1182 bi := 0
1183 var buff = make([]byte, 32*1024)
1184 var total int64 = 0
1185 var initTime = time.Time{}
1186 if sum.Start == initTime {
1187     sum.Start = time.Now()
1188 }
1189 for bi = 0; ; bi++ {
1190     count, err := file.Read(buff)
1191     if count <= 0 || err != nil {
1192         break
1193     }
1194     if (sum.SumType & SUM_SUM64) != 0 {
1195         s := sum.Sum64
1196         for _, c := range buff[0:count] {
1197             s += uint64(c)
1198         }
1199         sum.Sum64 = s
1200     }
1201     if (sum.SumType & SUM_UNIXFILE) != 0 {
1202         sum.Crc32Val = byteCRC32add(sum.Crc32Val, buff, uint64(count))
1203     }
1204     if (sum.SumType & SUM_CRCIEEE) != 0 {
1205         sum.Crc32Val = crc32.Update(sum.Crc32Val, &sum.Crc32Table, buff[0:count])
1206     }
1207     // <a href="https://en.wikipedia.org/wiki/BSD_checksum">BSD checksum</a>
1208     if (sum.SumType & SUM_SUM16_BSD) != 0 {
1209         s := sum.Sum16
1210         for _, c := range buff[0:count] {
1211             s = (s >> 1) + ((s & 1) << 15)
1212             s += int(c)
1213             s &= 0xFFFF
1214             //fmt.Printf("BSDsum: %d[%d] %d\n", sum.Size+int64(i), i, s)
1215         }
1216         sum.Sum16 = s
1217     }
1218     if (sum.SumType & SUM_SUM16_SYSV) != 0 {
1219         for bj := 0; bj < count; bj++ {
1220             sum.Sum16 += int(buff[bj])
1221         }
1222     }
1223     total += int64(count)
1224 }
1225 sum.Done = time.Now()
1226 sum.Files += 1
1227 sum.Size += total
1228 if !isin("-s", argv) {
1229     fmt.Printf("%v ", total)
1230 }
1231 return 0
1232 }
1233 }
1234 // <a name="grep">grep</a>
1235 // "lines", "lin" or "lnp" for "(text) line processor" or "scanner"
1236 // a*,!ab,c, ... sequential combination of patterns
1237 // what "LINE" is should be definable
1238 // generic line-by-line processing
1239 // grep [-v]
1240 // cat -n -v

```

```

1241 // uniq [-c]
1242 // tail -f
1243 // sed s/x/y/ or awk
1244 // grep with line count like wc
1245 // rewrite contents if specified
1246 func (gsh*GshContext)xGrep(path string, rexpv[]string)(int){
1247     file, err := os.OpenFile(path, os.O_RDONLY, 0)
1248     if err != nil {
1249         fmt.Printf("--E-- grep %v (%v)\n", path, err)
1250         return -1
1251     }
1252     defer file.Close()
1253     if gsh.CmdTrace { fmt.Printf("--I-- grep %v %v\n", path, rexpv) }
1254     //reader := bufio.NewReaderSize(file, LINESIZE)
1255     reader := bufio.NewReaderSize(file, 80)
1256     li := 0
1257     found := 0
1258     for li = 0; ; li++ {
1259         line, err := reader.ReadString('\n')
1260         if len(line) <= 0 {
1261             break
1262         }
1263         if 150 < len(line) {
1264             // maybe binary
1265             break;
1266         }
1267         if err != nil {
1268             break
1269         }
1270         if 0 <= strings.Index(string(line), rexpv[0]) {
1271             found += 1
1272             fmt.Printf("%s:%d: %s", path, li, line)
1273         }
1274     }
1275     //fmt.Printf("total %d lines %s\n", li, path)
1276     //if( 0 < found ){ fmt.Printf("(found %d lines %s)\n", found, path); }
1277     return found
1278 }
1279
1280 // <a name="finder">Finder</a>
1281 // finding files with it name and contents
1282 // file names are Ored
1283 // show the content with %x fmt list
1284 // ls -R
1285 // tar command by adding output
1286 type fileSum struct {
1287     Err int64 // access error or so
1288     Size int64 // content size
1289     DupSize int64 // content size from hard links
1290     Blocks int64 // number of blocks (of 512 bytes)
1291     DupBlocks int64 // Blocks pointed from hard links
1292     HLinks int64 // hard links
1293     Words int64
1294     Lines int64
1295     Files int64
1296     Dirs int64 // the num. of directories
1297     SymLink int64
1298     Flats int64 // the num. of flat files
1299     MaxDepth int64
1300     MaxNamlen int64 // max. name length
1301     nextRepo time.Time
1302 }
1303 func showFusage(dir string, fusage *fileSum){
1304     bsume := float64(((fusage.Blocks-fusage.DupBlocks)/2)*1024)/1000000.0
1305     //bsumdup := float64((fusage.Blocks/2)*1024)/1000000.0
1306     fmt.Printf("%v: %v files (%vd %vs %vh) %.6f MB (%.2f MBK)\n",
1307         dir,
1308         fusage.Files,
1309         fusage.Dirs,
1310         fusage.SymLink,
1311         fusage.HLinks,
1312         float64(fusage.Size)/1000000.0, bsume);
1313 }
1314
1315 const (
1316     S_IFMT = 0170000
1317     S_IFCHR = 0020000
1318     S_IFDIR = 0040000
1319     S_IFREG = 0100000
1320     S_IFLNK = 0120000
1321     S_IFSOCK = 0140000
1322 )
1323 func cumFinfo(fsum *fileSum, path string, stater error, fstat syscall.Stat_t, argv[]string, verb bool)(*fileSum){
1324     now := time.Now()
1325     if time.Second <= now.Sub(fsum.nextRepo) {
1326         if !fsum.nextRepo.IsZero(){
1327             tstamp := now.Format(time.Stamp)
1328             showFusage(tstamp, fsum)
1329         }
1330         fsum.nextRepo = now.Add(time.Second)
1331     }
1332     if stater != nil {
1333         fsum.Err += 1
1334         return fsum
1335     }
1336     fsum.Files += 1
1337     if 1 < fstat.Nlink {
1338         // must count only once...
1339         // at least ignore ones in the same directory
1340         //if finfo.Mode().IsRegular() {
1341             if (fstat.Mode & S_IFMT) == S_IFREG {
1342                 fsum.HLinks += 1
1343                 fsum.DupBlocks += int64(fstat.Blocks)
1344                 //fmt.Printf("---Dup HardLink %v %s\n", fstat.Nlink, path)
1345             }
1346         }
1347         //fsum.Size += finfo.Size()
1348         fsum.Size += fstat.Size
1349         fsum.Blocks += int64(fstat.Blocks)
1350         //if verb { fmt.Printf("(%8dBlk) %s", fstat.Blocks/2, path) }
1351         if isin("-ls", argv){
1352             //if verb { fmt.Printf("%4d %8d ", fstat.Blksize, fstat.Blocks) }
1353             // fmt.Printf("%d\t", fstat.Blocks/2)
1354         }
1355         //if finfo.IsDir()
1356         if (fstat.Mode & S_IFMT) == S_IFDIR {
1357             fsum.Dirs += 1
1358         }
1359         //if (finfo.Mode() & os.Modesymlink) != 0
1360         if (fstat.Mode & S_IFMT) == S_IFLNK {
1361             //if verb { fmt.Printf("symlink(%v,%s)\n", fstat.Mode, finfo.Name()) }
1362             //{ fmt.Printf("symlink(%o,%s)\n", fstat.Mode, finfo.Name()) }
1363             fsum.SymLink += 1
1364         }
1365     }

```

```

1365     return fsum
1366 }
1367 func (gsh*GshContext)xxFindEntv(depth int,total *fileSum,dir string, dstat syscall.Stat_t, ei int, envt []string,npatv[]string,argv[]string)(*fileSum){
1368     nols := isin("-grep",argv)
1369     // sort envt
1370     /*
1371     if isin("-t",argv){
1372         sort.Slice(filev, func(i,j int) bool {
1373             return 0 < filev[i].ModTime().Sub(filev[j].ModTime())
1374         })
1375     }
1376     */
1377     /*
1378     if isin("-u",argv){
1379         sort.Slice(filev, func(i,j int) bool {
1380             return 0 < filev[i].AccTime().Sub(filev[j].AccTime())
1381         })
1382     }
1383     if isin("-U",argv){
1384         sort.Slice(filev, func(i,j int) bool {
1385             return 0 < filev[i].CreateTime().Sub(filev[j].CreateTime())
1386         })
1387     }
1388     */
1389     /*
1390     if isin("-S",argv){
1391         sort.Slice(filev, func(i,j int) bool {
1392             return filev[j].Size() < filev[i].Size()
1393         })
1394     }
1395     */
1396     for _,filename := range envt {
1397         for _,npat := range npatv {
1398             match := true
1399             if npat == "*" {
1400                 match = true
1401             }else{
1402                 match, _ = filepath.Match(npat,filename)
1403             }
1404             path := dir + DIRSEP + filename
1405             if !match {
1406                 continue
1407             }
1408             var fstat syscall.Stat_t
1409             staterr := syscall.Lstat(path,&fstat)
1410             if staterr != nil {
1411                 if !isin("-w",argv){fmt.Printf("ufind: %v\n",staterr) }
1412                 continue;
1413             }
1414             if isin("-du",argv) && (fstat.Mode & S_IFMT) == S_IFDIR {
1415                 // should not show size of directory in "-du" mode ...
1416             }else
1417             if !nols && !isin("-s",argv) && (!isin("-du",argv) || isin("-a",argv)) {
1418                 if isin("-du",argv) {
1419                     fmt.Printf("%d\t",fstat.Blocks/2)
1420                 }
1421                 showFileInfo(path,argv)
1422             }
1423             if true { // && isin("-du",argv)
1424                 total = cumFinfo(total,path,staterr,fstat,argv,false)
1425             }
1426             /*
1427             if isin("-wc",argv) {
1428             }
1429             */
1430             if gsh.lastChecksum.SumType != 0 {
1431                 gsh.xChecksum(path,argv,&gsh.lastChecksum);
1432             }
1433             x := isinX("-grep",argv); // -grep will be convenient like -ls
1434             if 0 <= x && x+1 <= len(argv) { // -grep will be convenient like -ls
1435                 if !isRegFile(path){
1436                     found := gsh.xGrep(path,argv[x+1:])
1437                     if 0 < found {
1438                         foundv := gsh.CmdCurrent.FoundFile
1439                         if len(foundv) < 10 {
1440                             gsh.CmdCurrent.FoundFile =
1441                                 append(gsh.CmdCurrent.FoundFile,path)
1442                         }
1443                     }
1444                 }
1445             }
1446             if !isin("-r0",argv) { // -d 0 in du, -depth n in find
1447                 //total.Depth += 1
1448                 if (fstat.Mode & S_IFMT) == S_IFLNK {
1449                     continue
1450                 }
1451                 if dstat.Rdev != fstat.Rdev {
1452                     fmt.Printf("--I-- don't follow different device %v(%v) %v(%v)\n",
1453                         dir,dstat.Rdev,path,fstat.Rdev)
1454                 }
1455                 if (fstat.Mode & S_IFMT) == S_IFDIR {
1456                     total = gsh.xxFind(depth+1,total,path,npatv,argv)
1457                 }
1458             }
1459         }
1460     }
1461     return total
1462 }
1463 func (gsh*GshContext)xxFind(depth int,total *fileSum,dir string,npatv[]string,argv[]string)(*fileSum){
1464     nols := isin("-grep",argv)
1465     dirfile,oerr := os.OpenFile(dir,os.O_RDONLY,0)
1466     if oerr == nil {
1467         //fmt.Printf("--I-- %v(%v)[%d]\n",dir,dirfile,dirfile.Fd())
1468         defer dirfile.Close()
1469     }else{
1470     }
1471     prev := *total
1472     var dstat syscall.Stat_t
1473     staterr := syscall.Lstat(dir,&dstat) // should be flstat
1474     /*
1475     if staterr != nil {
1476         if !isin("-w",argv){ fmt.Printf("ufind: %v\n",staterr) }
1477         return total
1478     }
1479     */
1480     //filev,err := ioutil.ReadDir(dir)
1481     //_,err := ioutil.ReadDir(dir) // ReadDir() heavy and bad for huge directory
1482     /*
1483     if err != nil {
1484         if !isin("-w",argv){ fmt.Printf("ufind: %v\n",err) }
1485         return total
1486     }
1487     */
1488     if depth == 0 {

```

```

1489     total = cumFinfo(total,dir,staterr,dstat,argv,true)
1490     if !mols && !isin("-s",argv) && (!isin("-du",argv) || isin("-a",argv)) {
1491         showFileInfo(dir,argv)
1492     }
1493 }
1494 // it it is not a directory, just scan it and finish
1495
1496 for ei := 0; ; ei++ {
1497     entv,rderr := dirfile.Readdirnames(8*1024)
1498     if len(entv) == 0 || rderr != nil {
1499         //if rderr != nil { fmt.Printf("[%d] len=%d (%v)\n",ei,len(entv),rderr) }
1500         break
1501     }
1502     if 0 < ei {
1503         fmt.Printf("--I-- xxFind[%d] %d large-dir: %s\n",ei,len(entv),dir)
1504     }
1505     total = gsh.xxFindEntv(depth,total,dir,dstat,ei,entv,npatv,argv)
1506 }
1507 if isin("-du",argv) {
1508     // if in "du" mode
1509     fmt.Printf("%d\t%s\n",total.Blocks-prev.Blocks)/2,dir)
1510 }
1511 return total
1512 }
1513
1514 // {ufind|fu|ls} [Files] [// Names] [-- Expressions]
1515 // Files is "." by default
1516 // Names is "*" by default
1517 // Expressions is "-print" by default for "ufind", or -du for "fu" command
1518 func (gsh*GshContext)xFind(argv[]string){
1519     if 0 < len(argv) && strBegins(argv[0],"?"){
1520         showFound(gsh,argv)
1521         return
1522     }
1523     if isin("-cksum",argv) || isin("-sum",argv) {
1524         gsh.lastChecksum = CheckSum{}
1525         if isin("-sum",argv) && isin("-add",argv) {
1526             gsh.lastChecksum.SumType |= SUM_SUM64
1527         }else
1528         if isin("-sum",argv) && isin("-size",argv) {
1529             gsh.lastChecksum.SumType |= SUM_SIZE
1530         }else
1531         if isin("-sum",argv) && isin("-bsd",argv) {
1532             gsh.lastChecksum.SumType |= SUM_SUM16_BSD
1533         }else
1534         if isin("-sum",argv) && isin("-sysv",argv) {
1535             gsh.lastChecksum.SumType |= SUM_SUM16_SYSV
1536         }else
1537         if isin("-sum",argv) {
1538             gsh.lastChecksum.SumType |= SUM_SUM64
1539         }
1540     }
1541     if isin("-unix",argv) {
1542         gsh.lastChecksum.SumType |= SUM_UNIXFILE
1543         gsh.lastChecksum.Crc32Table = *crc32.MakeTable(CRC32UNIX)
1544     }
1545     if isin("-ieee",argv){
1546         gsh.lastChecksum.SumType |= SUM_CRCIEEE
1547         gsh.lastChecksum.Crc32Table = *crc32.MakeTable(CRC32IEEE)
1548     }
1549     gsh.lastChecksum.RusgAtStart = Getrusagev()
1550 }
1551 var total = fileSum{}
1552 npats := []string{}
1553 for _,v := range argv {
1554     if 0 < len(v) && v[0] != '-' {
1555         npats = append(npats,v)
1556     }
1557     if v == "/" { break }
1558     if v == "--" { break }
1559     if v == "-grep" { break }
1560     if v == "-ls" { break }
1561 }
1562 if len(npats) == 0 {
1563     npats = []string{"*"}
1564 }
1565 cwd := "."
1566 // if to be fullpath :: cwd, _ := os.Getwd()
1567 if len(npats) == 0 { npats = []string{"*"} }
1568 fusage := gsh.xxFind(0,&total,cwd,npats,argv)
1569 if gsh.lastChecksum.SumType != 0 {
1570     var sumi uint64 = 0
1571     sum := &gsh.lastChecksum
1572     if (sum.SumType & SUM_SIZE) != 0 {
1573         sumi = uint64(sum.Size)
1574     }
1575     if (sum.SumType & SUM_SUM64) != 0 {
1576         sumi = sum.Sum64
1577     }
1578     if (sum.SumType & SUM_SUM16_SYSV) != 0 {
1579         s := uint32(sum.Sum16)
1580         r := (s & 0xPPFF) + ((s & 0xPPFFPPFF) >> 16)
1581         s = (r & 0xPPFF) + (r >> 16)
1582         sum.Crc32Val = uint32(s)
1583         sumi = uint64(s)
1584     }
1585     if (sum.SumType & SUM_SUM16_BSD) != 0 {
1586         sum.Crc32Val = uint32(sum.Sum16)
1587         sumi = uint64(sum.Sum16)
1588     }
1589     if (sum.SumType & SUM_UNIXFILE) != 0 {
1590         sum.Crc32Val = byteCRC32end(sum.Crc32Val,uint64(sum.Size))
1591         sumi = uint64(byteCRC32end(sum.Crc32Val,uint64(sum.Size)))
1592     }
1593     if 1 < sum.Files {
1594         fmt.Printf("%v %v // %v / %v files, %v/file\n",
1595             sumi,sum.Size,
1596             abssize(sum.Size),sum.Files,
1597             abssize(sum.Size)/sum.Files)
1598     }else{
1599         fmt.Printf("%v %v %v\n",
1600             sumi,sum.Size,npats[0])
1601     }
1602 }
1603 if !isin("-grep",argv) {
1604     showFusage("total",fusage)
1605 }
1606 if !isin("-s",argv){
1607     hits := len(gsh.CmdCurrent.FoundFile)
1608     if 0 < hits {
1609         fmt.Printf("--I-- %d files hits // can be refered with !%df\n",
1610             hits,len(gsh.CommandHistory))
1611     }
1612 }
1613 if gsh.lastChecksum.SumType != 0 {

```

```

1613     if isin("-ru",argv) {
1614         sum := gsh.lastCheckSum
1615         sum.Done = time.Now()
1616         gsh.lastCheckSum.RusageAtEnd = Getrusagev()
1617         elps := sum.Done.Sub(sum.Start)
1618         fmt.Printf("--cksum-size: %v (%v) / %v files, %v/file\r\n",
1619             sum.Size,abssize(sum.Size),sum.Files,abssize(sum.Size/sum.Files))
1620         nanos := int64(elps)
1621         fmt.Printf("--cksum-time: %v/total, %v/file, %.1f files/s, %v\r\n",
1622             abftime(nanos),
1623             abftime(nanos/sum.Files),
1624             (float64(sum.Files)*1000000000.0)/float64(nanos),
1625             abbspeed(sum.Size,nanos))
1626         diff := RusageSubv(sum.RusageAtEnd,sum.RusageAtStart)
1627         fmt.Printf("--cksum-rusg: %v\n",sRusagef("",argv,diff))
1628     }
1629 }
1630 return
1631 }
1632
1633 func showFiles(files[]string){
1634     sp := ""
1635     for i,file := range files {
1636         if 0 < i { sp = " " } else { sp = "" }
1637         fmt.Printf(sp+"%s",escapeWhiteSP(file))
1638     }
1639 }
1640 func showFound(gshCtx *GshContext, argv[]string){
1641     for i,v := range gshCtx.CommandHistory {
1642         if 0 < len(v.FoundFile) {
1643             fmt.Printf("!\d (%d) ",i,len(v.FoundFile))
1644             if isin("-ls",argv){
1645                 fmt.Printf("\n")
1646                 for _,file := range v.FoundFile {
1647                     fmt.Printf("%s //sub number?")
1648                     showFileInfo(file,argv)
1649                 }
1650             }else{
1651                 showFiles(v.FoundFile)
1652                 fmt.Printf("\n")
1653             }
1654         }
1655     }
1656 }
1657
1658 func showMatchFile(filev []os.FileInfo, npat,dir string, argv[]string)(string,bool){
1659     fname := ""
1660     found := false
1661     for _,v := range filev {
1662         match, _ := filepath.Match(npat,(v.Name()))
1663         if match {
1664             fname = v.Name()
1665             found = true
1666             //fmt.Printf("!\d %s\n",i,v.Name())
1667             showIfExecutable(fname,dir,argv)
1668         }
1669     }
1670     return fname,found
1671 }
1672 func showIfExecutable(name,dir string,argv[]string)(ffullpath string,ffound bool){
1673     var fullpath string
1674     if strBegins(name,DIRSEP){
1675         fullpath = name
1676     }else{
1677         fullpath = dir + DIRSEP + name
1678     }
1679     fi, err := os.Stat(fullpath)
1680     if err != nil {
1681         fullpath = dir + DIRSEP + name + ".go"
1682         fi, err = os.Stat(fullpath)
1683     }
1684     if err == nil {
1685         fm := fi.Mode()
1686         if fm.IsRegular() {
1687             // R_OK=4, W_OK=2, X_OK=1, F_OK=0
1688             if syscall.Access(fullpath,5) == nil {
1689                 ffullpath = fullpath
1690                 ffound = true
1691                 if ! isin("-s", argv) {
1692                     showFileInfo(fullpath,argv)
1693                 }
1694             }
1695         }
1696     }
1697     return ffullpath, ffound
1698 }
1699 func which(list string, argv []string) (fullpathv []string, itis bool){
1700     if len(argv) <= 1 {
1701         fmt.Printf("Usage: which comand [-s] [-a] [-ls]\n")
1702         return []string(""), false
1703     }
1704     path := argv[1]
1705     if strBegins(path,"/") {
1706         // should check if executable?
1707         _,exOK := showIfExecutable(path,"",argv)
1708         fmt.Printf("--D-- %v exOK=%v\n",path,exOK)
1709         return []string(path),exOK
1710     }
1711     pathenv, efound := os.LookupEnv(list)
1712     if ! efound {
1713         fmt.Printf("--E-- which: no \"%s\" environment\n",list)
1714         return []string(""), false
1715     }
1716     showall := isin("-a",argv) || 0 <= strings.Index(path,"*")
1717     dirv := strings.Split(pathenv,PATHSEP)
1718     ffound := false
1719     ffullpath := path
1720     for _, dir := range dirv {
1721         if 0 <= strings.Index(path,"*") { // by wild-card
1722             list, _ := ioutil.ReadDir(dir)
1723             ffullpath, ffound = showMatchFile(list,path,dir,argv)
1724         }else{
1725             ffullpath, ffound = showIfExecutable(path,dir,argv)
1726         }
1727         //if ffound && !isin("-a", argv) {
1728         if ffound && !showall {
1729             break;
1730         }
1731     }
1732     return []string(ffullpath), ffound
1733 }
1734
1735 func stripLeadingWSParg(argv[]string) ([]string){
1736     for ; 0 < len(argv); {

```

```

1737     if len(argv[0]) == 0 {
1738         argv = argv[1:]
1739     }else{
1740         break
1741     }
1742 }
1743 return argv
1744 }
1745 func xEval(argv []string, nlend bool){
1746     argv = stripLeadingWSparg(argv)
1747     if len(argv) == 0 {
1748         fmt.Printf("eval [%%format] [Go-expression]\n")
1749         return
1750     }
1751     pfmt := "%v"
1752     if argv[0][0] == '%' {
1753         pfmt = argv[0]
1754         argv = argv[1:]
1755     }
1756     if len(argv) == 0 {
1757         return
1758     }
1759     gocode := strings.Join(argv, " ");
1760     //fmt.Printf("eval [%v] [%v]\n",pfmt,gocode)
1761     fset := token.NewFileSet()
1762     rval, _ := types.Eval(fset,nil,token.NoPos,gocode)
1763     fmt.Printf(pfmt,rval.Value)
1764     if nlend { fmt.Printf("\n") }
1765 }
1766
1767 func getval(name string) (found bool, val int) {
1768     /* should expand the name here */
1769     if name == "gsh.pid" {
1770         return true, os.Getpid()
1771     }else
1772     if name == "gsh.ppid" {
1773         return true, os.Getppid()
1774     }
1775     return false, 0
1776 }
1777
1778 func echo(argv []string, nlend bool){
1779     for ai := 1; ai < len(argv); ai++ {
1780         if 1 < ai {
1781             fmt.Printf(" ");
1782         }
1783         arg := argv[ai]
1784         found, val := getval(arg)
1785         if found {
1786             fmt.Printf("%d",val)
1787         }else{
1788             fmt.Printf("%s",arg)
1789         }
1790     }
1791     if nlend {
1792         fmt.Printf("\n");
1793     }
1794 }
1795
1796 func resfile() string {
1797     return "gsh.tmp"
1798 }
1799 //var resF *File
1800 func resmap() {
1801     //_, err := os.OpenFile(resfile(), os.O_RDWR|os.O_CREATE, os.ModeAppend)
1802     // https://developpaper.com/solution-to-golang-bad-file-descriptor-problem/
1803     _, err := os.OpenFile(resfile(), os.O_RDWR|os.O_CREATE, 0600)
1804     if err != nil {
1805         fmt.Printf("refF could not open: %s\n",err)
1806     }else{
1807         fmt.Printf("refF opened\n")
1808     }
1809 }
1810
1811 // @@2020-0821
1812 func gshScanArg(str string,strip int)(argv []string){
1813     var si = 0
1814     var sb = 0
1815     var inBracket = 0
1816     var arg1 = make([]byte,LINESIZE)
1817     var ax = 0
1818     debug := false
1819
1820     for ; si < len(str); si++ {
1821         if str[si] != ' ' {
1822             break
1823         }
1824     }
1825     sb = si
1826     for ; si < len(str); si++ {
1827         if sb <= si {
1828             if debug {
1829                 fmt.Printf("--Da- +%d %2d-%2d %s ... %s\n",
1830                     inBracket,sb,si,arg1[0:ax],str[si:])
1831             }
1832         }
1833         ch := str[si]
1834         if ch == '{' {
1835             inBracket += 1
1836             if 0 < strip && inBracket <= strip {
1837                 //fmt.Printf("stripLEV %d <= %d?\n",inBracket,strip)
1838                 continue
1839             }
1840         }
1841         if 0 < inBracket {
1842             if ch == '}' {
1843                 inBracket -= 1
1844                 if 0 < strip && inBracket < strip {
1845                     //fmt.Printf("stripLEV %d < %d?\n",inBracket,strip)
1846                     continue
1847                 }
1848             }
1849             arg1[ax] = ch
1850             ax += 1
1851             continue
1852         }
1853         if str[si] == ' ' {
1854             argv = append(argv,string(arg1[0:ax]))
1855             if debug {
1856                 fmt.Printf("--Da- [%v][%v-%v] %s ... %s\n",
1857                     -1+len(argv),sb,si,str[sb:si],string(str[si:]))
1858             }
1859             sb = si+1
1860             ax = 0

```

```

1861         continue
1862     }
1863     argl[ax] = ch
1864     ax += 1
1865 }
1866 if sb < si {
1867     argv = append(argv, string(argl[0:ax]))
1868     if debug {
1869         fmt.Printf("--Da- [%v][%-v] %s ... %s\n",
1870             -1+len(argv), sb, si, string(argl[0:ax]), string(str[si:]))
1871     }
1872 }
1873 if debug {
1874     fmt.Printf("--Da- %d [%s] => [%d]%\n", strip, str, len(argv), argv)
1875 }
1876 return argv
1877 }
1878
1879 // should get stderr (into tmpfile ?) and return
1880 func (gsh*GshContext)Popen(name, mode string) (pin*os.File, pout*os.File, err bool) {
1881     var pv = []int{-1, -1}
1882     syscall.Pipe(pv)
1883
1884     xarg := gshScanArg(name, 1)
1885     name = strings.Join(xarg, " ")
1886
1887     pin = os.NewFile(uintptr(pv[0]), "StdoutOf-"+name)
1888     pout = os.NewFile(uintptr(pv[1]), "StdinOf-"+name)
1889     fdix := 0
1890     dir := "?"
1891     if mode == "r" {
1892         dir = "<"
1893         fdix = 1 // read from the stdout of the process
1894     } else {
1895         dir = ">"
1896         fdix = 0 // write to the stdin of the process
1897     }
1898     gshPA := gsh.gshPA
1899     savfd := gshPA.Files[fdix]
1900
1901     var fd uintptr = 0
1902     if mode == "r" {
1903         fd = pout.Fd()
1904         gshPA.Files[fdix] = pout.Fd()
1905     } else {
1906         fd = pin.Fd()
1907         gshPA.Files[fdix] = pin.Fd()
1908     }
1909     // should do this by Goroutine?
1910     if false {
1911         fmt.Printf("--Ip- Opened fd[%v] %s %v\n", fd, dir, name)
1912         fmt.Printf("--RED1 [%d,%d,%d]->[%d,%d,%d]\n",
1913             os.Stdin.Fd(), os.Stdout.Fd(), os.Stderr.Fd(),
1914             pin.Fd(), pout.Fd(), pout.Fd())
1915     }
1916     savi := os.Stdin
1917     savo := os.Stdout
1918     save := os.Stderr
1919     os.Stdin = pin
1920     os.Stdout = pout
1921     os.Stderr = pout
1922     gsh.BackGround = true
1923     gsh.gshelllh(name)
1924     gsh.BackGround = false
1925     os.Stdin = savi
1926     os.Stdout = savo
1927     os.Stderr = save
1928
1929     gshPA.Files[fdix] = savfd
1930     return pin, pout, false
1931 }
1932
1933 // <a name="ex-commands">External commands</a>
1934 func (gsh*GshContext)excommand(exec bool, argv []string) (notif bool, exit bool) {
1935     if gsh.CmdTrace { fmt.Printf("--I-- excommand[%v](%v)\n", exec, argv) }
1936
1937     gshPA := gsh.gshPA
1938     fullpathv, itis := which("PATH", []string{"which", argv[0], "-s"})
1939     if itis == false {
1940         return true, false
1941     }
1942     fullpath := fullpathv[0]
1943     argv = unescapeWhiteSPV(argv)
1944     if 0 < strings.Index(fullpath, ".go") {
1945         nargv := argv // []string{}
1946         gofullpathv, itis := which("PATH", []string{"which", "go", "-s"})
1947         if itis == false {
1948             fmt.Printf("--F-- Go not found\n")
1949             return false, true
1950         }
1951         gofullpath := gofullpathv[0]
1952         nargv = []string{ gofullpath, "run", fullpath }
1953         fmt.Printf("--I-- %s %s %s %s\n", gofullpath,
1954             nargv[0], nargv[1], nargv[2])
1955         if exec {
1956             syscall.Exec(gofullpath, nargv, os.Environ())
1957         } else {
1958             pid, _ := syscall.ForkExec(gofullpath, nargv, &gshPA)
1959             if gsh.BackGround {
1960                 fmt.Fprintf(stderr, "--Ip- in Background pid[%d]%(v)\n", pid, len(argv), nargv)
1961                 gsh.BackGroundJobs = append(gsh.BackGroundJobs, pid)
1962             } else {
1963                 rusage := syscall.Rusage {}
1964                 syscall.Wait4(pid, nil, 0, &rusage)
1965                 gsh.LastRusage = rusage
1966                 gsh.CmdCurrent.Rusagev[1] = rusage
1967             }
1968         }
1969     } else {
1970         if exec {
1971             syscall.Exec(fullpath, argv, os.Environ())
1972         } else {
1973             pid, _ := syscall.ForkExec(fullpath, argv, &gshPA)
1974             //fmt.Printf("[%d]\n", pid); // '&' to be background
1975             if gsh.BackGround {
1976                 fmt.Fprintf(stderr, "--Ip- in Background pid[%d]%(v)\n", pid, len(argv), argv)
1977                 gsh.BackGroundJobs = append(gsh.BackGroundJobs, pid)
1978             } else {
1979                 rusage := syscall.Rusage {}
1980                 syscall.Wait4(pid, nil, 0, &rusage);
1981                 gsh.LastRusage = rusage
1982                 gsh.CmdCurrent.Rusagev[1] = rusage
1983             }
1984         }
1985     }

```



```

1985     }
1986     return false,false
1987 }
1988
1989 // <a name="builtin">Builtin Commands</a>
1990 func (gshCtx *GshContext) sleep(argv []string) {
1991     if len(argv) < 2 {
1992         fmt.Printf("Sleep 100ms, 100us, 100ns, ...\n")
1993         return
1994     }
1995     duration := argv[1];
1996     d, err := time.ParseDuration(duration)
1997     if err != nil {
1998         d, err = time.ParseDuration(duration+"s")
1999         if err != nil {
2000             fmt.Printf("duration ? %s (%s)\n",duration,err)
2001             return
2002         }
2003     }
2004     //fmt.Printf("Sleep %v\n",duration)
2005     time.Sleep(d)
2006     if 0 < len(argv[2:]) {
2007         gshCtx.gshellv(argv[2:])
2008     }
2009 }
2010 func (gshCtx *GshContext)repeat(argv []string) {
2011     if len(argv) < 2 {
2012         return
2013     }
2014     start0 := time.Now()
2015     for ri, _ := strconv.Atoi(argv[1]); 0 < ri; ri-- {
2016         if 0 < len(argv[2:]) {
2017             //start := time.Now()
2018             gshCtx.gshellv(argv[2:])
2019             end := time.Now()
2020             elps := end.Sub(start0);
2021             if( 1000000000 < elps ){
2022                 fmt.Printf("(repeat#%d %v)\n",ri,elps);
2023             }
2024         }
2025     }
2026 }
2027
2028 func (gshCtx *GshContext)gen(argv []string) {
2029     gshPA := gshCtx.gshPA
2030     if len(argv) < 2 {
2031         fmt.Printf("Usage: %s N\n",argv[0])
2032         return
2033     }
2034     // should br repeated by "repeat" command
2035     count, _ := strconv.Atoi(argv[1])
2036     fd := gshPA.Files[1] // Stdout
2037     file := os.NewFile(fd,"internalStdOut")
2038     fmt.Printf("-- Gen. Count=%d to [%d]\n",count,file.Fd())
2039     //buf := []byte{}
2040     outdata := "0123 5678 0123 5678 0123 5678 0123 5678\r"
2041     for gi := 0; gi < count; gi++ {
2042         file.WriteString(outdata)
2043     }
2044     //file.WriteString("\n")
2045     fmt.Printf("\n(%d B)\n",count*len(outdata));
2046     //file.Close()
2047 }
2048
2049 // <a name="rexec">Remote Execution</a> // 2020-0820
2050 func Elapsed(from time.Time)(string){
2051     elps := time.Now().Sub(from)
2052     if 1000000000 < elps {
2053         return fmt.Sprintf("[%5d.%02ds]",elps/1000000000,(elps%1000000000)/10000000)
2054     }else
2055     if 1000000 < elps {
2056         return fmt.Sprintf("[%3d.%03dms]",elps/1000000,(elps%1000000)/1000)
2057     }else{
2058         return fmt.Sprintf("[%3d.%03dus]",elps/1000,(elps%1000))
2059     }
2060 }
2061 func abftime(nanos int64)(string){
2062     if 1000000000 < nanos {
2063         return fmt.Sprintf("%d.%02ds",nanos/1000000000,(nanos%1000000000)/10000000)
2064     }else
2065     if 1000000 < nanos {
2066         return fmt.Sprintf("%d.%03dms",nanos/1000000,(nanos%1000000)/1000)
2067     }else{
2068         return fmt.Sprintf("%d.%03dus",nanos/1000,(nanos%1000))
2069     }
2070 }
2071 func abssize(size int64)(string){
2072     fsize := float64(size)
2073     if 1024*1024*1024 < size {
2074         return fmt.Sprintf("%.2fGiB",fsize/(1024*1024*1024))
2075     }else
2076     if 1024*1024 < size {
2077         return fmt.Sprintf("%.3fMiB",fsize/(1024*1024))
2078     }else{
2079         return fmt.Sprintf("%.3fKiB",fsize/1024)
2080     }
2081 }
2082 func absze(size int64)(string){
2083     fsize := float64(size)
2084     if 1024*1024*1024 < size {
2085         return fmt.Sprintf("%.2fGiB",fsize/(1024*1024*1024))
2086     }else
2087     if 1024*1024 < size {
2088         return fmt.Sprintf("%.3fMiB",fsize/(1024*1024))
2089     }else{
2090         return fmt.Sprintf("%.3fKiB",fsize/1024)
2091     }
2092 }
2093 func abbspeed(totalB int64,ns int64)(string){
2094     MBs := (float64(totalB)/1000000) / (float64(ns)/1000000000)
2095     if 1000 <= MBs {
2096         return fmt.Sprintf("%.3fGB/s",MBs/1000)
2097     }
2098     if 1 <= MBs {
2099         return fmt.Sprintf("%.3fMB/s",MBs)
2100     }else{
2101         return fmt.Sprintf("%.3fKB/s",MBs*1000)
2102     }
2103 }
2104 func abspeed(totalB int64,ns time.Duration)(string){
2105     MBs := (float64(totalB)/1000000) / (float64(ns)/1000000000)
2106     if 1000 <= MBs {
2107         return fmt.Sprintf("%.3fGBps",MBs/1000)
2108     }

```

```

2109     if 1 <= MBs {
2110         return fmt.Sprintf("%6.3fMBps",MBs)
2111     }else{
2112         return fmt.Sprintf("%6.3fKBps",MBs*1000)
2113     }
2114 }
2115 func fileRelay(what string,in*os.File,out*os.File,size int64,bsiz int)(wcount int64){
2116     Start := time.Now()
2117     buff := make([]byte,bsiz)
2118     var total int64 = 0
2119     var rem int64 = size
2120     nio := 0
2121     Prev := time.Now()
2122     var PrevSize int64 = 0
2123
2124     fmt.Printf(Elapsed(Start)+"--In- X: %s (%v/%v/%v) START\n",
2125         what,absize(total),size,nio)
2126
2127     for i:= 0; ; i++ {
2128         var len = bsiz
2129         if int(rem) < len {
2130             len = int(rem)
2131         }
2132         Now := time.Now()
2133         Elps := Now.Sub(Prev);
2134         if 100000000 < Now.Sub(Prev) {
2135             fmt.Printf(Elapsed(Start)+"--In- X: %s (%v/%v/%v) %s\n",
2136                 what,absize(total),size,nio,
2137                 abspeed((total-PrevSize),Elps))
2138             Prev = Now;
2139             PrevSize = total
2140         }
2141         rlen := len
2142         if in != nil {
2143             // should watch the disconnection of out
2144             rcc,err := in.Read(buff[0:rlen])
2145             if err != nil {
2146                 fmt.Printf(Elapsed(Start)+"--En- X: %s read(%v,%v)<%v\n",
2147                     what,rcc,err,in.Name())
2148                 break
2149             }
2150             rlen = rcc
2151             if string(buff[0:10]) == "(SoftEOF " {
2152                 var ecc int64 = 0
2153                 fmt.Sscanf(string(buff),"(SoftEOF %v",&ecc)
2154                 fmt.Printf(Elapsed(Start)+"--En- X: %s Recv ((SoftEOF %v))/%v\n",
2155                     what,ecc,total)
2156                 if ecc == total {
2157                     break
2158                 }
2159             }
2160         }
2161         wlen := rlen
2162         if out != nil {
2163             wcc,err := out.Write(buff[0:wlen])
2164             if err != nil {
2165                 fmt.Printf(Elapsed(Start)+"--En-- X: %s write(%v,%v)>%v\n",
2166                     what,wcc,err,out.Name())
2167                 break
2168             }
2169             wlen = wcc
2170         }
2171         if wlen < rlen {
2172             fmt.Printf(Elapsed(Start)+"--En- X: %s incomplete write (%v/%v)\n",
2173                 what,wlen,rlen)
2174             break;
2175         }
2176     }
2177     nio += 1
2178     total += int64(rlen)
2179     rem -= int64(rlen)
2180     if rem <= 0 {
2181         break
2182     }
2183 }
2184
2185 Done := time.Now()
2186 Elps := float64(Done.Sub(Start))/1000000000 //Seconds
2187 TotalMB := float64(total)/1000000 //MB
2188 MBps := TotalMB / Elps
2189 fmt.Printf(Elapsed(Start)+"--In- X: %s (%v/%v/%v) %v %6.3fMB/s\n",
2190     what,total,size,nio,absize(total),MBps)
2191 return total
2192 }
2193 func tcpPush(clnt *os.File){
2194     // shrink socket buffer and recover
2195     usleep(100);
2196 }
2197 func (gsh*GshContext)RexecServer(argv[]string){
2198     debug := true
2199     Start0 := time.Now()
2200     Start := Start0
2201     // if local == "" { local = "0.0.0.0:9999" }
2202     local := "0.0.0.0:9999"
2203
2204     if 0 < len(argv) {
2205         if argv[0] == "-s" {
2206             debug = false
2207             argv = argv[1:]
2208         }
2209     }
2210     if 0 < len(argv) {
2211         argv = argv[1:]
2212     }
2213     port, err := net.ResolveTCPAddr("tcp",local);
2214     if err != nil {
2215         fmt.Printf("--En- S: Address error: %s (%s)\n",local,err)
2216         return
2217     }
2218     fmt.Printf(Elapsed(Start)+"--In- S: Listening at %s...\n",local);
2219     sconn, err := net.ListenTCP("tcp", port)
2220     if err != nil {
2221         fmt.Printf(Elapsed(Start)+"--En- S: Listen error: %s (%s)\n",local,err)
2222         return
2223     }
2224
2225     reqbuf := make([]byte,LINESIZE)
2226     res := ""
2227     for {
2228         fmt.Printf(Elapsed(Start0)+"--In- S: Listening at %s...\n",local);
2229         aconn, err := sconn.AcceptTCP()
2230         Start = time.Now()
2231         if err != nil {
2232             fmt.Printf(Elapsed(Start)+"--En- S: Accept error: %s (%s)\n",local,err)

```

```

2233     return
2234 }
2235 clnt, _ := aconn.File()
2236 fd := clnt.Fd()
2237 ar := aconn.RemoteAddr()
2238 if debug { fmt.Printf(Elapsed(Start0)+"--In- S: Accepted TCP at %s [%d] <- %v\n",
2239     local,fd,ar) }
2240 res = fmt.Sprintf("220 GShell/%s Server\r\n",VERSION)
2241 fmt.Fprintf(clnt,"%s",res)
2242 if debug { fmt.Printf(Elapsed(Start)+"--In- S: %s",res) }
2243 count, err := clnt.Read(reqbuf)
2244 if err != nil {
2245     fmt.Printf(Elapsed(Start)+"--En- C: (%v %v) %v",
2246         count,err,string(reqbuf))
2247 }
2248 req := string(reqbuf[:count])
2249 if debug { fmt.Printf(Elapsed(Start)+"--In- C: %v",string(req)) }
2250 reqv := strings.Split(string(req),"")
2251 cmdv := gshScanArg(reqv[0],0)
2252 //cmdv := strings.Split(reqv[0]," ")
2253 switch cmdv[0] {
2254     case "HELO":
2255         res = fmt.Sprintf("250 %v",req)
2256     case "GET":
2257         // download {remotefile|-zN} [localfile]
2258         var dsize int64 = 32*1024*1024
2259         var bsize int = 64*1024
2260         var fname string = ""
2261         var in *os.File = nil
2262         var pseudoEOF = false
2263         if l < len(cmdv) {
2264             fname = cmdv[l]
2265             if strBegins(fname,"-z") {
2266                 fmt.Sscanf(fname[2:],"%d",&dsize)
2267             }else{
2268                 if strBegins(fname,"{") {
2269                     xin,xout,err := gsh.Popen(fname,"r")
2270                     if err {
2271                         }else{
2272                             xout.Close()
2273                             defer xin.Close()
2274                             in = xin
2275                             dsize = MaxStreamSize
2276                             pseudoEOF = true
2277                         }
2278                     }else{
2279                         xin,err := os.Open(fname)
2280                         if err != nil {
2281                             fmt.Printf("--En- GET (%v)\n",err)
2282                         }else{
2283                             defer xin.Close()
2284                             in = xin
2285                             fi,_ := xin.Stat()
2286                             dsize = fi.Size()
2287                         }
2288                     }
2289                 }
2290                 //fmt.Printf(Elapsed(Start)+"--In- GET %v:%v\n",dsize,bsize)
2291                 res = fmt.Sprintf("200 %v\r\n",dsize)
2292                 fmt.Fprintf(clnt,"%v",res)
2293                 tcpPush(clnt); // should be separated as line in receiver
2294                 fmt.Printf(Elapsed(Start)+"--In- S: %v",res)
2295                 wcount := fileRelay("SendGET",in,clnt,dsize,bsize)
2296                 if pseudoEOF {
2297                     in.Close() // pipe from the command
2298                     // show end of stream data (its size) by OOB?
2299                     SoftEOF := fmt.Sprintf("(SoftEOF %v)",wcount)
2300                     fmt.Printf(Elapsed(Start)+"--In- S: Send %v\n",SoftEOF)
2301                 }
2302                 tcpPush(clnt); // to let SoftEOF data apper at the top of received data
2303                 fmt.Fprintf(clnt,"%v\r\n",SoftEOF)
2304                 tcpPush(clnt); // to let SoftEOF alone in a packet (separate with 200 OK)
2305                 // with client generated random?
2306                 //fmt.Printf("--In- L: close %v (%v)\n",in.Fd(),in.Name())
2307             }
2308             res = fmt.Sprintf("200 GET done\r\n")
2309         case "PUT":
2310             // upload {srcfile|-zN} [dstfile]
2311             var dsize int64 = 32*1024*1024
2312             var bsize int = 64*1024
2313             var fname string = ""
2314             var out *os.File = nil
2315             if l < len(cmdv) { // localfile
2316                 fmt.Sscanf(cmdv[l],"%d",&dsize)
2317             }
2318             if 2 < len(cmdv) {
2319                 fname = cmdv[2]
2320                 if fname == "-" {
2321                     // nul dev
2322                 }else{
2323                     if strBegins(fname,"{") {
2324                         xin,xout,err := gsh.Popen(fname,"w")
2325                         if err {
2326                             }else{
2327                                 xin.Close()
2328                                 defer xout.Close()
2329                                 out = xout
2330                             }
2331                         }else{
2332                             // should write to temporary file
2333                             // should suppress ^C on tty
2334                             xout,err := os.OpenFile(fname,os.O_CREATE|os.O_RDWR|os.O_TRUNC,0600)
2335                             //fmt.Printf("--In- S: open(%v) out(%v) err(%v)\n",fname,xout,err)
2336                             if err != nil {
2337                                 fmt.Printf("--En- PUT (%v)\n",err)
2338                             }else{
2339                                 out = xout
2340                             }
2341                         }
2342                     }
2343                     fmt.Printf(Elapsed(Start)+"--In- L: open(%v,w) %v (%v)\n",
2344                         fname,local,err)
2345                 }
2346                 fmt.Printf(Elapsed(Start)+"--In- PUT %v (%v)\n",dsize,bsize)
2347                 fmt.Printf(Elapsed(Start)+"--In- S: 200 %v OK\r\n",dsize)
2348                 fmt.Fprintf(clnt,"200 %v OK\r\n",dsize)
2349                 fileRelay("RecvPUT",clnt,out,dsize,bsize)
2350                 res = fmt.Sprintf("200 PUT done\r\n")
2351             default:
2352                 res = fmt.Sprintf("400 What? %v",req)
2353         }
2354     }
2355     swcc,serr := clnt.Write([]byte(res))
2356     if serr != nil {
2357         fmt.Printf(Elapsed(Start)+"--In- S: (%v=%v er=%v) %v",swcc,serr,res)
2358     }else{

```

```

2357         fmt.Printf(Elapsed(Start)+"--In- S: %v",res)
2358     }
2359     aconn.Close();
2360     c1nt.Close();
2361 }
2362     sconn.Close();
2363 }
2364 func (gsh*GshContext)RexecClient(argv[]string)(int,string){
2365     debug := true
2366     Start := time.Now()
2367     if len(argv) == 1 {
2368         return -1,"EmptyARG"
2369     }
2370     argv = argv[1:]
2371     if argv[0] == "-serv" {
2372         gsh.RexecServer(argv[1:])
2373         return 0,"Server"
2374     }
2375     remote := "0.0.0.0:9999"
2376     if argv[0][0] == '@' {
2377         remote = argv[0][1:]
2378         argv = argv[1:]
2379     }
2380     if argv[0] == "-s" {
2381         debug = false
2382         argv = argv[1:]
2383     }
2384     dport, err := net.ResolveTCPAddr("tcp",remote);
2385     if err != nil {
2386         fmt.Printf(Elapsed(Start)+"Address error: %s (%s)\n",remote,err)
2387         return -1,"AddressError"
2388     }
2389     fmt.Printf(Elapsed(Start)+"--In- C: Connecting to %s\n",remote)
2390     serv, err := net.DialTCP("tcp",nil,dport)
2391     if err != nil {
2392         fmt.Printf(Elapsed(Start)+"Connection error: %s (%s)\n",remote,err)
2393         return -1,"CannotConnect"
2394     }
2395     if debug {
2396         al := serv.LocalAddr()
2397         fmt.Printf(Elapsed(Start)+"--In- C: Connected to %v <- %v\n",remote,al)
2398     }
2399
2400     req := ""
2401     res := make([]byte,LINESIZE)
2402     count,err := serv.Read(res)
2403     if err != nil {
2404         fmt.Printf("--En- S: (%3d,%v) %v",count,err,string(res))
2405     }
2406     if debug { fmt.Printf(Elapsed(Start)+"--In- S: %v",string(res)) }
2407
2408     if argv[0] == "GET" {
2409         savPA := gsh.gshPA
2410         var bsize int = 64*1024
2411         req = fmt.Sprintf("%v\n",strings.Join(argv, " "))
2412         fmt.Printf(Elapsed(Start)+"--In- C: %v",req)
2413         fmt.Pprintf(serv,req)
2414         count,err = serv.Read(res)
2415         if err != nil {
2416             }else{
2417                 var dsize int64 = 0
2418                 var out *os.File = nil
2419                 var out_tobeclosed *os.File = nil
2420                 var fname string = ""
2421                 var rcode int = 0
2422                 var pid int = -1
2423                 fmt.Sscanf(string(res),"%d %d",&rcode,&dsize)
2424                 fmt.Printf(Elapsed(Start)+"--In- S: %v",string(res[0:count]))
2425                 if 3 <= len(argv) {
2426                     fname = argv[2]
2427                     if strBegins(fname,"{") {
2428                         xin,xout,err := gsh.Popen(fname,"w")
2429                         if err {
2430                             }else{
2431                                 xin.Close()
2432                                 defer xout.Close()
2433                                 out = xout
2434                                 out_tobeclosed = xout
2435                                 pid = 0 // should be its pid
2436                             }
2437                         }else{
2438                             // should write to temporary file
2439                             // should suppress ^C on tty
2440                             xout,err := os.OpenFile(fname,os.O_CREATE|os.O_RDWR|os.O_TRUNC,0600)
2441                             if err != nil {
2442                                 fmt.Print("--En- %v\n",err)
2443                             }
2444                             out = xout
2445                             //fmt.Printf("--In-- %d > %s\n",out.Pd(),fname)
2446                         }
2447                     }
2448                     in,_ := serv.File()
2449                     fileRelay("RecvGET",in,out,dsize,bsize)
2450                     if 0 <= pid {
2451                         gsh.gshPA = savPA // recovery of Pd(), and more?
2452                         fmt.Printf(Elapsed(Start)+"--In- L: close Pipe > %v\n",fname)
2453                         out_tobeclosed.Close()
2454                         //syscall.Wait4(pid,nil,0,nil) //@@
2455                     }
2456                 }
2457             }else
2458             if argv[0] == "PUT" {
2459                 remote, _ := serv.File()
2460                 var local *os.File = nil
2461                 var dsize int64 = 32*1024*1024
2462                 var bsize int = 64*1024
2463                 var ofile string = "-"
2464                 //fmt.Printf("--I-- Rex %v\n",argv)
2465                 if 1 < len(argv) {
2466                     fname := argv[1]
2467                     if strBegins(fname,"-z") {
2468                         fmt.Sscanf(fname[2:],"%d",&dsize)
2469                     }else
2470                     if strBegins(fname,"{") {
2471                         xin,xout,err := gsh.Popen(fname,"r")
2472                         if err {
2473                             }else{
2474                                 xout.Close()
2475                                 defer xin.Close()
2476                                 //in = xin
2477                                 local = xin
2478                                 fmt.Printf("--In- [%d] < Upload output of %v\n",
2479                                     local.Pd(),fname)
2480                                 ofile = "-from."+fname

```

```

2481         dsize = MaxStreamSize
2482     }
2483     }else{
2484         xlocal_err := os.Open(fname)
2485         if err != nil {
2486             fmt.Printf("--En- (%s)\n",err)
2487             local = nil
2488         }else{
2489             local = xlocal
2490             fi, _ := local.Stat()
2491             dsize = fi.Size()
2492             defer local.Close()
2493             //fmt.Printf("--I-- Rex in(%v / %v)\n",ofile,dsize)
2494         }
2495         ofile = fname
2496         fmt.Printf(Elapsed(Start)+"--In- L: open(%v,r)=%v %v (%v)\n",
2497             fname,dsize,local,err)
2498     }
2499 }
2500 if 2 < len(argv) && argv[2] != "" {
2501     ofile = argv[2]
2502     //fmt.Printf("(%d)%v B.ofile=%v\n",len(argv),argv,ofile)
2503 }
2504 //fmt.Printf(Elapsed(Start)+"--I-- Rex out(%v)\n",ofile)
2505 fmt.Printf(Elapsed(Start)+"--In- PUT %v (/ %v)\n",dsize,bsize)
2506 req = fmt.Sprintf("PUT %v %v \r\n",dsize,ofile)
2507 if debug { fmt.Printf(Elapsed(Start)+"--In- C: %v",req) }
2508 fmt.Fprintf(serv,"%v",req)
2509 count,err = serv.Read(res)
2510 if debug { fmt.Printf(Elapsed(Start)+"--In- S: %v",string(res[0:count])) }
2511 fileRelay("SendPUT",local,remote,dsize,bsize)
2512 }else{
2513     req = fmt.Sprintf("%v\r\n",strings.Join(argv," "))
2514     if debug { fmt.Printf(Elapsed(Start)+"--In- C: %v",req) }
2515     fmt.Fprintf(serv,"%v",req)
2516     //fmt.Printf("--In- sending RexRequest(%v)\n",len(req))
2517 }
2518 //fmt.Printf(Elapsed(Start)+"--In- waiting RexResponse...\n")
2519 count,err = serv.Read(res)
2520 ress := ""
2521 if count == 0 {
2522     ress = "(nil)\r\n"
2523 }else{
2524     ress = string(res[:count])
2525 }
2526 if err != nil {
2527     fmt.Printf(Elapsed(Start)+"--En- S: (%d,%v) %v",count,err,ress)
2528 }else{
2529     fmt.Printf(Elapsed(Start)+"--In- S: %v",ress)
2530 }
2531 serv.Close()
2532 //conn.Close()
2533 }
2534 var stat string
2535 var rcode int
2536 fmt.Sscanf(ress,"%d %s",&rcode,&stat)
2537 //fmt.Printf("--D-- Client: %v (%v)",rcode,stat)
2538 return rcode,ress
2539 }
2540 }
2541 // <a name="remote-sh">Remote Shell</a>
2542 // gcp file [...] { [host]:[port]:[dir] | dir } // -p | -no-p
2543 func (gsh*GshContext)FileCopy(argv[[]string){
2544     var host = ""
2545     var port = ""
2546     var upload = false
2547     var download = false
2548     var xargv = []string{"rex-gcp"}
2549     var srcv = []string{}
2550     var dstv = []string{}
2551     argv = argv[1:]
2552 }
2553 for _,v := range argv {
2554     /*
2555     if v[0] == '-' { // might be a pseudo file (generated date)
2556         continue
2557     }
2558     */
2559     obj := strings.Split(v,":")
2560     //fmt.Printf("%d %v %v\n",len(obj),v,obj)
2561     if 1 < len(obj) {
2562         host = obj[0]
2563         file := ""
2564         if 0 < len(host) {
2565             gsh.LastServer.host = host
2566         }else{
2567             host = gsh.LastServer.host
2568             port = gsh.LastServer.port
2569         }
2570         if 2 < len(obj) {
2571             port = obj[1]
2572             if 0 < len(port) {
2573                 gsh.LastServer.port = port
2574             }else{
2575                 port = gsh.LastServer.port
2576             }
2577             file = obj[2]
2578         }else{
2579             file = obj[1]
2580         }
2581         if len(srcv) == 0 {
2582             download = true
2583             srcv = append(srcv,file)
2584             continue
2585         }
2586         upload = true
2587         dstv = append(dstv,file)
2588         continue
2589     }
2590     /*
2591     idx := strings.Index(v,":")
2592     if 0 <= idx {
2593         remote = v[0:idx]
2594         if len(srcv) == 0 {
2595             download = true
2596             srcv = append(srcv,v[idx+1:])
2597             continue
2598         }
2599         upload = true
2600         dstv = append(dstv,v[idx+1:])
2601         continue
2602     }
2603     */
2604     if download {

```

```

2605         dstv = append(dstv,v)
2606     }else{
2607         srcv = append(srcv,v)
2608     }
2609 }
2610 hostport := "@" + host + ":" + port
2611 if upload {
2612     if host != "" { xargv = append(xargv,hostport) }
2613     xargv = append(xargv,"PUT")
2614     xargv = append(xargv,srcv[0]...)
2615     xargv = append(xargv,dstv[0]...)
2616     //fmt.Printf("--I-- FileCopy PUT gsh://%/s/%v < %v // %v\n",hostport,dstv,srcv,xargv)
2617     fmt.Printf("--I-- FileCopy PUT gsh://%/s/%v < %v\n",hostport,dstv,srcv)
2618     gsh.RexecClient(xargv)
2619 }else
2620 if download {
2621     if host != "" { xargv = append(xargv,hostport) }
2622     xargv = append(xargv,"GET")
2623     xargv = append(xargv,srcv[0]...)
2624     xargv = append(xargv,dstv[0]...)
2625     //fmt.Printf("--I-- FileCopy GET gsh://%/v/%v > %v // %v\n",hostport,srcv,dstv,xargv)
2626     fmt.Printf("--I-- FileCopy GET gsh://%/v/%v > %v\n",hostport,srcv,dstv)
2627     gsh.RexecClient(xargv)
2628 }else{
2629 }
2630 }
2631
2632 // target
2633 func (gsh*GshContext)Trelpath(rloc string)(string){
2634     cwd, _ := os.Getwd()
2635     os.Chdir(gsh.RWD)
2636     os.Chdir(rloc)
2637     twd, _ := os.Getwd()
2638     os.Chdir(cwd)
2639
2640     tpath := twd + "/" + rloc
2641     return tpath
2642 }
2643 // join to rremote GShell - [user@]host[:port] or cd host[:port]:path
2644 func (gsh*GshContext)Rjoin(argv[]string){
2645     if len(argv) <= 1 {
2646         fmt.Printf("--I-- current server = %v\n",gsh.RSERV)
2647         return
2648     }
2649     serv := argv[1]
2650     servv := strings.Split(serv,":")
2651     if 1 <= len(servv) {
2652         if servv[0] == "lo" {
2653             servv[0] = "localhost"
2654         }
2655     }
2656     switch len(servv) {
2657     case 1:
2658         //if strings.Index(serv,":") < 0 {
2659             serv = servv[0] + ":" + fmt.Sprintf("%d",GSH_PORT)
2660         //}
2661     case 2: // host:port
2662         serv = strings.Join(servv,":")
2663     }
2664     xargv := []string{"rex-join", "@"+serv,"HELO"}
2665     rcode,stat := gsh.RexecClient(xargv)
2666     if (rcode / 100) == 2 {
2667         fmt.Printf("--I-- OK Joined (%v) %v\n",rcode,stat)
2668         gsh.RSERV = serv
2669     }else{
2670         fmt.Printf("--I-- NG, could not joined (%v) %v\n",rcode,stat)
2671     }
2672 }
2673 func (gsh*GshContext)Rexec(argv[]string){
2674     if len(argv) <= 1 {
2675         fmt.Printf("--I-- rexec command [ | {file || {command} ]\n",gsh.RSERV)
2676         return
2677     }
2678 }
2679 /*
2680 nargv := gshScanArg(strings.Join(argv, " "),0)
2681 fmt.Printf("--D-- nargc=%d [%v]\n",len(nargv),nargv)
2682 if nargv[1][0] != '{' {
2683     nargv[1] = "{" + nargv[1] + "}"
2684     fmt.Printf("--D-- nargc=%d [%v]\n",len(nargv),nargv)
2685 }
2686 argv = nargv
2687 */
2688 nargv := []string{}
2689 argv = append(nargv,"{"+strings.Join(argv[1:]," ")+"}")
2690 fmt.Printf("--D-- nargc=%d %v\n",len(nargv),nargv)
2691 argv = nargv
2692
2693 xargv := []string{"rex-exec", "@"+gsh.RSERV,"GET"}
2694 xargv = append(xargv,argv...)
2695 xargv = append(xargv,"/dev/tty")
2696 rcode,stat := gsh.RexecClient(xargv)
2697 if (rcode / 100) == 2 {
2698     fmt.Printf("--I-- OK Rexec (%v) %v\n",rcode,stat)
2699 }else{
2700     fmt.Printf("--I-- NG Rexec (%v) %v\n",rcode,stat)
2701 }
2702 }
2703 func (gsh*GshContext)Rchdir(argv[]string){
2704     if len(argv) <= 1 {
2705         return
2706     }
2707     cwd, _ := os.Getwd()
2708     os.Chdir(gsh.RWD)
2709     os.Chdir(argv[1])
2710     twd, _ := os.Getwd()
2711     gsh.RWD = twd
2712     fmt.Printf("--I-- JWD=%v\n",twd)
2713     os.Chdir(cwd)
2714 }
2715 func (gsh*GshContext)Rpwd(argv[]string){
2716     fmt.Printf("%v\n",gsh.RWD)
2717 }
2718 func (gsh*GshContext)Rls(argv[]string){
2719     cwd, _ := os.Getwd()
2720     os.Chdir(gsh.RWD)
2721     argv[0] = "-ls"
2722     gsh.xFind(argv)
2723     os.Chdir(cwd)
2724 }
2725 func (gsh*GshContext)Rput(argv[]string){
2726     var local string = ""
2727     var remote string = ""
2728     if 1 < len(argv) {

```

```

2729     local = argv[1]
2730     remote = local // base name
2731 }
2732 if 2 < len(argv) {
2733     remote = argv[2]
2734 }
2735 fmt.Printf("--I-- jput from=%v to=%v\n",local,gsh.Trelpath(remote))
2736 }
2737 func (gsh*GshContext)Rget(argv []string){
2738     var remote string = ""
2739     var local string = ""
2740     if 1 < len(argv) {
2741         remote = argv[1]
2742         local = remote // base name
2743     }
2744     if 2 < len(argv) {
2745         local = argv[2]
2746     }
2747     fmt.Printf("--I-- jget from=%v to=%v\n",gsh.Trelpath(remote),local)
2748 }
2749 //
2750 // <a name="network">network</a>
2751 // -s, -si, -so // bi-directional, source, sync (maybe socket)
2752 func (gshCtx*GshContext)sconnect(inTCP bool, argv []string) {
2753     gshPA := gshCtx.gshPA
2754     if len(argv) < 2 {
2755         fmt.Printf("Usage: -s [host]:[port[.udp]]\n")
2756         return
2757     }
2758     remote := argv[1]
2759     if remote == ":" { remote = "0.0.0.0:9999" }
2760
2761     if inTCP { // TCP
2762         dport, err := net.ResolveTCPAddr("tcp",remote);
2763         if err != nil {
2764             fmt.Printf("Address error: %s (%s)\n",remote,err)
2765             return
2766         }
2767         conn, err := net.DialTCP("tcp",nil,dport)
2768         if err != nil {
2769             fmt.Printf("Connection error: %s (%s)\n",remote,err)
2770             return
2771         }
2772         file, _ := conn.File();
2773         fd := file.Fd()
2774         fmt.Printf("Socket: connected to %s, socket[%d]\n",remote,fd)
2775
2776         savfd := gshPA.Files[1]
2777         gshPA.Files[1] = fd;
2778         gshCtx.gshellv(argv[2:])
2779         gshPA.Files[1] = savfd
2780         file.Close()
2781         conn.Close()
2782     }else{
2783         //dport, err := net.ResolveUDPAddr("udp4",remote);
2784         dport, err := net.ResolveUDPAddr("udp",remote);
2785         if err != nil {
2786             fmt.Printf("Address error: %s (%s)\n",remote,err)
2787             return
2788         }
2789         //conn, err := net.DialUDP("udp4",nil,dport)
2790         conn, err := net.DialUDP("udp",nil,dport)
2791         if err != nil {
2792             fmt.Printf("Connection error: %s (%s)\n",remote,err)
2793             return
2794         }
2795         file, _ := conn.File();
2796         fd := file.Fd()
2797
2798         ar := conn.RemoteAddr()
2799         //al := conn.LocalAddr()
2800         fmt.Printf("Socket: connected to %s [%s], socket[%d]\n",
2801             remote,ar.String(),fd)
2802
2803         savfd := gshPA.Files[1]
2804         gshPA.Files[1] = fd;
2805         gshCtx.gshellv(argv[2:])
2806         gshPA.Files[1] = savfd
2807         file.Close()
2808         conn.Close()
2809     }
2810 }
2811 func (gshCtx*GshContext)saccept(inTCP bool, argv []string) {
2812     gshPA := gshCtx.gshPA
2813     if len(argv) < 2 {
2814         fmt.Printf("Usage: -ac [host]:[port[.udp]]\n")
2815         return
2816     }
2817     local := argv[1]
2818     if local == ":" { local = "0.0.0.0:9999" }
2819     if inTCP { // TCP
2820         port, err := net.ResolveTCPAddr("tcp",local);
2821         if err != nil {
2822             fmt.Printf("Address error: %s (%s)\n",local,err)
2823             return
2824         }
2825         //fmt.Printf("Listen at %s...\n",local);
2826         sconn, err := net.ListenTCP("tcp", port)
2827         if err != nil {
2828             fmt.Printf("Listen error: %s (%s)\n",local,err)
2829             return
2830         }
2831         //fmt.Printf("Accepting at %s...\n",local);
2832         aconn, err := sconn.AcceptTCP()
2833         if err != nil {
2834             fmt.Printf("Accept error: %s (%s)\n",local,err)
2835             return
2836         }
2837         file, _ := aconn.File()
2838         fd := file.Fd()
2839         fmt.Printf("Accepted TCP at %s [%d]\n",local,fd)
2840
2841         savfd := gshPA.Files[0]
2842         gshPA.Files[0] = fd;
2843         gshCtx.gshellv(argv[2:])
2844         gshPA.Files[0] = savfd
2845
2846         sconn.Close();
2847         aconn.Close();
2848         file.Close();
2849     }else{
2850         //port, err := net.ResolveUDPAddr("udp4",local);
2851         port, err := net.ResolveUDPAddr("udp",local);
2852         if err != nil {

```

```

2853     fmt.Printf("Address error: %s (%s)\n",local,err)
2854     return
2855 }
2856 fmt.Printf("Listen UDP at %s...\n",local);
2857 //uconn, err := net.ListenUDP("udp4", port)
2858 uconn, err := net.ListenUDP("udp", port)
2859 if err != nil {
2860     fmt.Printf("Listen error: %s (%s)\n",local,err)
2861     return
2862 }
2863 file, _ := uconn.File()
2864 fd := file.Fd()
2865 ar := uconn.RemoteAddr()
2866 remote := ""
2867 if ar != nil { remote = ar.String() }
2868 if remote == "" { remote = "?" }
2869
2870 // not yet received
2871 //fmt.Printf("Accepted at %s [%d] <- %s\n",local,fd,"")
2872
2873 savfd := gshPA.Files[0]
2874 gshPA.Files[0] = fd;
2875 savenv := gshPA.Env
2876 gshPA.Env = append(savenv, "REMOTE_HOST="+remote)
2877 gshCtx.gshellv(argv[2:])
2878 gshPA.Env = savenv
2879 gshPA.Files[0] = savfd
2880
2881 uconn.Close();
2882 file.Close();
2883 }
2884 }
2885
2886 // empty line command
2887 func (gshCtx*GshContext)xPwd(argv[]string){
2888     // execute context command, pwd + date
2889     // context notation, representation scheme, to be resumed at re-login
2890     cwd, _ := os.Getwd()
2891     switch {
2892     case isin("-a",argv):
2893         gshCtx.ShowChdirHistory(argv)
2894     case isin("-ls",argv):
2895         showFileInfo(cwd,argv)
2896     default:
2897         fmt.Printf("%s\n",cwd)
2898     case isin("-v",argv): // obsolete empty command
2899         t := time.Now()
2900         date := t.Format(time.UnixDate)
2901         exe, _ := os.Executable()
2902         host, _ := os.Hostname()
2903         fmt.Printf(" PWD=\"%s\"",cwd)
2904         fmt.Printf(" HOST=\"%s\"",host)
2905         fmt.Printf(" DATE=\"%s\"",date)
2906         fmt.Printf(" TIME=\"%s\"",t.String())
2907         fmt.Printf(" PID=\"%s\"",os.Getpid())
2908         fmt.Printf(" EXE=\"%s\"",exe)
2909         fmt.Printf("\n")
2910     }
2911 }
2912
2913 // <a name="history">History</a>
2914 // these should be browsed and edited by HTTP browser
2915 // show the time of command with -t and direcotry with -ls
2916 // openfile-history, sort by -a -m -c
2917 // sort by elapsed time by -t -s
2918 // search by "more" like interface
2919 // edit history
2920 // sort history, and wc or uniq
2921 // CPU and other resource consumptions
2922 // limit showing range (by time or so)
2923 // export / import history
2924 func (gshCtx *GshContext)xHistory(argv []string){
2925     atWorkDirX := -1
2926     if 1 < len(argv) && strBegins(argv[1],"0") {
2927         atWorkDirX,_ = strconv.Atoi(argv[1][1:])
2928     }
2929     //fmt.Printf("--D-- showHistory(%v)\n",argv)
2930     for i, v := range gshCtx.CommandHistory {
2931         // exclude commands not to be listed by default
2932         // internal commands may be suppressed by default
2933         if v.CmdLine == "" && !isin("-a",argv) {
2934             continue;
2935         }
2936         if 0 <= atWorkDirX {
2937             if v.WorkDirX != atWorkDirX {
2938                 continue
2939             }
2940         }
2941         if !isin("-n",argv){ // like "fc"
2942             fmt.Printf("%-2d ",i)
2943         }
2944         if isin("-v",argv){
2945             fmt.Println(v) // should be with it date
2946         }else{
2947             if isin("-l",argv) || isin("-l0",argv) {
2948                 elps := v.EndAt.Sub(v.StartAt);
2949                 start := v.StartAt.Format(time.Stamp)
2950                 fmt.Printf("@%d ",v.WorkDirX)
2951                 fmt.Printf("[%v] %11v/t ",start,elps)
2952             }
2953             if isin("-l",argv) && !isin("-l0",argv){
2954                 fmt.Printf("%v",Rusagef("%t %u\t// %s",argv,v.Rusagev))
2955             }
2956             if isin("-at",argv) { // isin("-ls",argv){
2957                 dhi := v.WorkDirX // workdir history index
2958                 fmt.Printf("@%d %s\t",dhi,v.WorkDir)
2959                 // show the FileInfo of the output command??
2960             }
2961             fmt.Printf("%s",v.CmdLine)
2962             fmt.Printf("\n")
2963         }
2964     }
2965 }
2966 // ln - history index
2967 func searchHistory(gshCtx GshContext, gline string) (string, bool, bool){
2968     if gline[0] == '!' {
2969         hix, err := strconv.Atoi(gline[1:])
2970         if err != nil {
2971             fmt.Printf("--E-- (%s : range)\n",hix)
2972             return "", false, true
2973         }
2974         if hix < 0 || len(gshCtx.CommandHistory) <= hix {
2975             fmt.Printf("--E-- (%d : out of range)\n",hix)
2976             return "", false, true

```



```

2977     }
2978     return gshCtx.CommandHistory[hix].CmdLine, false, false
2979 }
2980 // search
2981 //for i, v := range gshCtx.CommandHistory {
2982 //}
2983 return gline, false, false
2984 }
2985 func (gsh*GshContext)cmdStringInHistory(hix int)(cmd string, ok bool){
2986     if 0 <= hix && hix < len(gsh.CommandHistory) {
2987         return gsh.CommandHistory[hix].CmdLine,true
2988     }
2989     return "",false
2990 }
2991
2992 // temporary adding to PATH environment
2993 // cd name -lib for LD_LIBRARY_PATH
2994 // chdir with directory history (date + full-path)
2995 // -s for sort option (by visit date or so)
2996 func (gsh*GshContext)ShowChdirHistory1(i int,v GChdirHistory, argv []string){
2997     fmt.Printf("%s-2d ",v.CmdIndex) // the first command at this WorkDir
2998     fmt.Printf("%sd ",i)
2999     fmt.Printf("[%v] ",v.MovedAt.Format(time.Stamp))
3000     showFileInfo(v.Dir,argv)
3001 }
3002 func (gsh*GshContext)ShowChdirHistory(argv []string){
3003     for i, v := range gsh.ChdirHistory {
3004         gsh.ShowChdirHistory1(i,v,argv)
3005     }
3006 }
3007 func skipOpts(argv[]string)(int){
3008     for i,v := range argv {
3009         if strBegins(v,"-") {
3010             }else{
3011                 return i
3012             }
3013         }
3014     }
3015     return -1
3016 }
3017 func (gshCtx*GshContext)xChdir(argv []string){
3018     cdhist := gshCtx.ChdirHistory
3019     if isin("?",argv) || isin("-t",argv) || isin("-a",argv) {
3020         gshCtx.ShowChdirHistory(argv)
3021         return
3022     }
3023     pwd, _ := os.Getwd()
3024     dir := ""
3025     if len(argv) <= 1 {
3026         dir = toFullpath("-")
3027     }else{
3028         i := skipOpts(argv[1:])
3029         if i < 0 {
3030             dir = toFullpath("-")
3031         }else{
3032             dir = argv[1+i]
3033         }
3034     }
3035     if strBegins(dir,"@") {
3036         if dir == "@0" { // obsolete
3037             dir = gshCtx.StartDir
3038         }else
3039         if dir == "@!" {
3040             index := len(cdhist) - 1
3041             if 0 < index { index -- 1 }
3042             dir = cdhist[index].Dir
3043         }else{
3044             index, err := stroconv.Atoi(dir[1:])
3045             if err != nil {
3046                 fmt.Printf("--E-- xChdir(%v)\n",err)
3047                 dir = "?"
3048             }else
3049             if len(gshCtx.ChdirHistory) <= index {
3050                 fmt.Printf("--E-- xChdir(history range error)\n")
3051                 dir = "?"
3052             }else{
3053                 dir = cdhist[index].Dir
3054             }
3055         }
3056     }
3057     if dir != "?" {
3058         err := os.Chdir(dir)
3059         if err != nil {
3060             fmt.Printf("--E-- xChdir(%s)(%v)\n",argv[1],err)
3061         }else{
3062             cwd, _ := os.Getwd()
3063             if cwd != pwd {
3064                 hist1 := GChdirHistory { }
3065                 hist1.Dir = cwd
3066                 hist1.MovedAt = time.Now()
3067                 hist1.CmdIndex = len(gshCtx.CommandHistory)+1
3068                 gshCtx.ChdirHistory = append(cdhist,hist1)
3069                 if !isin("-s",argv){
3070                     //cwd, _ := os.Getwd()
3071                     //fmt.Printf("%s\n",cwd)
3072                     ix := len(gshCtx.ChdirHistory)-1
3073                     gshCtx.ShowChdirHistory1(ix,hist1,argv)
3074                 }
3075             }
3076         }
3077     }
3078     if isin("-ls",argv){
3079         cwd, _ := os.Getwd()
3080         showFileInfo(cwd,argv);
3081     }
3082 }
3083 func TimeValSub(tv1 *syscall.Timeval, tv2 *syscall.Timeval){
3084     *tv1 = syscall.NsecToTimeval(tv1.Nano() - tv2.Nano())
3085 }
3086 func RusageSubv(ru1, ru2 [2]syscall.Rusage){([2]syscall.Rusage){
3087     TimeValSub(&ru1[0].Utime,&ru2[0].Utime)
3088     TimeValSub(&ru1[0].Stime,&ru2[0].Stime)
3089     TimeValSub(&ru1[1].Utime,&ru2[1].Utime)
3090     TimeValSub(&ru1[1].Stime,&ru2[1].Stime)
3091     return ru1
3092 }
3093 func TimeValAdd(tv1 syscall.Timeval, tv2 syscall.Timeval)(syscall.Timeval){
3094     tvs := syscall.NsecToTimeval(tv1.Nano() + tv2.Nano())
3095     return tvs
3096 }
3097 /*
3098 func RusageAddv(ru1, ru2 [2]syscall.Rusage){([2]syscall.Rusage){
3099     TimeValAdd(ru1[0].Utime,ru2[0].Utime)
3100     TimeValAdd(ru1[0].Stime,ru2[0].Stime)
3101     TimeValAdd(ru1[1].Utime,ru2[1].Utime)

```

```

3101     TimeValAdd(ru1[1].Stime,ru2[1].Stime)
3102     return ru1
3103 }
3104 */
3105
3106 // <a name="rusage">Resource Usage</a>
3107 func sRusagef(fmtspec string, argv []string, ru [2]syscall.Rusage)(string){
3108     // ru[0] self , ru[1] children
3109     ut := TimeValAdd(ru[0].Utime,ru[1].Utime)
3110     st := TimeValAdd(ru[0].Stime,ru[1].Stime)
3111     uu := (ut.Sec*1000000 + int64(ut.Usec)) * 1000
3112     su := (st.Sec*1000000 + int64(st.Usec)) * 1000
3113     tu := uu + su
3114     ret := fmt.Sprintf("%v/sum",abftime(tu))
3115     ret += fmt.Sprintf(" , %v/usr",abftime(uu))
3116     ret += fmt.Sprintf(" , %v/sys",abftime(su))
3117     return ret
3118 }
3119 func Rusagef(fmtspec string, argv []string, ru [2]syscall.Rusage)(string){
3120     ut := TimeValAdd(ru[0].Utime,ru[1].Utime)
3121     st := TimeValAdd(ru[0].Stime,ru[1].Stime)
3122     fmt.Printf("%d.%06ds/u ",ut.Sec,ut.Usec) //ru[1].Utime.Sec,ru[1].Utime.Usec)
3123     fmt.Printf("%d.%06ds/s ",st.Sec,st.Usec) //ru[1].Stime.Sec,ru[1].Stime.Usec)
3124     return ""
3125 }
3126 func Getrusagev()([2]syscall.Rusage){
3127     var ruv = [2]syscall.Rusage{
3128         syscall.Getrusage(syscall.RUSAGE_SELF,&ruv[0])
3129         syscall.Getrusage(syscall.RUSAGE_CHILDREN,&ruv[1])
3130     }
3131     return ruv
3132 }
3133 func showRusage(what string,argv []string, ru *syscall.Rusage){
3134     fmt.Printf("%s: ",what);
3135     fmt.Printf("Usr=%d.%06ds",ru.Utime.Sec,ru.Utime.Usec)
3136     fmt.Printf(" Sys=%d.%06ds",ru.Stime.Sec,ru.Stime.Usec)
3137     fmt.Printf(" Rss=%vB",ru.Maxrss)
3138     if isin("-l",argv) {
3139         fmt.Printf(" MinFlt=%v",ru.Minflt)
3140         fmt.Printf(" MajFlt=%v",ru.Majflt)
3141         fmt.Printf(" IxRSS=%vB",ru.Ixrss)
3142         fmt.Printf(" IdRSS=%vB",ru.Idrss)
3143         fmt.Printf(" Nswap=%vB",ru.Nswap)
3144         fmt.Printf(" Read=%v",ru.Inblock)
3145         fmt.Printf(" Write=%v",ru.Oblock)
3146     }
3147     fmt.Printf(" Snd=%v",ru.Msgsnd)
3148     fmt.Printf(" Rcv=%v",ru.Msgrcv)
3149     //if isin("-l",argv) {
3150         fmt.Printf(" Sig=%v",ru.Nsignals)
3151     //}
3152     fmt.Printf("\n");
3153 }
3154 func (gshCtx *GshContext)xTime(argv[]string)(bool){
3155     if 2 <= len(argv){
3156         gshCtx.LastRusage = syscall.Rusage{
3157             rusagev1 := Getrusagev()
3158             fin := gshCtx.gshellv(argv[1:])
3159             rusagev2 := Getrusagev()
3160             showRusage(argv[1],argv,&gshCtx.LastRusage)
3161             rusagev := RusageSubv(rusagev2,rusagev1)
3162             showRusage("self",argv,&rusagev[0])
3163             showRusage("chld",argv,&rusagev[1])
3164             return fin
3165         }else{
3166             rusage:= syscall.Rusage {
3167                 syscall.Getrusage(syscall.RUSAGE_SELF,&rusage)
3168                 showRusage("self",argv, &rusage)
3169                 syscall.Getrusage(syscall.RUSAGE_CHILDREN,&rusage)
3170                 showRusage("chld",argv, &rusage)
3171             }
3172             return false
3173         }
3174 }
3175 func (gshCtx *GshContext)xJobs(argv[]string){
3176     fmt.Printf("%d Jobs\n",len(gshCtx.BackGroundJobs))
3177     for ji, pid := range gshCtx.BackGroundJobs {
3178         //wstat := syscall.WaitStatus {0}
3179         rusage := syscall.Rusage {0}
3180         //wpid, err := syscall.Wait4(pid,&wstat,syscall.WNOHANG,&rusage);
3181         wpid, err := syscall.Wait4(pid,nil,syscall.WNOHANG,&rusage);
3182         if err != nil {
3183             fmt.Printf("--E-- %%d [%d] (%v)\n",ji,pid,err)
3184         }else{
3185             fmt.Printf("%%d[%d] (%d)\n",ji,pid,wpid)
3186             showRusage("chld",argv,&rusage)
3187         }
3188     }
3189 }
3190 func (gshCtx *GshContext)inBackground(argv[]string)(bool){
3191     if gsh.CmdTrace { fmt.Printf("--I-- inBackground(%v)\n",argv) }
3192     gsh.BackGround = true // set background option
3193     xfin := false
3194     xfin = gsh.gshellv(argv)
3195     gsh.BackGround = false
3196     return xfin
3197 }
3198 // -o file without command means just opening it and refer by #N
3199 // should be listed by "files" command
3200 func (gshCtx *GshContext)xOpen(argv[]string){
3201     var pv = []int{-1,-1}
3202     err := syscall.Pipe(pv)
3203     fmt.Printf("--I-- pipe()=#%d,%#d] (%v)\n",pv[0],pv[1],err)
3204 }
3205 func (gshCtx *GshContext)fromPipe(argv[]string){
3206 }
3207 func (gshCtx *GshContext)xClose(argv[]string){
3208 }
3209 // <a name="redirect">redirect</a>
3210 func (gshCtx *GshContext)redirect(argv[]string)(bool){
3211     if len(argv) < 2 {
3212         return false
3213     }
3214     cmd := argv[0]
3215     fname := argv[1]
3216     var file *os.File = nil
3217     fdix := 0
3218     mode := os.O_RDONLY
3219     switch {
3220     case cmd == "-i" || cmd == "<":
3221         fdix = 0
3222         mode = os.O_RDONLY
3223     }
3224 }

```

```

3225     case cmd == "-o" || cmd == ">":
3226         fdix = 1
3227         mode = os.O_RDWR | os.O_CREATE
3228     case cmd == "-a" || cmd == ">>":
3229         fdix = 1
3230         mode = os.O_RDWR | os.O_CREATE | os.O_APPEND
3231     }
3232     if fname[0] == '#' {
3233         fd, err := strconv.Atoi(fname[1:])
3234         if err != nil {
3235             fmt.Printf("--E-- (%v)\n",err)
3236             return false
3237         }
3238         file = os.NewFile(uintptr(fd),"MaybePipe")
3239     }else{
3240         xfile, err := os.OpenFile(argv[1], mode, 0600)
3241         if err != nil {
3242             fmt.Printf("--E-- (%s)\n",err)
3243             return false
3244         }
3245         file = xfile
3246     }
3247     gshPA := gshCtx.gshPA
3248     savfd := gshPA.Files[fdix]
3249     gshPA.Files[fdix] = file.Fd()
3250     fmt.Printf("--I-- Opened [%d] %s\n",file.Fd(),argv[1])
3251     gshCtx.gshellv(argv[2:])
3252     gshPA.Files[fdix] = savfd
3253
3254     return false
3255 }
3256
3257 //fmt.Fprintf(res, "GShell Status: %q", html.EscapeString(req.URL.Path))
3258 func httpHandler(res http.ResponseWriter, req *http.Request){
3259     path := req.URL.Path
3260     fmt.Printf("--I-- Got HTTP Request(%s)\n",path)
3261     {
3262         gshCtxBuf, _ := setupGshContext()
3263         gshCtx := *gshCtxBuf
3264         fmt.Printf("--I-- %s\n",path[1:])
3265         gshCtx.tgshelll(path[1:])
3266     }
3267     fmt.Fprintf(res, "Hello(^-^)/\n%s\n",path)
3268 }
3269 func (gshCtx *GshContext) httpServer(argv []string){
3270     http.HandleFunc("/", httpHandler)
3271     accport := "localhost:9999"
3272     fmt.Printf("--I-- HTTP Server Start at [%s]\n",accport)
3273     http.ListenAndServe(accport,nil)
3274 }
3275 func (gshCtx *GshContext)xGo(argv[]string){
3276     go gshCtx.gshellv(argv[1:]);
3277 }
3278 func (gshCtx *GshContext) xPs(argv[]string){}
3279 }
3280
3281 // <a name="plugin">Plugin</a>
3282 // plugin [-ls [names]] to list plugins
3283 // Reference: <a href="https://golang.org/src/plugin/">plugin</a> source code
3284 func (gshCtx *GshContext) whichPlugin(name string,argv[]string)(pi *PluginInfo){
3285     pi = nil
3286     for _,p := range gshCtx.PluginFuncs {
3287         if p.Name == name && pi == nil {
3288             pi = &p
3289         }
3290         if !isin("-s",argv){
3291             //fmt.Printf("%v %v ",i,p)
3292             if isin("-ls",argv){
3293                 showFileInfo(p.Path,argv)
3294             }else{
3295                 fmt.Printf("%s\n",p.Name)
3296             }
3297         }
3298     }
3299     return pi
3300 }
3301 func (gshCtx *GshContext) xPlugin(argv[]string) (error) {
3302     if len(argv) == 0 || argv[0] == "-ls" {
3303         gshCtx.whichPlugin("",argv)
3304         return nil
3305     }
3306     name := argv[0]
3307     Pin := gshCtx.whichPlugin(name,[]string{"-s"})
3308     if Pin != nil {
3309         os.Args = argv // should be recovered?
3310         Pin.Addr.(func())()
3311         return nil
3312     }
3313     sofile := toFullpath(argv[0] + ".so") // or find it by which($PATH)
3314
3315     p, err := plugin.Open(sofile)
3316     if err != nil {
3317         fmt.Printf("--E-- plugin.Open(%s)(%v)\n",sofile,err)
3318         return err
3319     }
3320     fname := "Main"
3321     f, err := p.Lookup(fname)
3322     if( err != nil ){
3323         fmt.Printf("--E-- plugin.Lookup(%s)(%v)\n",fname,err)
3324         return err
3325     }
3326     pin := PluginInfo {p,f,name,sofile}
3327     gshCtx.PluginFuncs = append(gshCtx.PluginFuncs,pin)
3328     fmt.Printf("--I-- added (%d)\n",len(gshCtx.PluginFuncs))
3329
3330     //fmt.Printf("--I-- first call(%s:%s)%v\n",sofile,fname,argv)
3331     os.Args = argv
3332     f.(func())()
3333     return err
3334 }
3335 func (gshCtx*GshContext)Args(argv[]string){
3336     for i,v := range os.Args {
3337         fmt.Printf("[%v] %v\n",i,v)
3338     }
3339 }
3340 func (gshCtx *GshContext) showVersion(argv[]string){
3341     if isin("-l",argv) {
3342         fmt.Printf("%v/%v (%v)",NAME,VERSION,DATE);
3343     }else{
3344         fmt.Printf("%v",VERSION);
3345     }
3346     if isin("-a",argv) {
3347         fmt.Printf("%s",AUTHOR)
3348     }

```

```

3349     if !isin("-n",argv) {
3350         fmt.Printf("\n")
3351     }
3352 }
3353
3354 // <a name="scanf">Scanf</a> // string decomposer
3355 // scanf [format] [input]
3356 func scanv(sstr string)(strv[]string){
3357     strv = strings.Split(sstr, " ")
3358     return strv
3359 }
3360 func scanUntil(src,end string)(rstr string,leng int){
3361     idx := strings.Index(src,end)
3362     if 0 <= idx {
3363         rstr = src[0:idx]
3364         return rstr,idx+leng(end)
3365     }
3366     return src,0
3367 }
3368
3369 // -bn -- display base-name part only // can be in some %fmt, for sed rewriting
3370 func (gsh*GshContext)printVal(fmts string, vstr string, optv[]string){
3371     //vint,err := strconv.Atoi(vstr)
3372     var ival int64 = 0
3373     n := 0
3374     err := error(nil)
3375     if strBegins(vstr,"") {
3376         vx,_ := strconv.Atoi(vstr[1:])
3377         if vx < len(gsh.iValues) {
3378             vstr = gsh.iValues[vx]
3379         }else{
3380         }
3381     }
3382     // should use Eval()
3383     if strBegins(vstr,"0x") {
3384         n,err = fmt.Sscanf(vstr[2:],"%x",&ival)
3385     }else{
3386         n,err = fmt.Sscanf(vstr,"%d",&ival)
3387     }//fmt.Printf("--D-- n=%d err=(%v) (%s)=%v\n",n,err,vstr, ival)
3388 }
3389 if n == 1 && err == nil {
3390     //fmt.Printf("--D-- formatn(%v) ival(%v)\n",fmts,ival)
3391     fmt.Printf("%"+fmts,ival)
3392 }else{
3393     if isin("-bn",optv){
3394         fmt.Printf("%"+fmts,filepath.Base(vstr))
3395     }else{
3396         fmt.Printf("%"+fmts,vstr)
3397     }
3398 }
3399 }
3400 func (gsh*GshContext)printfv(fmts,div string,argv[]string,optv[]string,list[]string){
3401     //fmt.Printf("{%d}",len(list))
3402     //curfmt := "v"
3403     outlen := 0
3404     curfmt := gsh.iFormat
3405
3406     if 0 < len(fmts) {
3407         for xi := 0; xi < len(fmts); xi++ {
3408             fch := fmts[xi]
3409             if fch == '%' {
3410                 if xi+1 < len(fmts) {
3411                     curfmt = string(fmts[xi+1])
3412                 }
3413                 xi += 1
3414             }
3415             if xi+1 < len(fmts) && fmts[xi+1] == '(' {
3416                 vals,leng := scanUntil(fmts[xi+2:],")")
3417                 //fmt.Printf("--D-- show fmt(%v) val(%v) next(%v)\n",curfmt,vals,leng)
3418                 gsh.printVal(curfmt,vals,optv)
3419                 xi += 2+leng-1
3420                 outlen += 1
3421             }
3422             continue
3423         }
3424         if fch == '-' {
3425             hi,leng := scanInt(fmts[xi+1:])
3426             if 0 < leng {
3427                 if hi < len(gsh.iValues) {
3428                     gsh.printVal(curfmt,gsh.iValues[hi],optv)
3429                     outlen += 1 // should be the real length
3430                 }else{
3431                     fmt.Printf("((out-range))")
3432                 }
3433                 xi += leng
3434                 continue;
3435             }
3436         }
3437         fmt.Printf("%c",fch)
3438         outlen += 1
3439     }
3440 }else{
3441     //fmt.Printf("--D-- print (%s)\n")
3442     for i,v := range list {
3443         if 0 < i {
3444             fmt.Printf(div)
3445         }
3446         gsh.printVal(curfmt,v,optv)
3447         outlen += 1
3448     }
3449 }
3450 if 0 < outlen {
3451     fmt.Printf("\n")
3452 }
3453 }
3454 func (gsh*GshContext)Scanv(argv[]string){
3455     //fmt.Printf("--D-- Scanv(%v)\n",argv)
3456     if len(argv) == 1 {
3457         return
3458     }
3459     argv = argv[1:]
3460     fmts := ""
3461     if strBegins(argv[0],"-F") {
3462         fmts = argv[0]
3463         gsh.iDelimiter = fmts
3464         argv = argv[1:]
3465     }
3466     input := strings.Join(argv, " ")
3467     if fmts == "" { // simple decomposition
3468         v := scanv(input)
3469         gsh.iValues = v
3470     }//fmt.Printf("%v\n",strings.Join(v,","))
3471 }else{
3472     v := make([]string,8)

```

```

3473     n,err := fmt.Sscanf(input,fmts,&v[0],&v[1],&v[2],&v[3])
3474     fmt.Printf("--D-- Sscanf ->(%v) n=%d err=(%v)\n",v,n,err)
3475     gsh.iValues = v
3476 }
3477 }
3478 func (gsh*GshContext)Printv(argv[]string){
3479     if false { //&#0U
3480         fmt.Printf("%v\n",strings.Join(argv[1:], " "))
3481         return
3482     }
3483     //fmt.Printf("--D-- Printv(%v)\n",argv)
3484     //fmt.Printf("%v\n",strings.Join(gsh.iValues," "))
3485     div := gsh.iDelimiter
3486     fmts := ""
3487     argv = argv[1:]
3488     if 0 < len(argv) {
3489         if strBegins(argv[0],"-F") {
3490             div = argv[0][2:]
3491             argv = argv[1:]
3492         }
3493     }
3494 }
3495 optv := []string{}
3496 for _,v := range argv {
3497     if strBegins(v,"-"){
3498         optv = append(optv,v)
3499         argv = argv[1:]
3500     }else{
3501         break;
3502     }
3503 }
3504 if 0 < len(argv) {
3505     fmts = strings.Join(argv, " ")
3506 }
3507 gsh.printfv(fmts,div,argv,optv,gsh.iValues)
3508 }
3509 func (gsh*GshContext)Basename(argv[]string){
3510     for i,v := range gsh.iValues {
3511         gsh.iValues[i] = filepath.Base(v)
3512     }
3513 }
3514 func (gsh*GshContext)Sortv(argv[]string){
3515     sv := gsh.iValues
3516     sort.Slice(sv , func(i,j int) bool {
3517         return sv[i] < sv[j]
3518     })
3519 }
3520 func (gsh*GshContext)Shiftv(argv[]string){
3521     vi := len(gsh.iValues)
3522     if 0 < vi {
3523         if isin("-r",argv) {
3524             top := gsh.iValues[0]
3525             gsh.iValues = append(gsh.iValues[1:],top)
3526         }else{
3527             gsh.iValues = gsh.iValues[1:]
3528         }
3529     }
3530 }
3531 }
3532 func (gsh*GshContext)Enq(argv[]string){
3533 }
3534 func (gsh*GshContext)Deq(argv[]string){
3535 }
3536 func (gsh*GshContext)Push(argv[]string){
3537     gsh.iValStack = append(gsh.iValStack,argv[1:])
3538     fmt.Printf("depth=%d\n",len(gsh.iValStack))
3539 }
3540 func (gsh*GshContext)Dump(argv[]string){
3541     for i,v := range gsh.iValStack {
3542         fmt.Printf("%d %v\n",i,v)
3543     }
3544 }
3545 func (gsh*GshContext)Pop(argv[]string){
3546     depth := len(gsh.iValStack)
3547     if 0 < depth {
3548         v := gsh.iValStack[depth-1]
3549         if isin("-cat",argv){
3550             gsh.iValues = append(gsh.iValues,v...)
3551         }else{
3552             gsh.iValues = v
3553         }
3554         gsh.iValStack = gsh.iValStack[0:depth-1]
3555         fmt.Printf("depth=%d %s\n",len(gsh.iValStack),gsh.iValues)
3556     }else{
3557         fmt.Printf("depth=%d\n",depth)
3558     }
3559 }
3560 }
3561 // <a name="interpreter">Command Interpreter</a>
3562 func (gshCtx*GshContext)gshellv(argv []string) (fin bool) {
3563     fin = false
3564 }
3565 if gshCtx.CmdTrace { fmt.Fprintf(os.Stderr,"--I-- gshellv(%d)\n",len(argv)) }
3566 if len(argv) <= 0 {
3567     return false
3568 }
3569 }
3570 xargv := []string{}
3571 for ai := 0; ai < len(argv); ai++ {
3572     xargv = append(xargv, strsubst(gshCtx,argv[ai],false))
3573 }
3574 argv = xargv
3575 if false {
3576     for ai := 0; ai < len(argv); ai++ {
3577         fmt.Printf("[%d] %s [%d]%T\n",
3578             ai,argv[ai],len(argv[ai]),argv[ai])
3579     }
3580 }
3581 cmd := argv[0]
3582 if gshCtx.CmdTrace { fmt.Fprintf(os.Stderr,"--I-- gshellv(%d)%v\n",len(argv),argv) }
3583 switch { // https://tour.golang.org/flowcontrol/11
3584 case cmd == "":
3585     gshCtx.xPwd([]string{}); // empty command
3586 case cmd == "-x":
3587     gshCtx.CmdTrace = ! gshCtx.CmdTrace
3588 case cmd == "-xt":
3589     gshCtx.CmdTime = ! gshCtx.CmdTime
3590 case cmd == "-ot":
3591     gshCtx.sconnect(true, argv)
3592 case cmd == "-ou":
3593     gshCtx.sconnect(false, argv)
3594 case cmd == "-it":
3595     gshCtx.saccept(true, argv)
3596 case cmd == "-iu":
3597     gshCtx.saccept(false, argv)

```

```

3597 case cmd == "-i" || cmd == "<" || cmd == "-o" || cmd == ">" || cmd == "-a" || cmd == ">>" || cmd == "-s" || cmd == "><":
3598     gshCtx.redirect(argv)
3599 case cmd == "|":
3600     gshCtx.fromPipe(argv)
3601 case cmd == "args":
3602     gshCtx.Args(argv)
3603 case cmd == "bg" || cmd == "-bg":
3604     rfin := gshCtx.inBackground(argv[1:])
3605     return rfin
3606 case cmd == "-bn":
3607     gshCtx.Basename(argv)
3608 case cmd == "call":
3609     _,_ = gshCtx.excommand(false,argv[1:])
3610 case cmd == "cd" || cmd == "chdir":
3611     gshCtx.xChdir(argv);
3612 case cmd == "-cksum":
3613     gshCtx.xFind(argv)
3614 case cmd == "-sum":
3615     gshCtx.xFind(argv)
3616 case cmd == "-sumtest":
3617     str := ""
3618     if 1 < len(argv) { str = argv[1] }
3619     crc := strCRC32(str,uint64(len(str)))
3620     fprintf(stderr,"%v %v\n",crc,len(str))
3621 case cmd == "close":
3622     gshCtx.xClose(argv)
3623 case cmd == "gcp":
3624     gshCtx.FileCopy(argv)
3625 case cmd == "dec" || cmd == "decode":
3626     gshCtx.Dec(argv)
3627 case cmd == "#define":
3628     case cmd == "dic" || cmd == "d":
3629         xDic(argv)
3630 case cmd == "dump":
3631     gshCtx.Dump(argv)
3632 case cmd == "echo" || cmd == "e":
3633     echo(argv,true)
3634 case cmd == "enc" || cmd == "encode":
3635     gshCtx.Enc(argv)
3636 case cmd == "env":
3637     env(argv)
3638 case cmd == "eval":
3639     xEval(argv[1:],true)
3640 case cmd == "ev" || cmd == "events":
3641     dumpEvents(argv)
3642 case cmd == "exec":
3643     _,_ = gshCtx.excommand(true,argv[1:])
3644     // should not return here
3645 case cmd == "exit" || cmd == "quit":
3646     // write Result code EXIT to 3>
3647     return true
3648 case cmd == "fdls":
3649     // dump the attributes of fds (of other process)
3650 case cmd == "-find" || cmd == "fin" || cmd == "ufind" || cmd == "uf":
3651     gshCtx.xFind(argv[1:])
3652 case cmd == "fx":
3653     gshCtx.xFind(argv[1:])
3654 case cmd == "fork":
3655     // mainly for a server
3656 case cmd == "-gen":
3657     gshCtx.gen(argv)
3658 case cmd == "-go":
3659     gshCtx.xGo(argv)
3660 case cmd == "-grep":
3661     gshCtx.xFind(argv)
3662 case cmd == "gdeg":
3663     gshCtx.Deq(argv)
3664 case cmd == "genq":
3665     gshCtx.Enq(argv)
3666 case cmd == "gpop":
3667     gshCtx.Pop(argv)
3668 case cmd == "gpush":
3669     gshCtx.Push(argv)
3670 case cmd == "history" || cmd == "hi": // hi should be alias
3671     gshCtx.xHistory(argv)
3672 case cmd == "jobs":
3673     gshCtx.xJobs(argv)
3674 case cmd == "lnsp" || cmd == "nlsp":
3675     gshCtx.SplitLine(argv)
3676 case cmd == "-ls":
3677     gshCtx.xFind(argv)
3678 case cmd == "nop":
3679     // do nothing
3680 case cmd == "pipe":
3681     gshCtx.xOpen(argv)
3682 case cmd == "plug" || cmd == "plugin" || cmd == "pin":
3683     gshCtx.xPlugin(argv[1:])
3684 case cmd == "print" || cmd == "-pr":
3685     // output internal slice // also sprintf should be
3686     gshCtx.Printv(argv)
3687 case cmd == "ps":
3688     gshCtx.xPs(argv)
3689 case cmd == "pstitle":
3690     // to be gsh.title
3691 case cmd == "rexecd" || cmd == "rexd":
3692     gshCtx.RexecServer(argv)
3693 case cmd == "rexec" || cmd == "rex":
3694     gshCtx.RexecClient(argv)
3695 case cmd == "repeat" || cmd == "rep": // repeat cond command
3696     gshCtx.repeat(argv)
3697 case cmd == "replay":
3698     gshCtx.xReplay(argv)
3699 case cmd == "scan":
3700     // scan input (or so in fscanf) to internal slice (like Files or map)
3701     gshCtx.Scanv(argv)
3702 case cmd == "set":
3703     // set name ...
3704 case cmd == "serv":
3705     gshCtx.httpServer(argv)
3706 case cmd == "shift":
3707     gshCtx.Shiftv(argv)
3708 case cmd == "sleep":
3709     gshCtx.sleep(argv)
3710 case cmd == "-sort":
3711     gshCtx.Sortv(argv)
3712
3713 case cmd == "j" || cmd == "join":
3714     gshCtx.Rjoin(argv)
3715 case cmd == "a" || cmd == "alpa":
3716     gshCtx.Rexec(argv)
3717 case cmd == "jcd" || cmd == "jchdir":
3718     gshCtx.Rchdir(argv)
3719 case cmd == "jget":
3720     gshCtx.Rget(argv)

```

```

3721 case cmd == "jls":
3722     gshCtx.Rls(argv)
3723 case cmd == "jput":
3724     gshCtx.Rput(argv)
3725 case cmd == "jpwd":
3726     gshCtx.Rpwd(argv)
3727
3728 case cmd == "time":
3729     fin = gshCtx.xTime(argv)
3730 case cmd == "ungets":
3731     if 1 < len(argv) {
3732         ungets(argv[1]+\n")
3733     }else{
3734     }
3735 case cmd == "pwd":
3736     gshCtx.xPwd(argv);
3737 case cmd == "ver" || cmd == "--ver" || cmd == "version":
3738     gshCtx.showVersion(argv)
3739 case cmd == "where":
3740     // data file or so?
3741 case cmd == "which":
3742     which("PATH",argv);
3743 case cmd == "gj" && 1 < len(argv) && argv[1] == "listen":
3744     go gj_server(argv[1]);
3745 case cmd == "gj" && 1 < len(argv) && argv[1] == "serve":
3746     go gj_server(argv[1]);
3747 case cmd == "gj" && 1 < len(argv) && argv[1] == "join":
3748     go gj_client(argv[1]);
3749 case cmd == "gj":
3750     jsend(argv);
3751 case cmd == "jsend":
3752     jsend(argv);
3753 default:
3754     if gshCtx.whichPlugin(cmd,[]string{"-s"}) != nil {
3755         gshCtx.xPlugin(argv)
3756     }else{
3757         notfound, _ := gshCtx.excommand(false,argv)
3758         if notfound {
3759             fmt.Printf("--E-- command not found (%v)\n",cmd)
3760         }
3761     }
3762 }
3763 return fin
3764 }
3765
3766 func (gsh*GshContext)gshelll(gline string) (rfin bool) {
3767     argv := strings.Split(string(gline)," ")
3768     fin := gsh.gshellv(argv)
3769     return fin
3770 }
3771 func (gsh*GshContext)tgshelll(gline string)(xfn bool){
3772     start := time.Now()
3773     fin := gsh.gshelll(gline)
3774     end := time.Now()
3775     elps := end.Sub(start);
3776     if gsh.CmdTime {
3777         fmt.Printf("--T-- " + time.Now().Format(time.Stamp) + " (%d.%09ds)\n",
3778             elps/1000000000,elps%1000000000)
3779     }
3780     return fin
3781 }
3782 func Ttyid() (int) {
3783     fi, err := os.Stdin.Stat()
3784     if err != nil {
3785         return 0;
3786     }
3787     //fmt.Printf("Stdin: %v Dev=%d\n",
3788     // fi.Mode(),fi.Mode()&os.ModeDevice)
3789     if (fi.Mode() & os.ModeDevice) != 0 {
3790         stat := syscall.Stat_t{};
3791         err := syscall.Fstat(0,&stat)
3792         if err != nil {
3793             //fmt.Printf("--I-- Stdin: (%v)\n",err)
3794         }else{
3795             //fmt.Printf("--I-- Stdin: rdev=%d %d\n",
3796             // stat.Rdev&0xFF,stat.Rdev);
3797             //fmt.Printf("--I-- Stdin: tty=%d\n",stat.Rdev&0xFF);
3798             return int(stat.Rdev & 0xFF)
3799         }
3800     }
3801     return 0
3802 }
3803 func (gshCtx *GshContext) ttyfile() string {
3804     //fmt.Printf("--I-- GSH_HOME=%s\n",gshCtx.GshHomeDir)
3805     ttyfile := gshCtx.GshHomeDir + "/" + "gsh-tty" +
3806         fmt.Sprintf("%02d",gshCtx.TerminalId)
3807     //strconv.Itoa(gshCtx.TerminalId)
3808     //fmt.Printf("--I-- ttyfile=%s\n",ttyfile)
3809     return ttyfile
3810 }
3811 func (gshCtx *GshContext) ttyline()(*os.File){
3812     file, err := os.OpenFile(gshCtx.ttyfile(),os.O_RDWR|os.O_CREATE|os.O_TRUNC,0600)
3813     if err != nil {
3814         fmt.Printf("--F-- cannot open %s (%s)\n",gshCtx.ttyfile(),err)
3815         return file;
3816     }
3817     return file
3818 }
3819 func (gshCtx *GshContext)getline(hix int, skipping bool, prevline string) (string) {
3820     if( skipping ){
3821         reader := bufio.NewReaderSize(os.Stdin,LINESIZE)
3822         line, _, _ := reader.ReadLine()
3823         return string(line)
3824     }else
3825     if true {
3826         return xgetline(hix,prevline,gshCtx)
3827     }
3828     /*
3829     else
3830     if( with_exgetline && gshCtx.GetLine != "" ){
3831         //var xhix int64 = int64(hix); // cast
3832         newenv := os.Environ()
3833         newenv = append(newenv, "GSH_LINENO="+strconv.FormatInt(int64(hix),10) )
3834
3835         tty := gshCtx.ttyline()
3836         tty.WriteString(prevline)
3837         Pa := os.ProcAttr {
3838             "", // start dir
3839             newenv, //os.Environ(),
3840             []*os.File{os.Stdin,os.Stdout,os.Stderr,tty},
3841             nil,
3842         }
3843         //fmt.Printf("--I-- getline=%s // %s\n",gsh_getlinev[0],gshCtx.GetLine)
3844         proc, err := os.StartProcess(gsh_getlinev[0],[]string{"getline","getline"},&Pa)

```

```

3845     if err != nil {
3846         fmt.Printf("--F-- getline process error (%v)\n",err)
3847         // for ; ; { }
3848         return "exit (getline program failed)"
3849     }
3850     //stat, err := proc.Wait()
3851     proc.Wait()
3852     buff := make([]byte,LINESIZE)
3853     count, err := tty.Read(buff)
3854     //_, err = tty.Read(buff)
3855     //fmt.Printf("--D-- getline (%d)\n",count)
3856     if err != nil {
3857         if ! (count == 0) { // && err.String() == "EOF" } {
3858             fmt.Printf("--E-- getline error (%s)\n",err)
3859         }
3860     }else{
3861         //fmt.Printf("--I-- getline OK \"%s\"\n",buff)
3862     }
3863     tty.Close()
3864     gline := string(buff[0:count])
3865     return gline
3866 }else
3867 /*
3868 {
3869     // if isatty {
3870     fmt.Printf("!%d",hix)
3871     fmt.Print(PROMPT)
3872     // }
3873     reader := bufio.NewReaderSize(os.Stdin,LINESIZE)
3874     line, _, _ := reader.ReadLine()
3875     return string(line)
3876 }
3877 }
3878
3879 //== begin ===== getline
3880 /*
3881 * getline.c
3882 * 2020-0819 extracted from dog.c
3883 * getline.go
3884 * 2020-0822 ported to Go
3885 */
3886 /*
3887 package main // getline main
3888 import (
3889     "fmt"           // <a href="https://golang.org/pkg/fmt/">fmt</a>
3890     "strings"      // <a href="https://golang.org/pkg/strings/">strings</a>
3891     "os"           // <a href="https://golang.org/pkg/os/">os</a>
3892     "syscall"     // <a href="https://golang.org/pkg/syscall/">syscall</a>
3893     "bytes"       // <a href="https://golang.org/pkg/bytes/">bytes</a>
3894     "os/exec"    // <a href="https://golang.org/pkg/os/exec/">os/exec</a>
3895 )
3896 */
3897 // C language compatibility functions
3898 var errno = 0
3899 var stdin *os.File = os.Stdin
3900 var stdout *os.File = os.Stdout
3901 var stderr *os.File = os.Stderr
3902 var EOF = -1
3903 var NULL = 0
3904 type FILE os.File
3905 type StrBuff []byte
3906 var NULL_FP *os.File = nil
3907 var NULLSP = 0
3908 //var LINESIZE = 1024
3909
3910 func system(cmdstr string)(int){
3911     PA := syscall.ProcAttr {
3912         "", // the starting directory
3913         os.Environ(),
3914         []uintptr{os.Stdin.Fd(),os.Stdout.Fd(),os.Stderr.Fd()},
3915         nil,
3916     }
3917     argv := strings.Split(cmdstr, " ")
3918     pid,err := syscall.ForkExec(argv[0],argv,&PA)
3919     if( err != nil ){
3920         fmt.Printf("--E-- syscall(%v) err(%v)\n",cmdstr,err)
3921     }
3922     syscall.Wait4(pid,nil,0,nil)
3923
3924     /*
3925     argv := strings.Split(cmdstr, " ")
3926     fmt.Fprintf(os.Stderr,"--I-- system(%v)\n",argv)
3927     //cmd := exec.Command(argv[0]:...)
3928     cmd := exec.Command(argv[0],argv[1],argv[2])
3929     cmd.Stdin = strings.NewReader("output of system")
3930     var out bytes.Buffer
3931     cmd.Stdout = &out
3932     var serr bytes.Buffer
3933     cmd.Stderr = &serr
3934     err := cmd.Run()
3935     if err != nil {
3936         fmt.Fprintf(os.Stderr,"--E-- system(%v)err(%v)\n",argv,err)
3937         fmt.Printf("ERR:%s\n",serr.String())
3938     }else{
3939         fmt.Printf("%s",out.String())
3940     }
3941     */
3942     return 0
3943 }
3944
3945 func atoi(str string)(ret int){
3946     ret,err := fmt.Sscanf(str,"%d",&ret)
3947     if err == nil {
3948         return ret
3949     }else{
3950         // should set errno
3951         return 0
3952     }
3953 }
3954 func getenv(name string)(string){
3955     val,got := os.LookupEnv(name)
3956     if got {
3957         return val
3958     }else{
3959         return "?"
3960     }
3961 }
3962 func strcpy(dst StrBuff, src string){
3963     var i int
3964     srcb := []byte(src)
3965     for i = 0; i < len(src) && srcb[i] != 0; i++ {
3966         dst[i] = srcb[i]
3967     }
3968     dst[i] = 0

```



```

3969 }
3970 func xstrcpy(dst StrBuff, src StrBuff){
3971     dst = src
3972 }
3973 func strcat(dst StrBuff, src StrBuff){
3974     dst = append(dst,src...)
3975 }
3976 func strdup(str StrBuff)(string){
3977     return string(str[0:strlen(str)])
3978 }
3979 func sstrlen(str string)(int){
3980     return len(str)
3981 }
3982 func strlen(str StrBuff)(int){
3983     var i int
3984     for i = 0; i < len(str) && str[i] != 0; i++ {
3985     }
3986     return i
3987 }
3988 func sizeof(data StrBuff)(int){
3989     return len(data)
3990 }
3991 func isatty(fd int)(ret int){
3992     return 1
3993 }
3994
3995 func fopen(file string,mode string)(fp*os.File){
3996     if mode == "r" {
3997         fp,err := os.Open(file)
3998         if( err != nil ){
3999             fmt.Printf("--E-- fopen(%s,%s)=(%v)\n",file,mode,err)
4000             return NULL_FP;
4001         }
4002         return fp;
4003     }else{
4004         fp,err := os.OpenFile(file,os.O_RDWR|os.O_CREATE|os.O_TRUNC,0600)
4005         if( err != nil ){
4006             return NULL_FP;
4007         }
4008         return fp;
4009     }
4010 }
4011 func fclose(fp*os.File){
4012     fp.Close()
4013 }
4014 func fflush(fp *os.File)(int){
4015     return 0
4016 }
4017 func fgetc(fp*os.File)(int){
4018     var buf [1]byte
4019     _,err := fp.Read(buf[0:1])
4020     if( err != nil ){
4021         return EOF;
4022     }else{
4023         return int(buf[0])
4024     }
4025 }
4026 func sfgets(str*string, size int, fp*os.File)(int){
4027     buf := make(StrBuff,size)
4028     var ch int
4029     var i int
4030     for i = 0; i < len(buf)-1; i++ {
4031         ch = fgetc(fp)
4032         //fprintf(stderr,"--fgets %d/%d %X\n",i,len(buf),ch)
4033         if( ch == EOF ){
4034             break;
4035         }
4036         buf[i] = byte(ch);
4037         if( ch == '\n' ){
4038             break;
4039         }
4040     }
4041     buf[i] = 0
4042     //fprintf(stderr,"--fgets %d/%d (%s)\n",i,len(buf),buf[0:i])
4043     return i
4044 }
4045 func fgets(buf StrBuff, size int, fp*os.File)(int){
4046     var ch int
4047     var i int
4048     for i = 0; i < len(buf)-1; i++ {
4049         ch = fgetc(fp)
4050         //fprintf(stderr,"--fgets %d/%d %X\n",i,len(buf),ch)
4051         if( ch == EOF ){
4052             break;
4053         }
4054         buf[i] = byte(ch);
4055         if( ch == '\n' ){
4056             break;
4057         }
4058     }
4059     buf[i] = 0
4060     //fprintf(stderr,"--fgets %d/%d (%s)\n",i,len(buf),buf[0:i])
4061     return i
4062 }
4063 func fputc(ch int , fp*os.File)(int){
4064     var buf [1]byte
4065     buf[0] = byte(ch)
4066     fp.Write(buf[0:1])
4067     return 0
4068 }
4069 func fputs(buf StrBuff, fp*os.File)(int){
4070     fp.Write(buf)
4071     return 0
4072 }
4073 func xfputss(str string, fp*os.File)(int){
4074     return fputs([]byte(str),fp)
4075 }
4076 func sscanf(str StrBuff,fmts string, params ...interface{})(int){
4077     fmt.Sscanf(string(str[0:strlen(str)]),fmts,params...)
4078     return 0
4079 }
4080 func fprintf(fp*os.File,fmts string, params ...interface{})(int){
4081     fmt.Fprintf(fp,fmts,params...)
4082     return 0
4083 }
4084
4085 // <a name="IME">Command Line IME</a>
4086 //----- MyIME
4087 var MyIMEVER = "MyIME/0.0.2";
4088 type RomKana struct {
4089     dic string // dictionary ID
4090     pat string // input pattern
4091     out string // output pattern
4092     hit int64 // count of hit and used

```

```

4093 }
4094 var dicents = 0
4095 var romkana [1024]RomKana
4096 var Romkan []RomKana
4097
4098 func isinDic(str string)(int){
4099     for i,v := range Romkan {
4100         if v.pat == str {
4101             return i
4102         }
4103     }
4104     return -1
4105 }
4106 const (
4107     DIC_COM_LOAD = "im"
4108     DIC_COM_DUMP = "s"
4109     DIC_COM_LIST = "ls"
4110     DIC_COM_ENA = "en"
4111     DIC_COM_DIS = "di"
4112 )
4113 func helpDic(argv []string){
4114     out := stderr
4115     cmd := ""
4116     if 0 < len(argv) { cmd = argv[0] }
4117     fprintf(out,"--- %v Usage\n",cmd)
4118     fprintf(out,"... Commands\n")
4119     fprintf(out,"... %v %3v [dicName] [dicURL] -- Import dictionary\n",cmd,DIC_COM_LOAD)
4120     fprintf(out,"... %v %3v [pattern] -- Search in dictionary\n",cmd,DIC_COM_DUMP)
4121     fprintf(out,"... %v %3v [dicName] -- List dictionaries\n",cmd,DIC_COM_LIST)
4122     fprintf(out,"... %v %3v [dicName] -- Disable dictionaries\n",cmd,DIC_COM_DIS)
4123     fprintf(out,"... %v %3v [dicName] -- Enable dictionaries\n",cmd,DIC_COM_ENA)
4124     fprintf(out,"... Keys ... %v\n","ESC can be used for '\\')")
4125     fprintf(out,"... \\c -- Reverse the case of the last character\n",)
4126     fprintf(out,"... \\i -- Replace input with translated text\n",)
4127     fprintf(out,"... \\j -- On/Off translation mode\n",)
4128     fprintf(out,"... \\l -- Force Lower Case\n",)
4129     fprintf(out,"... \\u -- Force Upper Case (software CapsLock)\n",)
4130     fprintf(out,"... \\v -- Show translation actions\n",)
4131     fprintf(out,"... \\x -- Replace the last input character with it Hexa-Decimal\n",)
4132 }
4133 func xDic(argv[]string){
4134     if len(argv) <= 1 {
4135         helpDic(argv)
4136         return
4137     }
4138     argv = argv[1:]
4139     var debug = false
4140     var info = false
4141     var silent = false
4142     var dump = false
4143     var builtin = false
4144     cmd := argv[0]
4145     argv = argv[1:]
4146     opt := ""
4147     arg := ""
4148
4149     if 0 < len(argv) {
4150         arg1 := argv[0]
4151         if arg1[0] == '-' {
4152             switch arg1 {
4153                 default: fmt.Printf("--Ed-- Unknown option(%v)\n",arg1)
4154                     return
4155                 case "-b": builtin = true
4156                 case "-d": debug = true
4157                 case "-s": silent = true
4158                 case "-v": info = true
4159             }
4160             opt = arg1
4161             argv = argv[1:]
4162         }
4163     }
4164
4165     dicName := ""
4166     dicURL := ""
4167     if 0 < len(argv) {
4168         arg = argv[0]
4169         dicName = arg
4170         argv = argv[1:]
4171     }
4172     if 0 < len(argv) {
4173         dicURL = argv[0]
4174         argv = argv[1:]
4175     }
4176     if false {
4177         fprintf(stderr,"--Dd-- com(%v) opt(%v) arg(%v)\n",cmd,opt,arg)
4178     }
4179     if cmd == DIC_COM_LOAD {
4180         //dicType := ""
4181         dicBody := ""
4182         if !builtin && dicName != "" && dicURL == "" {
4183             f,err := os.Open(dicName)
4184             if err == nil {
4185                 dicURL = dicName
4186             }else{
4187                 f,err = os.Open(dicName+".html")
4188                 if err == nil {
4189                     dicURL = dicName+".html"
4190                 }else{
4191                     f,err = os.Open("gshdic-"+dicName+".html")
4192                     if err == nil {
4193                         dicURL = "gshdic-"+dicName+".html"
4194                     }
4195                 }
4196             }
4197             if err == nil {
4198                 var buf = make([]byte,128*1024)
4199                 count,err := f.Read(buf)
4200                 f.Close()
4201                 if info {
4202                     fprintf(stderr,"--Id-- ReadDic(%v,%v)\n",count,err)
4203                 }
4204                 dicBody = string(buf[0:count])
4205             }
4206         }
4207         if dicBody == "" {
4208             switch arg {
4209                 default:
4210                     dicName = "WorldDic"
4211                     dicURL = WorldDic
4212                     if info {
4213                         fprintf(stderr,"--Id-- default dictionary \"%v\"\n",
4214                             dicName);
4215                     }
4216                 case "wnn":

```

```

4217         dicName = "WnnDic"
4218         dicURL = WnnDic
4219         case "sumomo":
4220             dicName = "SumomoDic"
4221             dicURL = SumomoDic
4222         case "sijimi":
4223             dicName = "SijimiDic"
4224             dicURL = SijimiDic
4225         case "jkl":
4226             dicName = "JKLJaDic"
4227             dicURL = JA_JKLDic
4228     }
4229     if debug {
4230         fprintf(stderr, "--Id-- %v URL=%v\n\n", dicName, dicURL);
4231     }
4232     dicv := strings.Split(dicURL, ",")
4233     if debug {
4234         fprintf(stderr, "--Id-- %v encoded data...\n", dicName)
4235         fprintf(stderr, "Type: %v\n", dicv[0])
4236         fprintf(stderr, "Body: %v\n", dicv[1])
4237         fprintf(stderr, "\n")
4238     }
4239     body, _ := base64.StdEncoding.DecodeString(dicv[1])
4240     dicBody = string(body)
4241 }
4242 if info {
4243     fmt.Printf("--Id-- %v %v\n", dicName, dicURL)
4244     fmt.Printf("%s\n", dicBody)
4245 }
4246 if debug {
4247     fprintf(stderr, "--Id-- dicName %v text...\n", dicName)
4248     fprintf(stderr, "%v\n", string(dicBody))
4249 }
4250 envv := strings.Split(dicBody, "\n");
4251 if info {
4252     fprintf(stderr, "--Id-- %v scan...\n", dicName);
4253 }
4254 var added int = 0
4255 var dup int = 0
4256 for i, v := range envv {
4257     var pat string
4258     var out string
4259     fmt.Sscanf(v, "%s %s", &pat, &out)
4260     if len(pat) <= 0 {
4261     } else {
4262         if 0 <= isinDic(pat) {
4263             dup += 1
4264             continue
4265         }
4266         romkana[dicents] = RomKana{dicName, pat, out, 0}
4267         dicents += 1
4268         added += 1
4269         Romkan = append(Romkan, RomKana{dicName, pat, out, 0})
4270         if debug {
4271             fmt.Printf("[%3v]:[%2v]%-8v [%2v] %v\n",
4272                 i, len(pat), pat, len(out), out)
4273         }
4274     }
4275 }
4276 if !silent {
4277     url := dicURL
4278     if strBegins(url, "data:") {
4279         url = "builtin"
4280     }
4281     fprintf(stderr, "--Id-- %v scan... %v added, %v dup. / %v total (%v)\n",
4282         dicName, added, dup, len(Romkan), url);
4283 }
4284 // should sort by pattern length for conplete match, for performance
4285 if debug {
4286     arg = "" // search pattern
4287     dump = true
4288 }
4289 }
4290 if cmd == DIC_COM_DUMP || dump {
4291     fprintf(stderr, "--Id-- %v dump... %v entries:\n", dicName, len(Romkan));
4292     var match = 0
4293     for i := 0; i < len(Romkan); i++ {
4294         dic := Romkan[i].dic
4295         pat := Romkan[i].pat
4296         out := Romkan[i].out
4297         if arg == "" || 0 <= strings.Index(pat, arg) || 0 <= strings.Index(out, arg) {
4298             fmt.Printf("\t\\%v\t%v [%2v]%-8v [%2v] %v\n",
4299                 i, dic, len(pat), pat, len(out), out)
4300             match += 1
4301         }
4302     }
4303     fprintf(stderr, "--Id-- %v matched %v / %v entries:\n", arg, match, len(Romkan));
4304 }
4305 }
4306 func loadDefaultDic(dic int){
4307     if( 0 < len(Romkan) ){
4308         return
4309     }
4310     //fprintf(stderr, "\r\n")
4311     xDic([]string{"dic", DIC_COM_LOAD});
4312 }
4313 var info = false
4314 if info {
4315     fprintf(stderr, "--Id-- Conguraturations!! WorldDic is now activated.\r\n")
4316     fprintf(stderr, "--Id-- enter \"dic\" command for help.\r\n")
4317 }
4318 }
4319 func readDic()(int){
4320     /*
4321     var rk *os.File;
4322     var dic = "MyIME-dic.txt";
4323     //rk = fopen("romkana.txt", "r");
4324     //rk = fopen("JK-JA-morse-dic.txt", "r");
4325     rk = fopen(dic, "r");
4326     if( rk == NULL_FP){
4327         if( true ){
4328             fprintf(stderr, "--%s-- Could not load %s\n", MyIMEVER, dic);
4329         }
4330         return -1;
4331     }
4332     if( true ){
4333         var di int;
4334         var line = make(StrBuff, 1024);
4335         var pat string
4336         var out string
4337         for di = 0; di < 1024; di++ {
4338             if( fgets(line, sizeof(line), rk) == NULLSP ){
4339                 break;
4340             }
4341         }
4342     }

```

```

4341     fmt.Sscanf(string(line[0:strlen(line)]), "%s %s", &pat, &out);
4342     //sscanf(line, "%s %[\r\n]", &pat, &out);
4343     romkana[di].pat = pat;
4344     romkana[di].out = out;
4345     //fprintf(stderr, "--Dd- %s %s\n", pat, out)
4346 }
4347 dicents += di
4348 if( false ){
4349     fprintf(stderr, "--%s-- loaded romkana.txt [%d]\n", MyIMEVER, di);
4350     for di = 0; di < dicents; di++ {
4351         fprintf(stderr,
4352             "%s %s\n", romkana[di].pat, romkana[di].out);
4353     }
4354 }
4355 }
4356 fclose(rk);
4357
4358 //romkana[dicents].pat = "//ddump"
4359 //romkana[dicents].pat = "//ddump" // dump the dic. and clean the command input
4360 */
4361 return 0;
4362 }
4363 func matchlen(stri string, pati string)(int){
4364     if strBegins(stri, pati) {
4365         return len(pati)
4366     }else{
4367         return 0
4368     }
4369 }
4370 func convs(src string)(string){
4371     var si int;
4372     var sx = len(src);
4373     var di int;
4374     var mi int;
4375     var dstb []byte
4376
4377     for si = 0; si < sx; { // search max. match from the position
4378         if strBegins(src[si:], "%x/") {
4379             // %x/integer/ // s/a/b/
4380             ix := strings.Index(src[si+3:], "/")
4381             if 0 < ix {
4382                 var iv int = 0
4383                 //fmt.Sscanf(src[si+3:si+3+ix], "%d", &iv)
4384                 fmt.Sscanf(src[si+3:si+3+ix], "%v", &iv)
4385                 sval := fmt.Sprintf("%x", iv)
4386                 bval := []byte(sval)
4387                 dstb = append(dstb, bval...)
4388                 si = si+3+ix+1
4389                 continue
4390             }
4391         }
4392         if strBegins(src[si:], "%d/") {
4393             // %d/integer/ // s/a/b/
4394             ix := strings.Index(src[si+3:], "/")
4395             if 0 < ix {
4396                 var iv int = 0
4397                 fmt.Sscanf(src[si+3:si+3+ix], "%v", &iv)
4398                 sval := fmt.Sprintf("%d", iv)
4399                 bval := []byte(sval)
4400                 dstb = append(dstb, bval...)
4401                 si = si+3+ix+1
4402                 continue
4403             }
4404         }
4405         if strBegins(src[si:], "%t") {
4406             now := time.Now()
4407             if true {
4408                 date := now.Format(time.Stamp)
4409                 dstb = append(dstb, []byte(date)... )
4410                 si = si+3
4411             }
4412             continue
4413         }
4414         var maxlen int = 0;
4415         var len int;
4416         mi = -1;
4417         for di = 0; di < dicents; di++ {
4418             len = matchlen(src[si:], romkana[di].pat);
4419             if( maxlen < len ){
4420                 maxlen = len;
4421                 mi = di;
4422             }
4423         }
4424         if( 0 < maxlen ){
4425             out := romkana[mi].out;
4426             dstb = append(dstb, []byte(out)...);
4427             si += maxlen;
4428         }else{
4429             dstb = append(dstb, src[si])
4430             si += 1;
4431         }
4432     }
4433     return string(dstb)
4434 }
4435 func trans(src string)(int){
4436     dst := convs(src);
4437     xfputss(dst, stderr);
4438     return 0;
4439 }
4440
4441 //----- LINEEDIT
4442 // "?" at the top of the line means searching history
4443
4444 // should be compatilbe with Telnet
4445 const (
4446     EV_MODE     = 255
4447     EV_IDLE     = 254
4448     EV_TIMEOUT  = 253
4449
4450     GO_UP       = 252 // k
4451     GO_DOWN    = 251 // j
4452     GO_RIGHT   = 250 // l
4453     GO_LEFT    = 249 // h
4454     DEL_RIGHT  = 248 // x
4455     GO_TOPL    = 'A'-0x40 // 0
4456     GO_ENDL    = 'E'-0x40 // $
4457
4458     GO_TOPW    = 239 // b
4459     GO_ENDW    = 238 // e
4460     GO_NEXTW   = 237 // w
4461
4462     GO_FORWCH  = 229 // f
4463     GO_PAIRCH  = 228 // %
4464 )

```

```

4465 GO_DEL      = 219 // d
4466
4467 HI_SRCH_FW   = 209 // /
4468 HI_SRCH_BK  = 208 // ?
4469 HI_SRCH_RFW = 207 // n
4470 HI_SRCH_RBK = 206 // N
4471 )
4472
4473 // should return number of octets ready to be read immediately
4474 //fprintf(stderr, "\n--Select(%v %v)\n", err, r.Bits[0])
4475
4476
4477 var EventRecvFd = -1 // file descriptor
4478 var EventSendFd = -1
4479 const EventFdOffset = 1000000
4480 const NormalFdOffset = 100
4481
4482 func putEvent(event int, evarg int){
4483     if true {
4484         if EventRecvFd < 0 {
4485             var pv = []int{-1,-1}
4486             syscall.Pipe(pv)
4487             EventRecvFd = pv[0]
4488             EventSendFd = pv[1]
4489             //fmt.Printf("--De-- EventPipe created[%v,%v]\n", EventRecvFd, EventSendFd)
4490         }
4491     }else{
4492         if EventRecvFd < 0 {
4493             // the document differs from this spec
4494             // https://golang.org/src/syscall/syscall_unix.go?s=8096:8158#L340
4495             sv, err := syscall.Socketpair(syscall.AF_UNIX, syscall.SOCK_STREAM, 0)
4496             EventRecvFd = sv[0]
4497             EventSendFd = sv[1]
4498             if err != nil {
4499                 fmt.Printf("--De-- EventSock created[%v,%v](%v)\n",
4500                     EventRecvFd, EventSendFd, err)
4501             }
4502         }
4503     }
4504     var buf = []byte{ byte(event)}
4505     n, err := syscall.Write(EventSendFd, buf)
4506     if err != nil {
4507         fmt.Printf("--De-- putEvent[%v](%3v)(%v %v)\n", EventSendFd, event, n, err)
4508     }
4509 }
4510 func ungets(str string){
4511     for _, ch := range str {
4512         putEvent(int(ch), 0)
4513     }
4514 }
4515 func (gsh*GshContext)xReplay(argv []string){
4516     hix := 0
4517     tempo := 1.0
4518     xtempo := 1.0
4519     repeat := 1
4520
4521     for _, a := range argv { // tempo
4522         if strBegins(a, "x") {
4523             fmt.Sscanf(a[1:], "%f", &xtempo)
4524             tempo = 1 / xtempo
4525             //fprintf(stderr, "--Dr-- tempo=[%v]%v\n", a[2:], tempo);
4526         }else
4527         if strBegins(a, "r") { // repeat
4528             fmt.Sscanf(a[1:], "%v", &repeat)
4529         }else
4530         if strBegins(a, "!") {
4531             fmt.Sscanf(a[1:], "%d", &hix)
4532         }else{
4533             fmt.Sscanf(a, "%d", &hix)
4534         }
4535     }
4536     if hix == 0 || len(argv) <= 1 {
4537         hix = len(gsh.CommandHistory)-1
4538     }
4539     fmt.Printf("--Ir-- Replay(!%v x%v r%v)\n", hix, xtempo, repeat)
4540     //dumpEvents(hix)
4541     //gsh.xScanReplay(hix, false, repeat, tempo, argv)
4542     go gsh.xScanReplay(hix, true, repeat, tempo, argv)
4543 }
4544
4545 // <a href="https://golang.org/pkg/syscall/#FdSet">syscall.Select</a>
4546 // 2020-0827 GShell-0.2.3
4547 /*
4548 func FpollIn1(fp *os.File, usec int)(uintptr){
4549     nfd := 1
4550
4551     rdv := syscall.FdSet {}
4552     fd1 := fp.Fd()
4553     bank1 := fd1/32
4554     mask1 := int32(1 <<< fd1)
4555     rdv.Bits[bank1] = mask1
4556
4557     fd2 := -1
4558     bank2 := -1
4559     var mask2 int32 = 0
4560
4561     if 0 <= EventRecvFd {
4562         fd2 = EventRecvFd
4563         nfd = fd2 + 1
4564         bank2 = fd2/32
4565         mask2 = int32(1 <<< fd2)
4566         rdv.Bits[bank2] |= mask2
4567         //fmt.Printf("--De-- EventPoll mask added [%d][%v][%v]\n", fd2, bank2, mask2)
4568     }
4569
4570     tout := syscall.NsecToTimeval(int64(usec*1000))
4571     //n, err := syscall.Select(nfd, &rdv, nil, nil, &tout) // spec. mismatch
4572     err := syscall.Select(nfd, &rdv, nil, nil, &tout)
4573     if err != nil {
4574         //fmt.Printf("--De-- select() err(%v)\n", err)
4575     }
4576     if err == nil {
4577         if 0 <= fd2 && (rdv.Bits[bank2] & mask2) != 0 {
4578             if false {
4579                 fmt.Printf("--De-- got Event\n")
4580             }
4581             return uintptr(EventFdOffset + fd2)
4582         }else
4583         if (rdv.Bits[bank1] & mask1) != 0 {
4584             return uintptr(NormalFdOffset + fd1)
4585         }else{
4586             return 1
4587         }
4588     }else{

```

```

4589     return 0
4590 }
4591 }
4592 */
4593 func fgetcTimeout1(fp *os.File,usec int)(int){
4594     READ1:
4595     //readyFd := FpollIn1(fp,usec)
4596     readyFd := CFPollIn1(fp,usec)
4597     if readyFd < 100 {
4598         return EV_TIMEOUT
4599     }
4600
4601     var buf [1]byte
4602
4603     if EventFdOffset <= readyFd {
4604         fd := int(readyFd-EventFdOffset)
4605         _,err := syscall.Read(fd,buf[0:1])
4606         if( err != nil ){
4607             return EOF;
4608         }else{
4609             if buf[0] == EV_MODE {
4610                 recvEvent(fd)
4611                 goto READ1
4612             }
4613             return int(buf[0])
4614         }
4615     }
4616
4617     _,err := fp.Read(buf[0:1])
4618     if( err != nil ){
4619         return EOF;
4620     }else{
4621         return int(buf[0])
4622     }
4623 }
4624
4625 func visibleChar(ch int)(string){
4626     switch {
4627     case '!' <= ch && ch <= '-':
4628         return string(ch)
4629     }
4630     switch ch {
4631     case '\t': return "\\s"
4632     case '\n': return "\\n"
4633     case '\r': return "\\r"
4634     case '\t': return "\\t"
4635     }
4636     switch ch {
4637     case 0x00: return "NUL"
4638     case 0x07: return "BEL"
4639     case 0x08: return "BS"
4640     case 0x0E: return "SO"
4641     case 0x0F: return "SI"
4642     case 0x1B: return "ESC"
4643     case 0x7F: return "DEL"
4644     }
4645     switch ch {
4646     case EV_IDLE: return fmt.Sprintf("IDLE")
4647     case EV_MODE: return fmt.Sprintf("MODE")
4648     }
4649     return fmt.Sprintf("%X",ch)
4650 }
4651 func recvEvent(fd int){
4652     var buf = make([]byte,1)
4653     _,_ = syscall.Read(fd,buf[0:1])
4654     if( buf[0] != 0 ){
4655         romkanmode = true
4656     }else{
4657         romkanmode = false
4658     }
4659 }
4660 func (gsh*GshContext)xScanReplay(hix int,replay bool,repeat int,tempo float64,argv[]string){
4661     var Start time.Time
4662     var events = []Event{}
4663     for _,e := range Events {
4664         if hix == 0 || e.CmdIndex == hix {
4665             events = append(events,e)
4666         }
4667     }
4668     elen := len(events)
4669     if 0 < elen {
4670         if events[elen-1].event == EV_IDLE {
4671             events = events[0:elen-1]
4672         }
4673     }
4674     for r := 0; r < repeat; r++ {
4675         for i,e := range events {
4676             nano := e.when.Nanosecond()
4677             micro := nano / 1000
4678             if Start.Second() == 0 {
4679                 Start = time.Now()
4680             }
4681             diff := time.Now().Sub(Start)
4682             if replay {
4683                 if e.event != EV_IDLE {
4684                     putEvent(e.event,0)
4685                     if e.event == EV_MODE { // event with arg
4686                         putEvent(int(e.evarg),0)
4687                     }
4688                 }
4689             }else{
4690                 fmt.Printf("%7.3fms #%-3v !%-3v [%v.%06d] %3v %02X %-4v %10.3fms\n",
4691                     float64(diff)/1000000.0,
4692                     i,
4693                     e.CmdIndex,
4694                     e.when.Format(time.Stamp),micro,
4695                     e.event,e.event,visibleChar(e.event),
4696                     float64(e.evarg)/1000000.0)
4697             }
4698             if e.event == EV_IDLE {
4699                 d := time.Duration(float64(time.Duration(e.evarg)) * tempo)
4700                 //nsleep(time.Duration(e.evarg))
4701                 nsleep(d)
4702             }
4703         }
4704     }
4705 }
4706 func dumpEvents(arg[]string){
4707     hix := 0
4708     if 1 < len(arg) {
4709         fmt.Sscanf(arg[1],"%d",&hix)
4710     }
4711     for i,e := range Events {
4712         nano := e.when.Nanosecond()

```

```

4713     micro := nano / 1000
4714     //if e.event != EV_TIMEOUT {
4715     if hix == 0 || e.CmdIndex == hix {
4716         fmt.Printf("#%-3v !%-3v [%v.%06d] %3v %02X %%-4v %10.3fms\n",i,
4717             e.CmdIndex,
4718             e.when.Format(time.Stamp),micro,
4719             e.event,e.event,visibleChar(e.event),float64(e.evarg)/1000000.0)
4720     }
4721     //}
4722 }
4723 }
4724 func fgetcTimeout(fp *os.File,usec int)(int){
4725     ch := fgetcTimeout1(fp,usec)
4726     if ch != EV_TIMEOUT {
4727         now := time.Now()
4728         if 0 < len(Events) {
4729             last := Events[len(Events)-1]
4730             dura := int64(now.Sub(last.when))
4731             Events = append(Events,Event{last.when,EV_IDLE,dura,last.CmdIndex})
4732         }
4733         Events = append(Events,Event{time.Now(),ch,0,CmdIndex})
4734     }
4735     return ch
4736 }
4737
4738 var AtConsoleLineTop = true
4739 var TtyMaxCol = 72 // to be obtained by ioctl1
4740 var EscTimeout = (100*1000)
4741 var (
4742     MODE_VicMode    bool    // vi compatible command mode
4743     MODE_ShowMode  bool    //
4744     romkanmode     bool    // shown translation mode, the mode to be retained
4745     MODE_Recursive bool    // recursive translation
4746     MODE_CapsLock  bool    // software CapsLock
4747     MODE_LowerLock bool    // force lower-case character lock
4748     MODE_ViInsert  int     // visible insert mode, should be like "I" icon in X Window
4749     MODE_ViTrace   bool    // output newline before translation
4750 )
4751 type IInput struct {
4752     lno      int
4753     lastlno int
4754     pch      []int // input queue
4755     prompt   string
4756     line     string
4757     right    string
4758     inJmode  bool
4759     pinJmode bool
4760     waitingMeta string // waiting meta character
4761     LastCmd   string
4762 }
4763 func (iin*IInput)Getc(timeoutUs int)(int){
4764     ch1 := EOF
4765     ch2 := EOF
4766     ch3 := EOF
4767     if( 0 < len(iin.pch) ){ // deQ
4768         ch1 = iin.pch[0]
4769         iin.pch = iin.pch[1:]
4770     }else{
4771         ch1 = fgetcTimeout(stdin,timeoutUs);
4772     }
4773     if( ch1 == 033 ){ // escape sequence
4774         ch2 = fgetcTimeout(stdin,EscTimeout);
4775         if( ch2 == EV_TIMEOUT ){
4776             }else{
4777                 ch3 = fgetcTimeout(stdin,EscTimeout);
4778                 if( ch3 == EV_TIMEOUT ){
4779                     iin.pch = append(iin.pch,ch2) // enQ
4780                 }else{
4781                     switch( ch2 ){
4782                         default:
4783                             iin.pch = append(iin.pch,ch2) // enQ
4784                             iin.pch = append(iin.pch,ch3) // enQ
4785                         case '|':
4786                             switch( ch3 ){
4787                                 case 'A': ch1 = GO_UP; // ^
4788                                 case 'B': ch1 = GO_DOWN; // v
4789                                 case 'C': ch1 = GO_RIGHT; // >
4790                                 case 'D': ch1 = GO_LEFT; // <
4791                                 case '3':
4792                                     ch4 := fgetcTimeout(stdin,EscTimeout);
4793                                     if( ch4 == '-' ){
4794                                         //fprintf(stderr,"x[%02X %02X %02X %02X]\n",ch1,ch2,ch3,ch4);
4795                                         ch1 = DEL_RIGHT
4796                                     }
4797                                 case '\\':
4798                                     //ch4 := fgetcTimeout(stdin,EscTimeout);
4799                                     //fprintf(stderr,"y[%02X %02X %02X %02X]\n",ch1,ch2,ch3,ch4);
4800                                     switch( ch3 ){
4801                                         case '-': ch1 = DEL_RIGHT
4802                                     }
4803                                 }
4804                             }
4805                         }
4806                 }
4807             }
4808         return ch1
4809     }
4810 }
4811 func (inn*IInput)clearline(){
4812     var i int
4813     fprintf(stderr,"\r");
4814     // should be ANSI ESC sequence
4815     for i = 0; i < TtyMaxCol; i++ { // to the max. position in this input action
4816         fputc(' ',os.Stderr);
4817     }
4818     fprintf(stderr,"\r");
4819 }
4820 func (iin*IInput)Redraw(){
4821     redraw(iin,iin.lno,iin.line,iin.right)
4822 }
4823 func redraw(iin *IInput,lno int,line string,right string){
4824     inMeta := false
4825     showMode := ""
4826     showMeta := "" // visible Meta mode on the cursor position
4827     showLino := fmt.Sprintf("%d!", lno)
4828     InsertMark := "" // in visible insert mode
4829     if MODE_VicMode {
4830     }else
4831     if 0 < len(iin.right) {
4832         InsertMark = " "
4833     }
4834     if( 0 < len(iin.waitingMeta) ){
4835         inMeta = true
4836     }

```

```

4837     if iin.waitingMeta[0] != 033 {
4838         showMeta = iin.waitingMeta
4839     }
4840 }
4841 if( romkanmode ){
4842     //romkanmark = " *";
4843 }else{
4844     //romkanmark = "";
4845 }
4846 if MODE_ShowMode {
4847     romkan := "-"
4848     inmeta := "-"
4849     inveri := ""
4850     if MODE_CapsLock {
4851         inmeta = "A"
4852     }
4853     if MODE_LowerLock {
4854         inmeta = "a"
4855     }
4856     if MODE_ViTrace {
4857         inveri = "v"
4858     }
4859     if MODE_VicMode {
4860         inveri = ":"
4861     }
4862     if romkanmode {
4863         romkan = "\343\201\202"
4864         if MODE_CapsLock {
4865             inmeta = "R"
4866         }else{
4867             inmeta = "r"
4868         }
4869     }
4870     if inMeta {
4871         inmeta = "\\\"
4872     }
4873     showMode = "["+romkan+inmeta+inveri+"]";
4874 }
4875 Pre := "\r" + showMode + showLino
4876 Output := ""
4877 Left := ""
4878 Right := ""
4879 if romkanmode {
4880     Left = convs(line)
4881     Right = InsertMark+convs(right)
4882 }else{
4883     Left = line
4884     Right = InsertMark+right
4885 }
4886 Output = Pre+Left
4887 if MODE_ViTrace {
4888     Output += iin.LastCmd
4889 }
4890 Output += showMeta+Right
4891 for len(Output) < TtyMaxCol { // to the max. position that may be dirty
4892     Output += " "
4893     // should be ANSI ESC sequence
4894     // not necessary just after newline
4895 }
4896 Output += Pre+Left+showMeta // to set the cursor to the current input position
4897 fprintf(stderr,"%s",Output)
4898 }
4899 if MODE_ViTrace {
4900     if 0 < len(iin.LastCmd) {
4901         iin.LastCmd = ""
4902         fprintf(stderr,"\r\n")
4903     }
4904 }
4905 AtConsoleLineTop = false
4906 }
4907 // <a href="https://golang.org/pkg/unicode/utf8/">utf8</a>
4908 func delHeadChar(str string)(rline string,head string){
4909     ,clen := utf8.DecodeRune([]byte(str))
4910     head = string(str[0:clen])
4911     return str[clen:],head
4912 }
4913 func delTailChar(str string)(rline string, last string){
4914     var i = 0
4915     var clen = 0
4916     for {
4917         ,siz := utf8.DecodeRune([]byte(str)[i:])
4918         if siz <= 0 { break }
4919         clen = siz
4920         i += siz
4921     }
4922     last = str[len(str)-clen:]
4923     return str[0:len(str)-clen],last
4924 }
4925 }
4926 // 3> for output and history
4927 // 4> for keylog?
4928 // <a name="getline">Command Line Editor</a>
4929 func xgetline(lno int, prevline string, gsh*GshContext)(string){
4930     var iin IInput
4931     iin.lastlno = lno
4932     iin.lno = lno
4933 }
4934 CmdIndex = len(gsh.CommandHistory)
4935 if( isatty(0) == 0 ){
4936     if( sfgets(&iin.line,LINESIZE,stdin) == NULL ){
4937         iin.line = "exit\n";
4938     }else{
4939     }
4940     return iin.line
4941 }
4942 if( true ){
4943     //var pts string;
4944     //pts = ptsname(0);
4945     //pts = ttyname(0);
4946     //fprintf(stderr,"--pts[0] = %s\n",pts?pts:"?");
4947 }
4948 if( false ){
4949     fprintf(stderr,"! ");
4950     fflush(stderr);
4951     sfgets(&iin.line,LINESIZE,stdin);
4952     return iin.line
4953 }
4954 system("/bin/stty -echo -icanon");
4955 xline := iin.xgetline1(prevline,gsh)
4956 system("/bin/stty echo sane");
4957 return xline
4958 }
4959 func (iin*IInput)Translate(cmdch int){
4960     romkanmode = !romkanmode;

```



```

4961     if MODE_ViTrace {
4962         fprintf(stderr,"%v\r\n",string(cmdch));
4963     }else
4964     if( cmdch == 'J' ){
4965         fprintf(stderr,"J\r\n");
4966         iin.inJmode = true
4967     }
4968     iin.Redraw();
4969     loadDefaultDic(cmdch);
4970     iin.Redraw();
4971 }
4972 func (iin*IInput)Replace(cmdch int){
4973     iin.LastCmd = fmt.Sprintf("%v",string(cmdch))
4974     iin.Redraw();
4975     loadDefaultDic(cmdch);
4976     dst := convs(iin.line+iin.right);
4977     iin.line = dst
4978     iin.right = ""
4979     if( cmdch == 'I' ){
4980         fprintf(stderr,"I\r\n");
4981         iin.inJmode = true
4982     }
4983     iin.Redraw();
4984 }
4985 // aa 12 alal
4986 func isAlpha(ch rune)(bool){
4987     if 'a' <= ch && ch <= 'z' || 'A' <= ch && ch <= 'Z' {
4988         return true
4989     }
4990     return false
4991 }
4992 func isAlnum(ch rune)(bool){
4993     if 'a' <= ch && ch <= 'z' || 'A' <= ch && ch <= 'Z' {
4994         return true
4995     }
4996     if '0' <= ch && ch <= '9' {
4997         return true
4998     }
4999     return false
5000 }
5001
5002 // 0.2.8 2020-0901 created
5003 // <a href="https://golang.org/pkg/unicode/utf8/#DecodeRuneInString">DecodeRuneInString</a>
5004 func (iin*IInput)GotoTOPW(){
5005     str := iin.line
5006     i := len(str)
5007     if i <= 0 {
5008         return
5009     }
5010     //i0 := i
5011     i -= 1
5012     lastSize := 0
5013     var lastRune rune
5014     var found = -1
5015     for 0 < i { // skip preamble spaces
5016         lastRune,lastSize = utf8.DecodeRuneInString(str[i:])
5017         if !isAlnum(lastRune) { // character, type, or string to be searched
5018             i -= lastSize
5019             continue
5020         }
5021         break
5022     }
5023     for 0 < i {
5024         lastRune,lastSize = utf8.DecodeRuneInString(str[i:])
5025         if lastSize <= 0 { continue } // not the character top
5026         if !isAlnum(lastRune) { // character, type, or string to be searched
5027             found = i
5028             break
5029         }
5030         i -= lastSize
5031     }
5032     if found < 0 && i == 0 {
5033         found = 0
5034     }
5035     if 0 <= found {
5036         if isAlnum(lastRune) { // or non-kana character
5037             }else{ // when positioning to the top o the word
5038                 i += lastSize
5039             }
5040             iin.right = str[i:] + iin.right
5041             if 0 < i {
5042                 iin.line = str[0:i]
5043             }else{
5044                 iin.line = ""
5045             }
5046         }
5047         //fmt.Printf("\n(%d,%d,%d)[%s][%s]\n",i0,i,found,iin.line,iin.right)
5048         //fmt.Printf("") // set debug messae at the end of line
5049     }
5050 // 0.2.8 2020-0901 created
5051 func (iin*IInput)GotoENDW(){
5052     str := iin.right
5053     if len(str) <= 0 {
5054         return
5055     }
5056     lastSize := 0
5057     var lastRune rune
5058     var lastW = 0
5059     i := 0
5060     inWord := false
5061
5062     lastRune,lastSize = utf8.DecodeRuneInString(str[0:])
5063     if isAlnum(lastRune) {
5064         r,z := utf8.DecodeRuneInString(str[lastSize:])
5065         if 0 < z && isAlnum(r) {
5066             inWord = true
5067         }
5068     }
5069     for i < len(str) {
5070         lastRune,lastSize = utf8.DecodeRuneInString(str[i:])
5071         if lastSize <= 0 { break } // broken data?
5072         if !isAlnum(lastRune) { // character, type, or string to be searched
5073             break
5074         }
5075         lastW = i // the last alnum if in alnum word
5076         i += lastSize
5077     }
5078     if inWord {
5079         goto DISP
5080     }
5081     for i < len(str) {
5082         lastRune,lastSize = utf8.DecodeRuneInString(str[i:])
5083         if lastSize <= 0 { break } // broken data?
5084         if isAlnum(lastRune) { // character, type, or string to be searched

```

```

5085         break
5086     }
5087     i += lastSize
5088 }
5089 for i < len(str) {
5090     lastRune,lastSize = utf8.DecodeRuneInString(str[i:])
5091     if lastSize <= 0 { break } // broken data?
5092     if !isAlnum(lastRune) { // character, type, or string to be searched
5093         break
5094     }
5095     lastW = i
5096     i += lastSize
5097 }
5098 DISP:
5099 if 0 < lastW {
5100     iin.line = iin.line + str[0:lastW]
5101     iin.right = str[lastW:]
5102 }
5103 //fmt.Printf("\n%d)[%s][%s]\n",i,iin.line,iin.right)
5104 //fmt.Printf("") // set debug messae at the end of line
5105 }
5106 // 0.2.8 2020-0901 created
5107 func (iin*Input)GotoNEXTW(){
5108     str := iin.right
5109     if len(str) <= 0 {
5110         return
5111     }
5112     lastSize := 0
5113     var lastRune rune
5114     var found = -1
5115     i := 1
5116     for i < len(str) {
5117         lastRune,lastSize = utf8.DecodeRuneInString(str[i:])
5118         if lastSize <= 0 { break } // broken data?
5119         if !isAlnum(lastRune) { // character, type, or string to be searched
5120             found = i
5121             break
5122         }
5123     }
5124     i += lastSize
5125 }
5126 if 0 < found {
5127     if isAlnum(lastRune) { // or non-kana character
5128     }else{ // when positioning to the top o the word
5129         found += lastSize
5130     }
5131     iin.line = iin.line + str[0:found]
5132     if 0 < found {
5133         iin.right = str[found:]
5134     }else{
5135         iin.right = ""
5136     }
5137 }
5138 //fmt.Printf("\n%d)[%s][%s]\n",i,iin.line,iin.right)
5139 //fmt.Printf("") // set debug messae at the end of line
5140 }
5141 // 0.2.8 2020-0902 created
5142 func (iin*Input)GotoPAIRCH(){
5143     str := iin.right
5144     if len(str) <= 0 {
5145         return
5146     }
5147     lastRune,lastSize := utf8.DecodeRuneInString(str[0:])
5148     if lastSize <= 0 {
5149         return
5150     }
5151     forw := false
5152     back := false
5153     pair := ""
5154     switch string(lastRune){
5155     case "{": pair = "}"; forw = true
5156     case "(": pair = ")"; forw = true
5157     case "[": pair = "]"; forw = true
5158     case "<": pair = ">"; forw = true
5159     case "<=": pair = ">="; forw = true
5160     case ">": pair = "<"; back = true
5161     case ">=": pair = "<="; back = true
5162     case "\\": pair = "\\\""; // context depednet, can be f" or back-double quote
5163     case "'": pair = "'"; // context depednet, can be f' or back-quote
5164     // case Japanese Kakkos
5165     }
5166     if forw {
5167         iin.SearchForward(pair)
5168     }
5169     if back {
5170         iin.SearchBackward(pair)
5171     }
5172 }
5173 // 0.2.8 2020-0902 created
5174 func (iin*Input)SearchForward(pat string)(bool){
5175     right := iin.right
5176     found := -1
5177     i := 0
5178     if strBegins(right,pat) {
5179         _,z := utf8.DecodeRuneInString(right[i:])
5180         if 0 < z {
5181             i += z
5182         }
5183     }
5184     for i < len(right) {
5185         if strBegins(right[i:],pat) {
5186             found = i
5187             break
5188         }
5189         _,z := utf8.DecodeRuneInString(right[i:])
5190         if z <= 0 { break }
5191         i += z
5192     }
5193     if 0 <= found {
5194         iin.line = iin.line + right[0:found]
5195         iin.right = iin.right[found:]
5196         return true
5197     }else{
5198         return false
5199     }
5200 }
5201 // 0.2.8 2020-0902 created
5202 func (iin*Input)SearchBackward(pat string)(bool){
5203     line := iin.line
5204     found := -1
5205     i := len(line)-1
5206     for i = i; 0 <= i; i-- {
5207         _,z := utf8.DecodeRuneInString(line[i:])
5208         if z <= 0 {

```

```

5209         continue
5210     }
5211     //fprintf(stderr, "-- %v %v\n", pat, line[i:])
5212     if strBegins(line[i:], pat) {
5213         found = i
5214         break
5215     }
5216 }
5217 //fprintf(stderr, "--%d\n", found)
5218 if 0 <= found {
5219     iin.right = line[found:] + iin.right
5220     iin.line = line[0:found]
5221     return true
5222 }else{
5223     return false
5224 }
5225 }
5226 // 0.2.8 2020-0902 created
5227 // search from top, end, or current position
5228 func (gsh*GshContext)SearchHistory(pat string, forw bool)(bool, string){
5229     if forw {
5230         for _, v := range gsh.CommandHistory {
5231             if 0 <= strings.Index(v.CmdLine, pat) {
5232                 //fprintf(stderr, "\n--De-- found !%v [%v]%v\n", i, pat, v.CmdLine)
5233                 return true, v.CmdLine
5234             }
5235         }
5236     }else{
5237         hlen := len(gsh.CommandHistory)
5238         for i := hlen-1; 0 < i; i-- {
5239             v := gsh.CommandHistory[i]
5240             if 0 <= strings.Index(v.CmdLine, pat) {
5241                 //fprintf(stderr, "\n--De-- found !%v [%v]%v\n", i, pat, v.CmdLine)
5242                 return true, v.CmdLine
5243             }
5244         }
5245     }
5246     //fprintf(stderr, "\n--De-- not-found(%v)\n", pat)
5247     return false, "(Not Found in History)"
5248 }
5249 // 0.2.8 2020-0902 created
5250 func (iin*IInput)GotoFORWSTR(pat string, gsh*GshContext){
5251     found := false
5252     if 0 < len(iin.right) {
5253         found = iin.SearchForward(pat)
5254     }
5255     if !found {
5256         found, line := gsh.SearchHistory(pat, true)
5257         if found {
5258             iin.line = line
5259             iin.right = ""
5260         }
5261     }
5262 }
5263 func (iin*IInput)GotoBACKSTR(pat string, gsh*GshContext){
5264     found := false
5265     if 0 < len(iin.line) {
5266         found = iin.SearchBackward(pat)
5267     }
5268     if !found {
5269         found, line := gsh.SearchHistory(pat, false)
5270         if found {
5271             iin.line = line
5272             iin.right = ""
5273         }
5274     }
5275 }
5276 func (iin*IInput)getString1(prompt string)(string){ // should be editable
5277     iin.clearline();
5278     fprintf(stderr, "\r%v", prompt)
5279     str := ""
5280     for {
5281         ch := iin.Getc(10*1000*1000)
5282         if ch == '\n' || ch == '\r' {
5283             break
5284         }
5285         sch := string(ch)
5286         str += sch
5287         fprintf(stderr, "%s", sch)
5288     }
5289     return str
5290 }
5291 }
5292 // search pattern must be an array and selectable with ^N/^P
5293 var SearchPat = ""
5294 var SearchForw = true
5295 }
5296 func (iin*IInput)xgetline1(prevline string, gsh*GshContext)(string){
5297     var ch int;
5298     MODE_ShowMode = false
5299     MODE_VicMode = false
5300     iin.Redraw();
5301     first := true
5302     for cix := 0; ; cix++ {
5303         iin.pinJmode = iin.inJmode
5304         iin.inJmode = false
5305         ch = iin.Getc(1000*1000)
5306         if ch != EV_TIMEOUT && first {
5307             first = false
5308             mode := 0
5309             if romkanmode {
5310                 mode = 1
5311             }
5312             now := time.Now()
5313             Events = append(Events, Event{now, EV_MODE, int64(mode), CmdIndex})
5314         }
5315         if ch == 033 {
5316             MODE_ShowMode = true
5317             MODE_VicMode = !MODE_VicMode
5318             iin.Redraw();
5319             continue
5320         }
5321         if MODE_VicMode {
5322             switch ch {
5323                 case '0': ch = GO_TOPL
5324                 case '$': ch = GO_ENDL
5325                 case 'b': ch = GO_TOPW
5326                 case 'e': ch = GO_ENDW
5327                 case 'w': ch = GO_NEXTW
5328                 case '%': ch = GO_PAIRCH

```

```

5333
5334 case 'j': ch = GO_DOWN
5335 case 'k': ch = GO_UP
5336 case 'h': ch = GO_LEFT
5337 case 'l': ch = GO_RIGHT
5338 case 'x': ch = DEL_RIGHT
5339 case 'a': MODE_VicMode = !MODE_VicMode
5340 ch = GO_RIGHT
5341 case 'i': MODE_VicMode = !MODE_VicMode
5342 iin.Redraw();
5343 continue
5344 case '~':
5345 right,head := delHeadChar(iin.right)
5346 if len([]byte(head)) == 1 {
5347 ch = int(head[0])
5348 if( 'a' <= ch && ch <= 'z' ){
5349 ch = ch + 'A'-'a'
5350 }else
5351 if( 'A' <= ch && ch <= 'Z' ){
5352 ch = ch + 'a'-'A'
5353 }
5354 iin.right = string(ch) + right
5355 }
5356 iin.Redraw();
5357 continue
5358 case 'f': // GO_FORWCH
5359 iin.Redraw();
5360 ch = iin.Getc(3*1000*1000)
5361 if ch == EV_TIMEOUT {
5362 iin.Redraw();
5363 continue
5364 }
5365 SearchPat = string(ch)
5366 SearchForw = true
5367 iin.GotoFORWSTR(SearchPat,gsh)
5368 iin.Redraw();
5369 continue
5370 case '/':
5371 SearchPat = iin.getstringl("/") // should be editable
5372 SearchForw = true
5373 iin.GotoFORWSTR(SearchPat,gsh)
5374 iin.Redraw();
5375 continue
5376 case '?':
5377 SearchPat = iin.getstringl("?") // should be editable
5378 SearchForw = false
5379 iin.GotoBACKSTR(SearchPat,gsh)
5380 iin.Redraw();
5381 continue
5382 case 'n':
5383 if SearchForw {
5384 iin.GotoFORWSTR(SearchPat,gsh)
5385 }else{
5386 iin.GotoBACKSTR(SearchPat,gsh)
5387 }
5388 iin.Redraw();
5389 continue
5390 case 'N':
5391 if !SearchForw {
5392 iin.GotoFORWSTR(SearchPat,gsh)
5393 }else{
5394 iin.GotoBACKSTR(SearchPat,gsh)
5395 }
5396 iin.Redraw();
5397 continue
5398 }
5399 }
5400 switch ch {
5401 case GO_TOPW:
5402 iin.GotoTOPW()
5403 iin.Redraw();
5404 continue
5405 case GO_ENDW:
5406 iin.GotoENDW()
5407 iin.Redraw();
5408 continue
5409 case GO_NEXTW:
5410 // to next space then
5411 iin.GotoNEXTW()
5412 iin.Redraw();
5413 continue
5414 case GO_PAIRCH:
5415 iin.GotoPAIRCH()
5416 iin.Redraw();
5417 continue
5418 }
5419 //fprintf(stderr, "A[%02X]\n",ch);
5420 if( ch == '\\\ | | ch == 033 ){
5421 MODE_ShowMode = true
5422 metach := ch
5423 iin.waitingMeta = string(ch)
5424 iin.Redraw();
5425 // set cursor //fprintf(stderr, "???\b\b\b\b")
5426 ch = fgetcTimeout(stdin,2000*1000)
5427 // reset cursor
5428 iin.waitingMeta = ""
5429
5430 cmdch := ch
5431 if( ch == EV_TIMEOUT ){
5432 if metach == 033 {
5433 continue
5434 }
5435 ch = metach
5436 }else
5437 /*
5438 if( ch == 'm' || ch == 'M' ){
5439 mch := fgetcTimeout(stdin,1000*1000)
5440 if mch == 'r' {
5441 romkanmode = true
5442 }else{
5443 romkanmode = false
5444 }
5445 }
5446 continue
5447 }else
5448 /*
5449 if( ch == 'k' || ch == 'K' ){
5450 MODE_Recursive = !MODE_Recursive
5451 iin.Translate(cmdch);
5452 continue
5453 }else
5454 if( ch == 'j' || ch == 'J' ){
5455 iin.Translate(cmdch);
5456 continue

```

```

5457     }else
5458     if( ch == 'i' || ch == 'I' ){
5459         iin.Replace(cmdch);
5460         continue
5461     }else
5462     if( ch == 'l' || ch == 'L' ){
5463         MODE_LowerLock = !MODE_LowerLock
5464         MODE_CapsLock = false
5465         if MODE_ViTrace {
5466             fprintf(stderr,"%v\r\n",string(cmdch));
5467         }
5468         iin.Redraw();
5469         continue
5470     }else
5471     if( ch == 'u' || ch == 'U' ){
5472         MODE_CapsLock = !MODE_CapsLock
5473         MODE_LowerLock = false
5474         if MODE_ViTrace {
5475             fprintf(stderr,"%v\r\n",string(cmdch));
5476         }
5477         iin.Redraw();
5478         continue
5479     }else
5480     if( ch == 'v' || ch == 'V' ){
5481         MODE_ViTrace = !MODE_ViTrace
5482         if MODE_ViTrace {
5483             fprintf(stderr,"%v\r\n",string(cmdch));
5484         }
5485         iin.Redraw();
5486         continue
5487     }else
5488     if( ch == 'c' || ch == 'C' ){
5489         if 0 < len(iin.line) {
5490             xline,tail := delTailChar(iin.line)
5491             if len([]byte(tail)) == 1 {
5492                 ch = int(tail[0])
5493                 if( 'a' <= ch && ch <= 'z' ){
5494                     ch = ch + 'A'-'a'
5495                 }else
5496                 if( 'A' <= ch && ch <= 'Z' ){
5497                     ch = ch + 'a'-'A'
5498                 }
5499                 iin.line = xline + string(ch)
5500             }
5501         }
5502         if MODE_ViTrace {
5503             fprintf(stderr,"%v\r\n",string(cmdch));
5504         }
5505         iin.Redraw();
5506         continue
5507     }else{
5508         iin.pch = append(iin.pch,ch) // push
5509         ch = '\\'
5510     }
5511 }
5512 switch( ch ){
5513 case 'P'-0x40: ch = GO_UP
5514 case 'N'-0x40: ch = GO_DOWN
5515 case 'B'-0x40: ch = GO_LEFT
5516 case 'F'-0x40: ch = GO_RIGHT
5517 }
5518 //fprintf(stderr,"B[%02X]\n",ch);
5519 switch( ch ){
5520 case 0:
5521     continue;
5522
5523 case '\t':
5524     iin.Replace('j');
5525     continue
5526 case 'X'-0x40:
5527     iin.Replace('j');
5528     continue
5529
5530 case EV_TIMEOUT:
5531     iin.Redraw();
5532     if iin.pinJmode {
5533         fprintf(stderr,"\\J\r\n")
5534         iin.inJmode = true
5535     }
5536     continue
5537 case GO_UP:
5538     if iin.lno == 1 {
5539         continue
5540     }
5541     cmd,ok := gsh.cmdStringInHistory(iin.lno-1)
5542     if ok {
5543         iin.line = cmd
5544         iin.right = ""
5545         iin.lno = iin.lno - 1
5546     }
5547     iin.Redraw();
5548     continue
5549 case GO_DOWN:
5550     cmd,ok := gsh.cmdStringInHistory(iin.lno+1)
5551     if ok {
5552         iin.line = cmd
5553         iin.right = ""
5554         iin.lno = iin.lno + 1
5555     }else{
5556         iin.line = ""
5557         iin.right = ""
5558         if iin.lno == iin.lastlno-1 {
5559             iin.lno = iin.lno + 1
5560         }
5561     }
5562     iin.Redraw();
5563     continue
5564 case GO_LEFT:
5565     if 0 < len(iin.line) {
5566         xline,tail := delTailChar(iin.line)
5567         iin.line = xline
5568         iin.right = tail + iin.right
5569     }
5570     iin.Redraw();
5571     continue;
5572 case GO_RIGHT:
5573     if( 0 < len(iin.right) && iin.right[0] != 0 ){
5574         xright,head := delHeadChar(iin.right)
5575         iin.right = xright
5576         iin.line += head
5577     }
5578     iin.Redraw();
5579     continue;
5580 case EOF:

```

```

5581         goto EXIT;
5582     case 'R'-0x40: // replace
5583         dst := convs(iin.line+iin.right);
5584         iin.line = dst
5585         iin.right = ""
5586         iin.Redraw();
5587         continue;
5588     case 'T'-0x40: // just show the result
5589         readDic();
5590         romkanmode = !romkanmode;
5591         iin.Redraw();
5592         continue;
5593     case 'L'-0x40:
5594         iin.Redraw();
5595         continue
5596     case 'K'-0x40:
5597         iin.right = ""
5598         iin.Redraw();
5599         continue
5600     case 'E'-0x40:
5601         iin.line += iin.right
5602         iin.right = ""
5603         iin.Redraw();
5604         continue
5605     case 'A'-0x40:
5606         iin.right = iin.line + iin.right
5607         iin.line = ""
5608         iin.Redraw();
5609         continue
5610     case 'U'-0x40:
5611         iin.line = ""
5612         iin.right = ""
5613         iin.clearline();
5614         iin.Redraw();
5615         continue;
5616     case DEL_RIGHT:
5617         if( 0 < len(iin.right) ){
5618             iin.right,_ = delHeadChar(iin.right)
5619             iin.Redraw();
5620         }
5621         continue;
5622     case 0x7F: // BS? not DEL
5623         if( 0 < len(iin.line) ){
5624             iin.line,_ = delTailChar(iin.line)
5625             iin.Redraw();
5626         }
5627         /*
5628         else
5629             if( 0 < len(iin.right) ){
5630                 iin.right,_ = delHeadChar(iin.right)
5631                 iin.Redraw();
5632             }
5633         */
5634         continue;
5635     case 'H'-0x40:
5636         if( 0 < len(iin.line) ){
5637             iin.line,_ = delTailChar(iin.line)
5638             iin.Redraw();
5639         }
5640         continue;
5641     }
5642     if( ch == '\n' || ch == '\r' ){
5643         iin.line += iin.right;
5644         iin.right = ""
5645         iin.Redraw();
5646         fputc(ch,stderr);
5647         AtConsoleLineTop = true
5648         break;
5649     }
5650     if MODE_CapsLock {
5651         if 'a' <= ch && ch <= 'z' {
5652             ch = ch+'A'-'a'
5653         }
5654     }
5655     if MODE_LowerLock {
5656         if 'A' <= ch && ch <= 'Z' {
5657             ch = ch+'a'-'A'
5658         }
5659     }
5660     iin.line += string(ch);
5661     iin.Redraw();
5662 }
5663 EXIT:
5664     return iin.line + iin.right;
5665 }
5666
5667 func getline_main(){
5668     line := xgetline(0,"",nil)
5669     fprintf(stderr,"%s\n",line);
5670 /*
5671     dp = strpbrk(line,"\r\n");
5672     if( dp != NULL ){
5673         *dp = 0;
5674     }
5675
5676     if( 0 ){
5677         fprintf(stderr,"\n(%d)\n",int(strlen(line)));
5678     }
5679     if( lseek(3,0,0) == 0 ){
5680         if( romkanmode ){
5681             var buf [8*1024]bytete;
5682             convs(line,buf);
5683             strcpy(line,buf);
5684         }
5685         write(3,line,strlen(line));
5686         ftruncate(3,lseek(3,0,SEEK_CUR));
5687         //fprintf(stderr,"outsize=%d\n",(int)lseek(3,0,SEEK_END));
5688         lseek(3,0,SEEK_SET);
5689         close(3);
5690     }else{
5691         fprintf(stderr,"\r\ngotline: ");
5692         trans(line);
5693         //printf("%s\n",line);
5694         printf("\n");
5695     }
5696 */
5697 }
5698 //== end ====== getline
5699
5700 //
5701 // $USERHOME/.gsh/
5702 //     gsh-rc.txt, or gsh-configure.txt
5703 //     gsh-history.txt
5704 //     gsh-aliases.txt // should be conditional?

```

```

5705 //
5706 func (gshCtx *GshContext)gshSetupHomedir()(bool) {
5707     homedir,found := userHomeDir()
5708     if !found {
5709         fmt.Printf("--E-- You have no UserHomeDir\n")
5710         return true
5711     }
5712     gshhome := homedir + "/" + GSH_HOME
5713     _, err2 := os.Stat(gshhome)
5714     if err2 != nil {
5715         err3 := os.Mkdir(gshhome,0700)
5716         if err3 != nil {
5717             fmt.Printf("--E-- Could not Create %s (%s)\n",
5718                 gshhome,err3)
5719             return true
5720         }
5721         fmt.Printf("--I-- Created %s\n",gshhome)
5722     }
5723     gshCtx.GshHomeDir = gshhome
5724     return false
5725 }
5726 func setupGshContext()(GshContext,bool){
5727     gshPA := syscall.ProcAttr {
5728         "", // the staring directory
5729         os.Environ(), // environ[]
5730         []uintptr{os.Stdin.Fd(),os.Stdout.Fd(),os.Stderr.Fd()},
5731         nil, // OS specific
5732     }
5733     cwd, _ := os.Getwd()
5734     gshCtx := GshContext {
5735         cwd, // StartDir
5736         "", // GetLine
5737         []GchdirHistory { {cwd,time.Now(),0} }, // ChdirHistory
5738         gshPA,
5739         []GCommandHistory{}, //something for invokation?
5740         GCommandHistory{}, // CmdCurrent
5741         false,
5742         []int{},
5743         syscall.Rusage{},
5744         "", // GshHomeDir
5745         Ttyid(),
5746         false,
5747         false,
5748         []PluginInfo{},
5749         []string{},
5750         "",
5751         "v",
5752         ValueStack{},
5753         GServer{"",""}, // LastServer
5754         "", // RSERVER
5755         cwd, // RWD
5756         CheckSum{},
5757     }
5758     err := gshCtx.gshSetupHomedir()
5759     return gshCtx, err
5760 }
5761 func (gsh*GshContext)gshelllh(gline string)(bool){
5762     ghist := gsh.CmdCurrent
5763     ghist.WorkDir,_ = os.Getwd()
5764     ghist.WorkDirX = len(gsh.ChdirHistory)-1
5765     //fmt.Printf("--D--ChdirHistory(#{@d}\n",len(gsh.ChdirHistory))
5766     ghist.StartAt = time.Now()
5767     rusagev1 := Getrusagev()
5768     gsh.CmdCurrent.FoundFile = []string{}
5769     fin := gsh.tgshelll(gline)
5770     rusagev2 := Getrusagev()
5771     ghist.Rusagev = RusageSubv(rusagev2,rusagev1)
5772     ghist.EndAt = time.Now()
5773     ghist.CmdLine = gline
5774     ghist.FoundFile = gsh.CmdCurrent.FoundFile
5775
5776     /* record it but not show in list by default
5777     if len(gline) == 0 {
5778         continue
5779     }
5780     if gline == "hi" || gline == "history" { // don't record it
5781         continue
5782     }
5783     */
5784     gsh.CommandHistory = append(gsh.CommandHistory, ghist)
5785     return fin
5786 }
5787 // <a name="main">Main loop</a>
5788 func script(gshCtxGiven *GshContext) (_ GshContext) {
5789     gshCtxBuf,err0 := setupGshContext()
5790     if err0 {
5791         return gshCtxBuf;
5792     }
5793     gshCtx := &gshCtxBuf
5794
5795     //fmt.Printf("--I-- GSH_HOME=%s\n",gshCtx.GshHomeDir)
5796     //resmap()
5797
5798     /*
5799     if false {
5800         gsh_getlinev, with_exgetline :=
5801             which("PATH",[string{"which","gsh-getline","-s"}])
5802         if with_exgetline {
5803             gsh_getlinev[0] = toFullpath(gsh_getlinev[0])
5804             gshCtx.GetLine = toFullpath(gsh_getlinev[0])
5805         }else{
5806             fmt.Printf("--W-- No gsh-getline found. Using internal getline.\n");
5807         }
5808     }
5809     */
5810
5811     ghist0 := gshCtx.CmdCurrent // something special, or gshrc script, or permanent history
5812     gshCtx.CommandHistory = append(gshCtx.CommandHistory,ghist0)
5813
5814     prevline := ""
5815     skipping := false
5816     for hix := len(gshCtx.CommandHistory); ; {
5817         gline := gshCtx.getline(hix,skipping,prevline)
5818         if skipping {
5819             if strings.Index(gline,"fi") == 0 {
5820                 fmt.Printf("fi\n");
5821                 skipping = false;
5822             }else{
5823                 //fmt.Printf("%s\n",gline);
5824             }
5825             continue
5826         }
5827         if strings.Index(gline,"if") == 0 {
5828             //fmt.Printf("--D-- if start: %s\n",gline);

```

```

5829     skipping = true;
5830     continue
5831 }
5832 }
5833 if false {
5834     os.Stdout.Write([]byte("gotline:"))
5835     os.Stdout.Write([]byte(gline))
5836     os.Stdout.Write([]byte("\n"))
5837 }
5838 gline = strsubst(gshCtx,gline,true)
5839 if false {
5840     fmt.Printf("fmt.Printf %%v - %v\n",gline)
5841     fmt.Printf("fmt.Printf %%s - %s\n",gline)
5842     fmt.Printf("fmt.Printf %%x - %x\n",gline)
5843     fmt.Printf("fmt.Printf %%U - %U\n",gline)
5844     os.Stdout.Write([]byte(gline))
5845     fmt.Printf("\n")
5846 }
5847 /*
5848 // should be cared in substitution ?
5849 if 0 < len(gline) && gline[0] == '!' {
5850     xgline, set, err := searchHistory(gshCtx,gline)
5851     if err {
5852         continue
5853     }
5854     if set {
5855         // set the line in command line editor
5856     }
5857     gline = xgline
5858 }
5859 */
5860 fin := gshCtx.gshellllh(gline)
5861 if fin {
5862     break;
5863 }
5864 prevline = gline;
5865 hix++;
5866 }
5867 return *gshCtx
5868 }
5869 func main() {
5870     gshCtxBuf := GshContext{}
5871     gsh := &gshCtxBuf
5872     argv := os.Args
5873
5874     if( isin("wss",argv) ){
5875         gj_server(argv[1:]);
5876         return;
5877     }
5878     if( isin("wsc",argv) ){
5879         gj_client(argv[1:]);
5880         return;
5881     }
5882     if 1 < len(argv) {
5883         if isin("version",argv){
5884             gsh.showVersion(argv)
5885             return
5886         }
5887         if argv[1] == "gj" {
5888             if argv[2] == "listen" { go gj_server(argv[2:]); }
5889             if argv[2] == "server" { go gj_server(argv[2:]); }
5890             if argv[2] == "serve" { go gj_server(argv[2:]); }
5891             if argv[2] == "client" { go gj_client(argv[2:]); }
5892             if argv[2] == "join" { go gj_client(argv[2:]); }
5893         }
5894         comx := isinX("-c",argv)
5895         if 0 < comx {
5896             gshCtxBuf,err := setupGshContext()
5897             gsh := &gshCtxBuf
5898             if !err {
5899                 gsh.gshellv(argv[comx+1:])
5900             }
5901             return
5902         }
5903     }
5904     if 1 < len(argv) && isin("-s",argv) {
5905     }else{
5906         gsh.showVersion(append(argv,[]string{"-l","-a"}...))
5907     }
5908     script(nil)
5909     //gshCtx := script(nil)
5910     //gshellll(gshCtx,"time")
5911 }
5912
5913 //</div></details>
5914 //<details id="gsh-todo"><summary>Considerations</summary><div class="gsh-src">
5915 // - inter gsh communication, possibly running in remote hosts -- to be remote shell
5916 // - merged histories of multiple parallel gsh sessions
5917 // - alias as a function or macro
5918 // - instant alias and environ export to the permanent > -./gsh/gsh-alias and gsh-environ
5919 // - retrieval PATH of files by its type
5920 // - gsh as an IME with completion using history and file names as dictionaies
5921 // - gsh a scheduler in precise time of within a millisecond
5922 // - all commands have its subucomand after "---" symbol
5923 // - filename expansion by "-find" command
5924 // - history of ext code and output of each commoand
5925 // - "script" output for each command by pty-tee or telnet-tee
5926 // - $BUILTIN command in PATH to show the priority
5927 // - "?" symbol in the command (not as in arguments) shows help request
5928 // - searching command with wild card like: which ssh-*
5929 // - longformat prompt after long idle time (should dismiss by BS)
5930 // - customizing by building plugin and dynamically linking it
5931 // - generating syntactic element like "if" by macro expansion (like CPP) >> alias
5932 // - "!" symbol should be used for negation, don't wast it just for job control
5933 // - don't put too long output to tty, record it into GSH_HOME/session-id/comand-id.log
5934 // - making canonical form of command at the start adding quotation or white spaces
5935 // - name(a,b,c) ... use "(" and ")" to show both delimiter and realm
5936 // - name? or name! might be useful
5937 // - htar format - packing directory contents into a single html file using data scheme
5938 // - filepath substitution should be done by each command, expecially in case of builtins
5939 // - @N substition for the history of working directory, and @spec for more generic ones
5940 // - @dir prefix to do the command at there, that means like (chdir @dir; command)
5941 // - GSH_PATH for plugins
5942 // - standard command output: list of data with name, size, resouce usage, modified time
5943 // - generic sort key option -nm name, -sz size, -ru rusage, -ts start-time, -tm mod-time
5944 // -wc word-count, grep match line count, ...
5945 // - standard command execution result: a list of string, -tm, -ts, -ru, -sz, ...
5946 // - -tailf-filename like tail -f filename, repeat close and open before read
5947 // - max. size and max. duration and timeout of (generated) data transfer
5948 // - auto. numbering, aliasing, IME completion of file name (especially rm of quieer name)
5949 // - IME "?" at the top of the command line means searching history
5950 // - IME %d/0x10000/ %x/ffff/
5951 // - IME ESC to go the edit mode like in vi, and use :command as :s/x/y/g to edit history
5952 // - gsh in WebAssembly

```



```

5953 // - gsh as a HTTP server of online-manual
5954 //---END--- (^-^)//ITS more</div></details>
5955
5956 //<span class="gsh-golang-data">
5957
5958 var WorldDic = //<span id="gsh-world-dic">
5959 "data:text/dic;base64,+
5960 "Iy87Xk1JUTUwVW44LWJlEg6L6e5pu4lCgyMDIwLTA4MTlhKQpZ2WthaSDkuJbnlYwKa28g44GT"+
5961 "Cm5uIOOckwpuaSDjgKY2hp100BoQp0aSDjgaEKaGg44GvCnLI00BmwrYSDjYsKaSDj"+
5962 "gYQK";
5963 //</span>
5964
5965 var WnnDic = //<span id="gsh-wnn-dic">
5966 "data:text/dic;base64,+
5967 "PglldEGyZ2hcnNldD0iVVRGLTgiPgo8dGV4dGFyZWEgY29scz04MCRyY3dzPTQwPgovL2Rp"+
5968 "Y3ZlcglHU2h1bGxccc0lNRVxzZG1jdG1vbmFyeVxzZm9yXHNHbW5ccy8vXHM5MDIwLTA4MzAK"+
5969 "RlNoZ2xwX2h1bGxccc0lNRVxzZG1jdG1vbmFyeVxzZm9yXHNHbW5ccy8vXHM5MDIwLTA4MzAK"+
5970 "44G+44GICeWjJgYUW1hZQnlKI3l1Y0K44Gq44G44GuCeS4remHjgpuYwThbm8J5Lit"+
5971 "6YeOcnhdCeOcjwP0YQnjgZ8Kc2KJ44GXCNNoaQnjgZcKbM8J44GuCm5hCeOBgqptYQnjg4K"+
5972 "ZQnjgYgKaEJ44GvCm5hCeOBgqptYQnjgYsKbM8J44GuCm5hCeOBgqptYQnjgZKZVxzCWJj"+
5973 "aG8KZG1jCWRpIwplY2hvcWVjaG8KcmVwGf5CXJlCGheqpyZXB1YXQJcmVwZWFOCmR0CWrh"+
5974 "dGVccysnYklsbSVkLVSIOiVWoiVWp0aW9uCXRp24KJXQJLJy8gdG8gYmUgTW4gYWN0"+
5975 "a89uCjwvdGV4dGFyZWE+CG==";
5976 //</span>
5977
5978 var SumomoDic = //<span id="gsh-sumomo-dic">
5979 "data:text/dic;base64,+
5980 "PglldEGyZ2hcnNldD0iVVRGLTgiPgo8dGV4dGFyZWEgY29scz04MCRyY3dzPTQwPgovL3Zl"+
5981 "cg1HU2h1bGxccc0lNRVxzZG1jdG1vbmFyeVxzZm9yXHNHbW1vbW9ccy8vXHM5MDIwLTA4MzAK"+
5982 "c3UJ44G2CmlCeOCgpubwnjga4KdQnjgYKY2hpCeOB0Qp0aQnjgjaEKdWNoaQnlh0KdXRP"+
5983 "CeWChQpzd1vbW8J44G244KC44KCCN1bW9tb21vCeOBmeOCguOCguCgqptb21vCeahgwpt"+
5984 "b21vbW8J5Gd44KCCiwsCeOAgQouLgnjgIIFKPC902Xh0YXJlYT4K";
5985 //</span>
5986
5987 var SijimiDic = //<span id="gsh-sijimi-dic">
5988 "data:text/dic;base64,+
5989 "PglldEGyZ2hcnNldD0iVVRGLTgiPgo8dGV4dGFyZWEgY29scz04MCRyY3dzPTQwPgovL3Zl"+
5990 "cg1HU2h1bGxccc0lNRVxzZG1jdG1vbmFyeVxzZm9yXHNHbW1vbW9ccy8vXHM5MDIwLTA4MzAK"+
5991 "CmNpCeOB1wpaZG44GXcmprCeOBmPtaQnjgBkRmMj44GqCmp1CeOBmOOCOp4eXUJ44KF"+
5992 "CmUJ44G2CmlCeOBgvrBwmjgZMKYnuJ44G2Cm5uCeOCkwpubwnjga4KY2hpCeOB0Qp0aQnjg"+
5993 "gaEKa2EJ44G2CmlCeOCiQoslAnjgIEKLi4J44CCNhuY5hCeS4qvp4anV1CeNngOp4hmkJ"+
5994 "5lqCmteveAnlgsKa29xCeWaiwprb3gJ5YCLCm5hbmFqdXVuaXgJnZiKbmfUwpldW5peHJg"+
5995 "77y77yScm5hbmFqdXVuaVgJ77Y77yScuS4g+WngeS6jHgJnZiKa29idW5uCeWai+WlHgp0"+
5996 "awthcmFxCeOB0eBi+OCiQp0aWthcmEJ5YqbcMnoaWthcmEJ5YqbcjwvdGV4dGFyZWE+CG="
5997 //</span>
5998
5999 var JA_JKLDic = //<span id="gsh-ja-jkl-dic">
6000 "data:text/dic;base64,+
6001 "Iy92ZJscU15SUIFamRyY2ptb3JzZWpKQWpKS0w0mJyAyMGowODE5KShelV4pL1NhdG94SVRT"+
6002 "CmtqamprBtGta2tsa2psIOS44ueVjApqamtgamwJ44GCmtqbanjgYQKa2tqbAnjgYKamtq"+
6003 "amwJ44GICmtq2trbAnjgYKa2pra2wJ44GLCmpraMrbAnjgY0Ka2tramwJ44GCmpraMps"+
6004 "CeOBkQpqaMqbAnjgZMkamtqa2psCeOB1QpqaMqa2wJ44GXcmprqamqbAnjgZKa2pqamts"+
6005 "CeOBmmpqamprBmRanJgZ0KamtsCeOBnwpra2prbAnjgjaEKa2pqa2wJ44GKcmqga2pqbAnjgYaK"+
6006 "a2tqa2tCeOBqApramtCeOBqppqa2prbAnjgjaEKa2tra2wJ44G6Cmpga2psCeOBpQpra2pq"+
6007 "bAnjga4Kamra2wJ44G6Cmpqa2tqbAnjgBkKampra2wJ44G1CmtsCeOBuApgga2tsCeOBuwpq"+
6008 "a2tqbAnjgBkKamra2psCeOBwprBmRanjgjaEKamra2psCeOCQpqa2tqga2wJ44KCCmtgamwJ"+
6009 "44KCCmpra2pqbAnjgYKampsCeOCiApra2tsCeOCiQpqaMtsCeOCiQpqa2pqa2wJ44KLCmpq"+
6010 "amwJ44KCCmqa2psCeOCjQpqa2psCeOCjvpramtramwJ44KQCCmtqamrbAnjgBkKa2pqaMwJ"+
6011 "44KCCmtqa2prbAnjgpmKa2pqa2psCeODvApra2wJ44KbCmtramprbAnjgpmKa2pramtqbAnjg"+
6012 "gIEK";
6013 //</span>
6014
6015 //</span>
6016 /*
6017 <style id="gsh-references-style">
6018 #references details { font-family:Georgia; }
6019 #references a { font-family:Georgia; }
6020 .wrap { white-space:normal; }
6021 </style>
6022 <details id="references"><summary>References</summary><div class="gsh-src">
6023 Web technology
6024 <a href="https://html.spec.whatwg.org">HTML: The Living Standard</a> (September 2020)
6025 <a href="https://html.spec.whatwg.org/dev/">Developer Version</a>
6026
6027 <a href="https://drafts.csswg.org">CSS Working Group Editor Drafts</a> (September 2020)
6028
6029 <a href="https://developer.mozilla.org/ja/docs/Web">MDN web docs</a>
6030 <a href="https://developer.mozilla.org/ja/docs/Web/HTML/Element">HTML</a>
6031 <a href="https://developer.mozilla.org/en-US/docs/Web/CSS">CSS</a> : <span class="wrap">
6032 <a href="https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors">selectors</a>
6033 <a href="https://developer.mozilla.org/en-US/docs/Web/CSS/background-repeat">repeat</a></span>
6034 <a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript">JavaScript</a>
6035 <a href="https://developer.mozilla.org/en-US/docs/Web/HTTP">HTTP</a>
6036 <a href="https://mdn-web-dna.s3-us-west-2.amazonaws.com/MDN-Browser-Compatibility-Report-2020.pdf">MDN Browser Compatibility Report (2020)</a>
6037
6038 Go language (August 2020 / Go 1.15)
6039 <a href="https://golang.org">The Go Programming Language</a>
6040 <a href="https://golang.org/pkg/">Packages</a>
6041 <a href="https://godoc.org/golang.org/x/net/websocket">WebSocket</a>
6042
6043 <a href="https://stackoverflow.com/">Stackoverflow</a>
6044 <!--
6045 <iframe src="https://golang.org" width="100%" height="300"></iframe>
6046 -->
6047 </div></details>
6048 */
6049 /*
6050 <details id="html-src" onclick="frame_open();"><summary>Raw Source</summary><div>
6051
6052 <!-- h2>The full of this HTML including the Go code is here.</h2 -->
6053 <details id="gsh-whole-view"><summary>Whole file</summary>
6054 <a name="whole-src-view"></a>
6055 <span id="src-frame"></span><!-- a window to show source code -->
6056 </details>
6057
6058 <details id="gsh-style-frame" onclick="fill_CSSView()"><summary>CSS part</summary>
6059 <a name="style-src-view"></a>
6060 <span id="gsh-style-view"></span>
6061 </details>
6062
6063 <details id="gsh-script-frame" onclick="fill_JavaScriptView()"><summary>JavaScript part</summary>
6064 <a name="script-src-view"></a>
6065 <span id="gsh-script-view"></span>
6066 </details>
6067
6068 <details id="gsh-data-frame" onclick="fill_DataView()"><summary>Builtin data part</summary>
6069 <a name="gsh-data-frame"></a>
6070 <span id="gsh-data-view"></span>
6071 </details>
6072
6073 </div></details>
6074 */
6075
6076 /*

```

```

6077 <div id="GshFooter0"></div>
6078 <!-- 2020-09-17 SatoxITS, visible script { -->
6079 <details><summary>GJScript</summary>
6080 <style> gjscript { font-family:Georgia; }</style>
6081 <pre id="gjscript_1" class="gjscript"> function gjtest1(){ alert('Hello GJScript!'); }
6082 gjtest1()
6083 </pre>
6084 <script>
6085   gjs = document.getElementById('gjscript_1');
6086   //eval(gjs.innerHTML);
6087   //gjs.outerHTML = "";
6088 </script>
6089 </details><!-- ----- END-OF-VISIBLE-PART ----- } -->
6090
6091 <!--
6092 // 2020-0906 added,
6093 https://developer.mozilla.org/en-US/docs/Web/CSS/z-index
6094 https://developer.mozilla.org/en-US/docs/Web/CSS/position
6095 -->
6096 <span id="GshGrid">(^_^)</small><small><small>{Hit j k l h}</small></span>
6097
6098 <span id="GStat"><br>
6099 </span>
6100 <span id="GMenu" onclick="GShellMenu(this)" draggable="true"></span>
6101 <span id="GTop"></span>
6102 <div id="GShellPlane" onclick="showGShellPlane();"></div>
6103 <div id="RawTextViewer"></div>
6104 <div id="RawTextViewerClose" onclick="hideRawTextViewer()"> CLOSE </div>
6105
6106 <style id="GshStyleDef">
6107 #LineNumbered table,tr,td {
6108   margin:0;
6109   padding:4px;
6110   spacing:0;
6111   border:12px;
6112 }
6113 textarea.LineNumber {
6114   font-size:12px;
6115   font-family:monospace,Courier New;
6116   color:#282;
6117   padding:4px;
6118   text-align:right;
6119 }
6120 textarea.LineNumbered {
6121   font-size:12px;
6122   font-family:monospace,Courier New;
6123   padding:4px;
6124   wrap:off;
6125 }
6126 #RawTextViewer{
6127   z-index:0;
6128   position:fixed; top:0px; left:0px;
6129   width:100%; xxxheight:50px; xheight:0px;
6130   overflow:auto;
6131   color:#fff; background-color:rgba(128,128,256,0.2);
6132   font-size:12px;
6133   spellcheck:false;
6134 }
6135 #RawTextViewerClose{
6136   z-index:0;
6137   position:fixed; top:-100px; left:-100px;
6138   color:#fff; background-color:rgba(128,128,256,0.2);
6139   font-size:20px; font-family:Georgia;
6140   white-space:pre;
6141 }
6142 #xxxGShellPlane{
6143   z-index:0;
6144   position:fixed; top:0px; left:0px;
6145   width:100%; height:50px;
6146   overflow:auto;
6147   color:#fff; background-color:rgba(128,128,256,0.3);
6148   font-size:12px;
6149 }
6150 #xxxGTop{
6151   z-index:9;
6152   opacity:1.0;
6153   position:fixed; top:0px; left:0px;
6154   width:320px; height:20px;
6155   color:#fff; background-color:rgba(32,32,160,0.15);
6156   color:#fff; font-size:12px;
6157 }
6158 #xxxGPos{
6159   z-index:12;
6160   position:fixed; top:0px; left:0px;
6161   opacity:1.0;
6162   width:640px; height:30px;
6163   color:#fff; background-color:rgba(0,0,0,0.2);
6164   color:#fff; font-size:12px;
6165 }
6166 #GMenu{
6167   z-index:100000000;
6168   position:fixed; top:250px; left:0px;
6169   opacity:1.0;
6170   width:100px; height:100px;
6171   color:#fff;
6172   color:#fff; background-color:rgba(0,0,0,0.0);
6173   color:#fff; font-size:16px; font-family:Georgia;
6174   background-repeat:no-repeat;
6175 }
6176 #xxxGStat{
6177   z-index:8;
6178   xopacity:0.0;
6179   position:fixed; top:20px; left:0px;
6180   xwidth:640px;
6181   width:100%; height:90px;
6182   color:#fff; background-color:rgba(0,0,128,0.04);
6183   font-size:20px; font-family:Georgia;
6184 }
6185 #GLog{
6186   z-index:10;
6187   position:fixed; top:50px; left:0px;
6188   opacity:1.0;
6189   width:640px; height:60px;
6190   color:#fff; background-color:rgba(0,0,128,0.10);
6191   font-size:12px;
6192 }
6193 #GshGrid {
6194   z-index:11;
6195   xopacity:0.0;
6196   position:fixed; top:0px; left:0px;
6197   width:320px; height:30px;
6198   color:#9f9; font-size:16px;
6199 }
6200 xbody {display:none;}

```

```

6201 .gsh-link{color:green;}
6202 #gsh {border-width:1;margin:0;padding:0;}
6203 #gsh {font-family:monospace,Courier New;color:#ddf;font-size:8px;}
6204 #gsh header{height:100px;}
6205 #gsh header{height:100px;background-image:url(GShell-Logo00.png);}
6206 #GshMenu{font-size:14pt;color:#c44;}
6207 .GshMenu{
6208   font-size:14pt;color:#2a2;padding:4px; text-align:right;
6209 }
6210 .GshMenu: hover{
6211   font-size:14pt;color:#fff;font-weight:bold;background-color:#2a2;
6212 }
6213 #GshFooter{height:100px;background-size:80px;background-repeat:no-repeat;}
6214 #gsh note{color:#000;font-size:10pt;}
6215 #gsh h2{color:#24a;font-family:Georgia;font-size:18pt;}
6216 #gsh h3{color:#24a;font-family:Georgia;font-size:16pt;}
6217 #gsh details{color:#888;background-color:#fff;font-family:monospace;}
6218 #gsh summary{font-size:16pt;color:#fff;background-color:#8af;xxxheight:30px;}
6219 #gsh summary{font-size:16pt;color:#fff;
6220   padding:2pt;
6221   line-height:1.0;
6222   vertical-align:middle;
6223   xxx-background-color:#8af;
6224   background-color:#6881AD;xxx-PBlue;
6225   xxxheight:30px;
6226 }
6227 #gsh pre{font-size:11pt;color:#223;background-color:#faaaaa;}
6228 #gsh a{color:#24a;}
6229 #gsh a[name]{color:#24a;font-size:16pt;}
6230 #gsh .gsh-src{white-space:pre;font-family:monospace,Courier New;font-size:11pt;}
6231 #gsh .gsh-src{background-color:#faaaaa;color:#223;}
6232 #gsh-src-src{spellcheck:false}
6233 #SrcTextarea{white-space:pre;font-family:Courier New;font-size:10pt;}
6234 #SrcTextarea{background-color:#faaaaa;color:#223;}
6235 .gsh-code {white-space:pre;font-family:Courier New !important;}
6236 .gsh-code {color:#024;font-size:11pt; background-color:#faaff;}
6237 .gsh-golang-data {display:none;}
6238 #gsh-WinId {color:#000;font-size:14pt;}
6239
6240 .gsh-document {font-size:11pt;background-color:#fff;font-family:Georgia;}
6241 .gsh-document {color:#000;background-color:#fff !important;}
6242 .gsh-document > h2{color:#000;background-color:#fff !important;}
6243 .gsh-document details{color:#000;background-color:#fff;font-family:Georgia;}
6244 .gsh-document p{max-width:550pt;color:#000;background-color:#fff;font-family:Georgia;}
6245 .gsh-document address{width:500pt;color:#000;background-color:#fff;font-family:Georgia;}
6246
6247 @media print {
6248   #gsh pre{font-size:11pt !important;}
6249 }
6250 </style>
6251
6252 <!--
6253 // Logo image should be drawn by JavaScript from a meta-font.
6254 // CSS seems not follow line-splitted URL
6255 -->
6256 <script id="gsh-data">
6257 //GShellLogo="QR-ITS-more.jp.png"
6258 GShellLogo="data:image/png;base64,\
6259 iVBORw0KGgoAAAANSUHEUgAAQAAEAAAB/CAYAAADvs3f4AAAAAXNSR0IARs4c6QAAAHlWE1m\
6260 TU0AKgAAAgAABAFAAAUAAABAAAPgEBAUAAABAAABAAARgEoAMAAABAAIAADpAAQAAAB\
6261 AAAATgAAAAABIAAAQAQAAEgAAABAAQAAQADAAAAQABAAcAgAEAAAAQAAAGGAAwAE\
6262 AAAAAQAAAH8AAAAAYx1BhgAAAAwSFLzAAALeWAAcXMBAAJcGAAAF3RJRFEUeAhtnQuUFWNz\
6263 x+tTukZ3iCggO/jY6Osb8WgmZAvn7uG4+blSTR7YnQXQdQpCK72anWLD2MSlRkeUaPnoCdu\
6264 4iuJX7jrIY25DODGmF2VqIBeiSggCoiMMA+mu+vu//ZMD9U1dau62aUbv91GRK93vvd6/g\
6265 fnXvdxstBA8SIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6266 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6267 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6268 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6269 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6270 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6271 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6272 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6273 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6274 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6275 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6276 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6277 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6278 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6279 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6280 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6281 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6282 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6283 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6284 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6285 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6286 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6287 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6288 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6289 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6290 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6291 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6292 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6293 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6294 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6295 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6296 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6297 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6298 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6299 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6300 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6301 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6302 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6303 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6304 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6305 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6306 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6307 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6308 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6309 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6310 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6311 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6312 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6313 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6314 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6315 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6316 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6317 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6318 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6319 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6320 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6321 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6322 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6323 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6324 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\

```

6325 wKgZ9FWrITCRJwJwh5+/ocSLzQ1zG52BvItG+wOpqXRYEwaRf rdbSgC5bD/PySxBhAKPW0\
6326 qzZ9y4L10uAB4kx5we8qdsHO6++b0nwjzFXyAUyVyeEce00117SAZkx0UgxtmZB9RcaVyx\
6327 ZCMBHj3dTCzUwWyKriwy4my7H9zt3R93/8XlJ0ESWety7q7j1J0dKxhMFAZK06D1Ww6e6\
6328 H52hUwLALQHQQ0ZyW66yztL57r9u40VJg4JBMJCaYhRtYexX8/cxu+rux9L5yef50a+W\
6329 Bv0wN2Zxrw7VADPzCEdPkpdsLx0KoferrvEM+yj7aEaA'yxzZjXm+61FzU146c7h0d6Q/m\
6330 Wncf9BtVxbs623hXpV1wLkjhJUBTFkRRbagLQCWibwuiiPykYhLhwZaq8YKoeMcji9Y9Ly\
6331 Pwk79U/55Bk75fSXMcwhj79Y5xv7q8YspvTbSG+55hdjnj6V6S6ErfyqVOL2xoeLrbmW\
6332 YwkqG5S2p1OK5dZjgs+2L1B426/gG+uosa6yuy0Y1jzCuoG41l1gVQ0Yep1w1uXUL4pPR\
6333 zD3GL6w1VE4jA35xePk1NLSuBb/34RcwB6JXGz6zrflBBjBbJH71LwBGDRVdb4ieXpghPb\
6334 NQT31qMhZ7HTvHrXnv45r8FpFWRNd1qVfV2qBlxFL16+rgDLV82CTnVbIDb2JfBpwMJP\
6335 aW3rYbqng9XMLmChjCvN5F5KMRc2LbzJbk8mU55cn4x/zrLdJQzN1rKkyu01pdgqccf\
6336 gKqP/ahfXooVi+JtoFimZuYjn8F70HhMAMxAdAeUX6cF7075Ukuyg50z33vV/20C7b+ach\
6337 Ltntp1tH3VeW84ipGt4JWAnu7Pn5xwvjx841MabC3Q8rFlzPCF7cOSF0b8NadzSfWqfFhB\
6338 nmdjITHGh3eSrt+42Mk5KwTxFMe35RJTVorP3rnm49VM0qfP8oiD191X61dvbXmkqjvb\
6339 NfydX9m8wim2MLKZeSL/VzQSkDpZcdCycyte71q/B4XKfQaNeK3mL4r29FQL/ga+/vrEO\
6340 gDTT00U9WUKVmf9MYULzjVpZxxu0FP00/pTdehOd/1XXxGZawfuXp6eG1lz+eme2X9lbo\
6341 oxxU119F0bLaGgQhafa5NVPhxjK7X0gLu0MRm+JAFesnnaKzLRhZXLyBf5ediUwKc1/wD7\
6342 FD+JL72vEtDPEIggWkzj6zP/5Dduzt+2HihxkFLnhsTumT011AJkyVScemp1WALAACzAB\
6343 dqV2Sx/S+nLNdPeLXVtD/skUr+JL5/9vsbl75z+bYNS8Q2EuQN/Oa3x1/FJZs/VZ30EGeGB\
6344 ePdtCYCRORKr3q6VL0pOf7KXvDaaZvc6jQECZx56CYcmz2/7CyuWar21INzXK4NOC075/\
6345 4yHTRK3KuyfEGjgmxT/xdbpt8uSR17L1luoF7GcmJUL7cXkfyqVfVpwpV9RPAeh07FRV\
6346 hU469cYkYkFX0cCyovrTRXZy1h/y3n7fii1bretHtSuVUMrtjpkRWYmKzEmdZPcIm\
6347 dM1f6+eW10/65LmCDDZYPFL2fDeycqj38aRabQSPK1sCGUCaKrd0UyzaauvgzK6zAVTE\
6348 LLQqFLXPJfjYjthCkphr+n++76LoLJid3d451+sndV9Yr4veCwQ9+SrXtX6G/arezLAWK\
6349 tgzv7Wk4n+28f/FFzUKIA3ky5ULm09CE8N3HgLin15rRnY32hsXoRntBmBvWmiPzT03\
6350 j0g8vnn35zccfYlYgCmlw2/fvCjoXytlieoL0xvRghMyN21/IJtL6W3j5y8j+71dyU57\
6351 xLJDmJm+x0QgtrucgEUTDvIPFcnovWaf2KAEvArG5T3tjBQGT+5rCIU+1BzxFIPJumpRVP\
6352 4YEuz9wP9x1fvw/0ppuyxDP9uNPYih91/XNkovNSd5GG8wms31CzfrkCQUTZSHj+wm8Q\
6353 JV7E3Xm6WjLsr6LV668T0ExtHjJ/4Cwd24+uzFvsJrsT11RkFoOALtznFzdf2SD12QrO\
6354 8Y5V88pdsb0VhRLQD6exvrE0j9y4g9DPKc52mjjz021LD7y3zFL8qmsDMOFARTVFWC3\
6355 N1NQGWk1EavG0Mrz78D2VefmHcdPFC86nbFB55KF1FMRHE6F0S0AtOm/d8V98km7D\
6356 C58YrseFLvLspLpbx79z64erd2NyuNLKileJdalUak7j0orr315x+YA9CbQBDf/cK7JkHdB\
6357 E5sg690KMH9pRjdv63vgEvYbdQucoc1VM9n0/QaPP3K21ve8zCwmJk30kx+30RKQEKiW\
6358 blxaFhe29Jg8of8Gkam659PmDGP5bmUkpmc22TR7BHSKjP0kMktCf/KAMsOJXtjE\
6359 v7q+/OzmZbN/25IoHT3+NPgZn2eyx7u120JDM9xoyTzB70ya+vndqW3URPijYxmbOe1/au\
6360 zq4BeygslmpGdL1KxcmlwXskz8Gwa940stveB+f71411K30105Mkod+r/TZvX0P9Ec3\
6361 xp7KQYU8JGTmcat+NeI/99v3xbh+21bh03cnot1JdfCznzkeapSDW/vjDg4XP4Kcnb8+W9p\
6362 9ezduK2Q3Fev051yqtcom30pzk9E5cs80r+FY150rCzr5HkPFMGaaQ3yAO9PydFDjJ\
6363 ppf5kjMq6rmYi3DfYK151400Kj1zezhBJ9NtWfBAGvUoLIaw82XvTahafS0dfpseeA\
6364 nRiFLK0m8Kxmp/Fby6aVgZft55SkWno2mPFSF3gqf3o4UGSj/v1548wLfbVvab720b\
6365 Xx/Mrwglf9zrXPQMBx5CiAfjHlyYjhsR7BKmfG8MtD+3CdJF2god1vNN334d60h7hyf\
6366 koSvF0pEpkZFeqJWQt1d706cdnp1H7zi0z933h0LHWYU1RrhZ7ptxeVe69XWH+3Jdasmf0\
6367 iEwS1G5j8EaJ2NR0adga7iEVOR2LBSccVCR80U5UelJbSpxVgHEcusJrKkYLW0VSSUinTmW\
6368 LaycFXhpSwIKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiK\
6369 QAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiK\
6370 QAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiK\
6371 QAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiK\
6372 QAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiK\
6373 QAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiKQAiK\
6374 m38w0cAAAAASUVORK5CYII=";





```

6697     color:#df8 !important;
6698     background-color:rgba(32,32,32,1.0) !important;
6699     line-height:0.0;
6700 }
6701 .GJWin{
6702     z-index:10000;
6703     display:inline;
6704     position:relative;
6705     flex-wrap: wrap;
6706     top:0; left:0px;
6707     width:285px !important; height:205px !important;
6708     border:1px solid #eea; border-radius:2px;
6709     margin:0px; padding:0px;
6710     font-size:8pt;
6711     line-height:0.0;
6712     color:#fff; background-color:rgba(0,0,64,0.1) !important;
6713 }
6714 .GJTab{
6715     display:inline;
6716     position:relative;
6717     top:0px; left:0px;
6718     margin:0px; padding:2px;
6719     border:0px solid #000; border-radius:2px;
6720     width:90px; height:20px;
6721     font-family:Georgia;
6722     font-size:9pt;
6723     line-height:1.0;
6724     white-space:nowrap;
6725     color:#fff; background-color:rgba(0,0,64,0.7);
6726     text-align:center;
6727     vertical-align:middle;
6728 }
6729 .GJStat:focus{
6730     color:#df8 !important;
6731     background-color:rgba(32,32,32,1.0) !important;
6732     line-height:1.0;
6733 }
6734 .GJStat{
6735     display:inline;
6736     position:relative;
6737     top:0px; left:0px;
6738     margin:0px; padding:2px;
6739     border:0px solid #00f; border-radius:2px;
6740     width:166px; height:20px;
6741     font-family:monospace;
6742     font-size:9pt;
6743     line-height:1.0;
6744     color:#fff; background-color:rgba(0,0,64,0.2);
6745     text-align:center;
6746     vertical-align:middle;
6747 }
6748 .GJIcon{
6749     display:inline;
6750     position:relative;
6751     top:0px; left:1px;
6752     border:2px solid #44a;
6753     margin:0px; padding:1px;
6754     width:13.2; height:13.2px;
6755     border-radius:2px;
6756     font-family:Georgia;
6757     font-size:13.2px;
6758     line-height:1.0;
6759     white-space:nowrap;
6760     color:#fff; background-color:rgba(32,32,160,0.8);
6761     text-align:center;
6762     vertical-align:middle;
6763     text-shadow:0px 0px;
6764 }
6765 .GJText:focus{
6766     color:#fff !important;
6767     background-color:rgba(32,32,160,0.8) !important;
6768     line-height:1.0;
6769 }
6770 .GJText{
6771     display:inline;
6772     position:relative;
6773     top:0px; left:0px;
6774     border:0px solid #000; margin:0px; padding:0px;
6775     width:280px; height:160px;
6776     border:0px;
6777     font-family:Courier New,monospace !important;
6778     font-size:8pt;
6779     line-height:1.0;
6780     white-space:pre;
6781     color:#fff; xbackground-color:rgba(0,0,64,0.5);
6782     background-color:rgba(32,32,128,0.8) !important;
6783 }
6784 .GJMode{
6785     display:inline;
6786     position:relative;
6787     top:0px; left:0px;
6788     border:0px solid #000; border-radius:0px;
6789     margin:0px; padding:0px;
6790     width:280px; height:20px;
6791     font-size:9pt;
6792     line-height:1.0;
6793     white-space:nowrap;
6794     color:#fff; background-color:rgba(0,0,64,0.7);
6795     text-align:left;
6796     vertical-align:middle;
6797 }
6798 </style>
6799
6800 <script id="gsh-script">
6801 // 2020-0909 added, permanent local storage
6802 // https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage
6803 var MyHistory = ""
6804 Permanent = localStorage;
6805 MyHistory = Permanent.getItem('MyHistory')
6806 if( MyHistory == null ){ MyHistory = "" }
6807 d = new Date()
6808 MyHistory = d.getTime()/1000+" "+document.URL+"\n" + MyHistory
6809 Permanent.setItem('MyHistory',MyHistory)
6810 //Permanent.setItem('MyWindow',window)
6811
6812 var GJLog_Win = null
6813 var GJLog_Tab = null
6814 var GJLog_Stat = null
6815 var GJLog_Text = null
6816 var GJWin_Mode = null
6817 var FProductInterval = 0
6818
6819 var GJ_FactoryID = -1
6820 var GJFactory = null

```

```

6821 if( e = document.getElementById('GJFactory_0') ){
6822   GJFactory_1.height = 0
6823   GJFactory = e
6824   e.setAttribute('class', 'GJFactory')
6825   var GJ_FactoryID = 0
6826 }else{
6827   GJFactory = GJFactory_1
6828   var GJ_FactoryID = 1
6829 }
6830
6831 function GJFactory_Destroy(){
6832   gjf = GJFactory
6833   //gjf = document.getElementById('GJFactory')
6834   //alert('gjf='+gjf)
6835   if( gjf != null ){
6836     if( gjf.childNodes != null ){
6837       for( i = 0; i < gjf.childNodes.length; i++ ){
6838         gjf.removeChild(gjf.childNodes[i])
6839       }
6840     }
6841     gjf.innerHTML = ''
6842     gjf.style.width = 0
6843     gjf.style.height = 0
6844     gjf.removeAttribute('style')
6845     GJLog_Win = GJLog_Tab = GJLog_Stat = GJLog_Text = GJWin_Mode = null
6846     window.clearInterval(FProductInterval)
6847     return '-- Destroy: work product destroyed'
6848   }else{
6849     return '-- Destroy: work product not exist'
6850   }
6851 }
6852
6853 var TransMode = false
6854 var onKeyControl = false
6855 var OnKeyShift = false
6856 var OnKeyAlt = false
6857 var OnKeyJ = false
6858 var OnKeyK = false
6859 var OnKeyL = false
6860
6861 function GJWin_OnKeyUp(ev){
6862   keycode = ev.code;
6863   if( keycode == 'ShiftLeft' ){
6864     OnKeyShift = false
6865   }else
6866   if( keycode == 'ControlLeft' ){
6867     onKeyControl = false
6868   }else
6869   if( keycode == 'AltLeft' ){
6870     OnKeyAlt = false
6871   }else
6872   if( keycode == 'KeyJ' ){ OnKeyJ = false }else
6873   if( keycode == 'KeyK' ){ OnKeyK = false }else
6874   if( keycode == 'KeyL' ){ OnKeyL = false }else
6875   {
6876   }
6877   ev.preventDefault()
6878 }
6879 function and(a,b){ if(a){ if(b){ return true; } return false; } }
6880 function GJWin_OnKeyDown(ev){
6881   keycode = ev.code;
6882   mode = ''
6883   key = ''
6884   if( keycode == 'ControlLeft' ){
6885     onKeyControl = true
6886     ev.preventDefault()
6887     return;
6888   }else
6889   if( keycode == 'ShiftLeft' ){
6890     OnKeyShift = true
6891     ev.preventDefault()
6892     return;
6893   }else
6894   if( keycode == 'AltLeft' ){
6895     ev.preventDefault()
6896     OnKeyAlt = true
6897     return;
6898   }else
6899   if( keycode == 'Backquote' ){
6900     TransMode = !TransMode
6901     ev.preventDefault()
6902   }else
6903   if( and(keycode == 'Space', OnKeyShift) ){
6904     TransMode = !TransMode
6905     ev.preventDefault()
6906   }else
6907   if( keycode == 'ShiftRight' ){
6908     TransMode = !TransMode
6909   }else
6910   if( keycode == 'Escape' ){
6911     TransMode = true
6912     ev.preventDefault()
6913   }else
6914   if( keycode == 'Enter' ){
6915     TransMode = false
6916     //ev.preventDefault()
6917   }
6918   if( keycode == 'KeyJ' ){ OnKeyJ = true }else
6919   if( keycode == 'KeyK' ){ OnKeyK = true }else
6920   if( keycode == 'KeyL' ){ OnKeyL = true }else
6921   {
6922   }
6923
6924   if( ev.altKey ){ key += 'Alt+' }
6925   if( onKeyControl ){ key += 'Ctrl+' }
6926   if( OnKeyShift ){ key += 'Shift+' }
6927   if( and(keycode != 'KeyJ', OnKeyJ) ){ key += 'J+' }
6928   if( and(keycode != 'KeyK', OnKeyK) ){ key += 'K+' }
6929   if( and(keycode != 'KeyL', OnKeyL) ){ key += 'L+' }
6930   key += keycode
6931
6932   if( TransMode ){
6933     //mode = "[\343\201\202r]"
6934     mode = "[\u202f]"
6935   }else{
6936     mode = '[---]'
6937   }
6938   ///// /gjmode.innerHTML = "[---]"
6939   GJWin_Mode.innerHTML = mode + ' ' + key
6940   //alert('Key:'+keycode)
6941   ev.stopPropagation()
6942   //ev.preventDefault()
6943 }
6944 function GJWin_OnScroll(ev){

```



```

6945     x = DragStartX = gsh.getBoundingClientRect().left.toFixed(0)
6946     y = DragStattY = gsh.getBoundingClientRect().top.toFixed(0)
6947     GJLog_append('OnScroll: x='+x+',y='+y)
6948 }
6949 document.addEventListener('scroll',GJWin_OnScroll)
6950 function GJWin_OnResize(ev){
6951     w = window.innerWidth
6952     h = window.innerHeight
6953     GJLog_append('OnResize: w='+w+',h='+h)
6954 }
6955 window.addEventListener('resize',GJWin_OnResize)
6956
6957 var DragStartX = 0
6958 var DragStartY = 0
6959 function GJWin_DragStart(ev){
6960     // maybe this is the grabbing position
6961     this.style.position = 'fixed'
6962     x = DragStartX = this.getBoundingClientRect().left.toFixed(0)
6963     y = DragStattY = this.getBoundingClientRect().top.toFixed(0)
6964     GJLog_Stat.value = 'DragStart: x='+x+',y='+y
6965 }
6966 function GJWin_Drag(ev){
6967     x = ev.clientX; y = ev.clientY // x = ev.pageX; y = ev.pageY
6968     this.style.left = x - DragStartX
6969     this.style.top = y - DragStartY
6970     this.style.zIndex = '30000'
6971     this.style.position = 'fixed'
6972     x = this.getBoundingClientRect().left.toFixed(0)
6973     y = this.getBoundingClientRect().top.toFixed(0)
6974     GJLog_Stat.value = 'x='+x+',y='+y
6975     ev.preventDefault()
6976     ev.stopPropagation()
6977 }
6978 function GJWin_DragEnd(ev){
6979     x = ev.clientX; y = ev.clientY
6980     //x = ev.pageX; y = ev.pageY
6981     this.style.left = x - DragStartX
6982     this.style.top = y - DragStartY
6983     this.style.zIndex = '30000'
6984     this.style.position = 'fixed'
6985     if( true ){
6986         console.log("Dropped: "+this.nodeName+'#'+this.id+' x='+x+' y='+y
6987             +' parent='+this.parentNode.id)
6988     }
6989     x = this.getBoundingClientRect().left.toFixed(0)
6990     y = this.getBoundingClientRect().top.toFixed(0)
6991     GJLog_Stat.value = 'x='+x+',y='+y
6992     ev.preventDefault()
6993     ev.stopPropagation()
6994 }
6995 function GJWin_DragIgnore(ev){
6996     ev.preventDefault()
6997     ev.stopPropagation()
6998 }
6999 // 2020-09-15 let every object have console view!
7000 var GJ_ConsoleID = 0
7001 var PrevReport = new Date()
7002 function GJLog_StatUpdate(){
7003     txa = GJLog_Stat;
7004     if( txa == null ){
7005         return;
7006     }
7007     tmLap0 = new Date();
7008     p = txa.parentNode;
7009     pw = txa.getBoundingClientRect().width;
7010     ph = txa.getBoundingClientRect().height;
7011     //txa.value += '#'+p.id+' pw='+pw+' ph='+ph+'\n';
7012     tx1 = '#'+p.id+' pw='+pw+' ph='+ph+'\n';
7013
7014     w = txa.getBoundingClientRect().width;
7015     h = txa.getBoundingClientRect().height;
7016     //txa.value += 'w='+w+', h='+h+'\n';
7017     tx1 += 'w='+w+', h='+h+'\n';
7018
7019     //txa.value += '\n';
7020     //txa.value += DateShort() + '\n';
7021     tx1 += '\n';
7022     tx1 += DateShort() + '\n';
7023     tmLap1 = new Date();
7024
7025     txa.value += tx1;
7026     tmLap2 = new Date();
7027
7028     // vertical centering of the last line
7029     sHeight = txa.scrollHeight - 30; // depends on the font-size
7030     tmLap3 = new Date();
7031
7032     txa.scrollTop = sHeight; // depends on the font-size
7033     tmLap4 = new Date();
7034
7035     now = tmLap0.getTime();
7036     if( PrevReport == 0 || 10000 <= now-PrevReport ){
7037         PrevReport = now;
7038         console.log('StatBarUpdate:
7039             + 'leng=' + txa.value.length + ' byte, '
7040             + 'time=' + (tmLap4 -tmLap0) + 'ms { '
7041             + 'tadd=' + (tmLap2 -tmLap1) + ', '
7042             + 'hcal=' + (tmLap3 -tmLap2) + ', '
7043             + 'sctl=' + (tmLap4 -tmLap3) + '}'
7044         );
7045     }
7046 }
7047 GJWin_StatUpdate = GJLog_StatUpdate;
7048 function GJ_showTime1(wid){
7049     //e = document.getElementById(wid);
7050     //console.log(wid.id+'.value.length='+wid.value.length)
7051     if( e != null ){
7052         //e.value = DateShort();
7053     }else{
7054         // should remove the Listener
7055     }
7056 }
7057 function GJWin_OnResizeTextarea(ev){
7058     this.value += 'resized: ' + '\n'
7059 }
7060 function GJ_NewConsole(wname){
7061     wid = wname + '_' + GJ_ConsoleID
7062     GJ_ConsoleID += 1
7063
7064     GJFactory.style.setProperty('width',360+'px'); //GJFsize
7065     GJFactory.style.setProperty('height',320+'px')
7066     e = GJFactory;
7067     console.log('GJFa #' +e.id+' from w='+e.style.width+', h='+e.style.height)
7068 }

```

```

7069 if( GJFactory.innerHTML == "" ){
7070   GJFactory.innerHTML = '<'+H3>GJ Factory_+ GJ_FactoryID +'<'+/H3><'+hr>\n'
7071 }else{
7072   GJFactory.innerHTML += '<'+hr>\n'
7073 }
7074
7075 gjwin = GJLog_Win = document.createElement('span')
7076 gjwin.id = wid
7077 gjwin.setAttribute('class','GJWin')
7078 gjwin.setAttribute('draggable','true')
7079 gjwin.addEventListener('dragstart',GJWin_DragStart)
7080 gjwin.addEventListener('drag',GJWin_Drag)
7081 gjwin.addEventListener('dragend',GJWin_Drag)
7082 gjwin.addEventListener('dragover',GJWin_DragIgnore)
7083 gjwin.addEventListener('dragenter',GJWin_DragIgnore)
7084 gjwin.addEventListener('dragleave',GJWin_DragIgnore)
7085 gjwin.addEventListener('dragexit',GJWin_DragIgnore)
7086 gjwin.addEventListener('drop',GJWin_DragIgnore)
7087 gjwin.addEventListener('keydown',GJWin_OnKeyDown)
7088
7089 gjtab = GJLog_Tab = document.createElement('textarea')
7090 gjtab.addEventListener('keydown',GJWin_OnKeyDown)
7091 gjtab.style.readonly = true
7092 gjtab.contenteditable = false
7093 gjtab.value = wid
7094 gjtab.id = wid + '_Tab'
7095 gjtab.setAttribute('class','GJTab')
7096 gjtab.setAttribute('spellcheck','false')
7097 gjwin.appendChild(gjtab)
7098
7099 gjstat = GJLog_Stat = document.createElement('textarea')
7100 gjstat.addEventListener('keydown',GJWin_OnKeyDown)
7101 gjstat.id = wid + '_Stat'
7102 gjstat.value = DateShort()
7103 gjstat.setAttribute('class','GJStat')
7104 gjstat.setAttribute('spellcheck','false')
7105 gjwin.appendChild(gjstat)
7106
7107 gjicon = document.createElement('span')
7108 gjicon.addEventListener('keydown',GJWin_OnKeyDown)
7109 gjicon.id = wid + '_Icon'
7110 gjicon.innerHTML = '<G<font color="#F4">J</font>'
7111 gjicon.setAttribute('class','GJIcon')
7112 gjicon.setAttribute('spellcheck','false')
7113 gjwin.appendChild(gjicon)
7114
7115 gjtext = GJLog_Text = document.createElement('textarea')
7116 gjtext.addEventListener('keydown',GJWin_OnKeyDown)
7117 gjtext.addEventListener('keyup',GJWin_OnKeyUp)
7118 gjtext.addEventListener('resize',GJWin_OnResizeTextarea)
7119 gjtext.id = wid + '_Text'
7120 gjtext.setAttribute('class','GJText')
7121 gjtext.setAttribute('spellcheck','false')
7122 gjwin.appendChild(gjtext)
7123
7124
7125 // user's mode as of IME
7126 gjmode = GJWin_Mode = document.createElement('textarea')
7127 gjmode.addEventListener('keydown',GJWin_OnKeyDown)
7128 gjmode.addEventListener('keydown',GJWin_OnKeyDown)
7129 gjmode.id = wid + '_Mode'
7130 gjmode.setAttribute('class','GJMode')
7131 gjmode.setAttribute('spellcheck','false')
7132 gjmode.innerHTML = '[---]'
7133 gjwin.appendChild(gjmode)
7134
7135 gjwin.zIndex = 30000
7136 GJFactory.appendChild(gjwin)
7137
7138 gjtab.scrollTop = 0
7139 gjstat.scrollTop = 0
7140
7141 //x = gjwin.getBoundingClientRect().left.toFixed(0)
7142 //y = gjwin.getBoundingClientRect().top.toFixed(0)
7143 //gjwin.style.position = 'static'
7144 //gjwin.style.left = 0
7145 //gjwin.style.top = 0
7146
7147 //update = '{+wid+'.value=DateShort()}',
7148 update = '{GJ showTime('+wid+')}',
7149 // 2020-09-19 this causes memory leaks
7150 //FProductInterval = window.setInterval(update,200)
7151 //FProductInterval = window.setInterval(GJWin_StatUpdate,200)
7152 //FProductInterval = window.setInterval(GJ_showTime,200,wid);
7153 FProductInterval = window.setInterval(GJ_showTime,200,gjstat);
7154 return update
7155 }
7156 function xxxGJF_StripClass(){
7157   GJLog_Win.style.removeProperty('width')
7158   GJLog_Tab.style.removeProperty('width')
7159   GJLog_Stat.style.removeProperty('width')
7160   GJLog_Text.style.removeProperty('width')
7161   return "Stripped classes"
7162 }
7163 function isElem(id){
7164   return document.getElementById(id) != null
7165 }
7166 function GJLog_append(...args){
7167   txt = GJLog_Text;
7168   if( txt == null ){
7169     return; // maybe GJLog element is removed
7170   }
7171   logs = args.join(' ')
7172   txt.value += logs + '\n'
7173   txt.scrollTop = txt.scrollHeight
7174   //GJLog_Stat.value = DateShort()
7175 }
7176 //window.addEventListener('time',GJLog_StatUpdate)
7177 function test_GJ_Console(){
7178   window.setInterval(GJLog_StatUpdate,1000);
7179   GJ_NewConsole('GJ_Console')
7180   e = GJFactory;
7181   console.log('GJF0 #'+e.id+' from w='+e.style.width+', h='+e.style.height)
7182   e.style.width = 360; //GJFsize
7183   e.style.height = 320;
7184   console.log('GJF0 #'+e.id+' to w='+e.style.width+', h='+e.style.height)
7185 }
7186 // test_GJ_Console();
7187
7188 var StopConsoleLog = true
7189 // 2020-09-15 added,
7190 // log should be saved to permanet memory
7191 // const px = new Proxy(console.log,{ alert() })
7192 __console_log = console.log

```

```

7193 __console_info = console.info
7194 __console_warn = console.warn
7195 __console_error = console.error
7196 __console_exception = console.exception
7197 // should pop callstack info.
7198 console.exception = function(...args){
7199     __console_exception(...args)
7200     alert('-- got console.exception("+args+")')
7201 }
7202 console.error = function(...args){
7203     __console_error(...args)
7204     alert('-- got console.error("+args+")')
7205 }
7206 console.warn = function(...args){
7207     __console_warn(...args)
7208     alert('-- got console.warn("+args+")')
7209 }
7210 console.info = function(...args){
7211     alert('-- got console.info("+args+")')
7212     __console_info(...args)
7213 }
7214 console.xxxlog = function(...args){ // rewrite xxxlog to log to enable it
7215     console_log(...args)
7216     if( StopConsoleLog ){
7217         return;
7218     }
7219     if( 0 <= args[0].indexOf('!') ){
7220         //alert('-- got console.log("+args+")')
7221     }
7222     GJLog_append(...args)
7223 }
7224
7225 //document.getElementById('GshFaviconURL').href = GShellFavicon
7226 //document.getElementById('GshFaviconURL').href = GShellInsideIcon
7227 //document.getElementById('GshFaviconURL').href = ITSmoreQR
7228 //document.getElementById('GshFaviconURL').href = GShellLogo
7229
7230 // id of GShell HTML elements
7231 var E_BANNER = "GshBanner" // banner element in HTML
7232 var E_FOOTER = "GshFooter" // footer element in HTML
7233 var E_GINDEX = "gsh-gindex" // index of Golang code of GShell
7234 var E_GOCODE = "gsh-gocode" // Golang code of GShell
7235 var E_TODO = "gsh-todo" // TODO of GShell
7236 var E_DICT = "gsh-dict" // Dictionary of GShell
7237
7238 function bannerElem(){ return document.getElementById(E_BANNER); }
7239 function bannerStyleFunc(){ return bannerElem().style; }
7240 var bannerStyle = bannerStyleFunc()
7241 function GshSetImages(){
7242     document.getElementById('GshFaviconURL').href = GShellInsideIcon
7243     bannerStyle.backgroundImage = "url("+GShellLogo+")";
7244     //bannerStyle.backgroundImage = "url("+GShellInsideIcon+")";
7245     //bannerStyle.backgroundImage = "url("+GShellFavicon+")";
7246     GMenu.style.backgroundImage = "url("+GShellInsideIcon+")";
7247     showFooter();
7248 }
7249
7250 function footerElem(){ return document.getElementById(E_FOOTER); }
7251 function footerStyle(){ return footerElem().style; }
7252 //footerElem().style.backgroundImage="url("+ITSmoreQR+")";
7253 //footerStyle().backgroundImage = "url("+ITSmoreQR+")";
7254
7255 function html_fold(e){
7256     if( e.innerHTML == "Fold" ){
7257         e.innerHTML = "Unfold"
7258         document.getElementById('gsh-menu-exit').innerHTML=""
7259         document.getElementById('GshStatement').open=false
7260         GshFeatures.open = false
7261         document.getElementById('html-src').open=false
7262         document.getElementById(E_GINDEX).open=false
7263         document.getElementById(E_GOCODE).open=false
7264         document.getElementById(E_TODO).open=false
7265         document.getElementById('References').open=false
7266     }else{
7267         e.innerHTML = "Fold"
7268         document.getElementById('GshStatement').open=true
7269         GshFeatures.open = true
7270         document.getElementById(E_GINDEX).open=true
7271         document.getElementById(E_GOCODE).open=true
7272         document.getElementById(E_TODO).open=true
7273         document.getElementById('References').open=true
7274     }
7275 }
7276 function html_pure(e){
7277     if( e.innerHTML == "Pure" ){
7278         document.getElementById('gsh').style.display=true
7279         //document.style.display = false
7280         e.innerHTML = "Unpure"
7281     }else{
7282         document.getElementById('gsh').style.display=false
7283         //document.style.display = true
7284         e.innerHTML = "Pure"
7285     }
7286 }
7287
7288 var bannerIsStopping = false
7289 //NOTE: .com/JSREF/prop_style_backgroundposition.asp
7290 function shiftBG(){
7291     bannerIsStopping = !bannerIsStopping
7292     bannerStyle.backgroundPosition = "0 0";
7293 }
7294 // status should be inherited on Window Fork(), so use the status in DOM
7295 function html_stop(e,toggle){
7296     if( toggle ){
7297         if( e.innerHTML == "Stop" ){
7298             bannerIsStopping = true
7299             e.innerHTML = "Start"
7300         }else{
7301             bannerIsStopping = false
7302             e.innerHTML = "Stop"
7303         }
7304     }else{
7305         // update JavaScript variable from DOM status
7306         if( e.innerHTML == "Stop" ){ // shown if it's running
7307             bannerIsStopping = false
7308         }else{
7309             bannerIsStopping = true
7310         }
7311     }
7312 }
7313 html_stop(document.getElementById('GshMenuStop'),false) // onInit.
7314 //html_stop(bannerElem(),false) // onInit.
7315
7316 //https://www.w3schools.com/jsref/met_win_setinterval.asp

```

```

7317 var banNshift = 0;
7318 function conslog(str){
7319     //console.log(str);
7320 }
7321 function shiftBanner(){
7322     var now = new Date().getTime();
7323     bpos = ((now/10%10000).toFixed(0)+'px' + " 0px";
7324     if( !bannerIsStopping ){
7325         bannerStyle.backgroundColor = bpos;
7326         //GshBanner.style.setProperty('background-position',bpos,'!important');
7327         banNshift += 1;
7328         conslog('shiftBanner <'+GshBanner.nodeName> '+banNshift
7329             + ' now'+(now%10)
7330             //+ ' stop'+bannerIsStopping
7331             + ' pos'+bpos
7332             + ' -> '+bannerStyle.backgroundColor);
7333     }
7334 }
7335 function Banner_init(){
7336     console.log('-- Banner Shift init. ');
7337     window.setInterval(shiftBanner,10); // onInit.
7338 }
7339
7340 // <a href="https://developer.mozilla.org/ja/docs/Web/API/Window/open">window.open()</a>
7341 // from embedded html to standalone page
7342 var MyChildren = 0
7343 function html_fork(){
7344     ResetPerfMon();
7345     ResetAffView();
7346     Reset_ShadingCanvas();
7347     GJFactory_Destroy();
7348     MyChildren += 1
7349     WinId = document.getElementById('gsh-WinId').innerHTML + "." + MyChildren;
7350     newwin = window.open("",WinId,"");
7351     src = document.getElementById('gsh');
7352     srctml = src.outerHTML
7353     newwin.document.write("/<<"+"html>\n");
7354     newwin.document.write(srctml);
7355     newwin.document.write("<"+"/html>\n");
7356     newwin.document.getElementById('gsh-menu-exit').innerHTML = "Close";
7357     newwin.document.getElementById('gsh-WinId').innerHTML = WinId;
7358     newwin.document.close();
7359     newwin.focus();
7360 }
7361 function html_close(){
7362     window.close()
7363 }
7364 function win_jump(win){
7365     //win = window.top;
7366     win = window.opener; // https://developer.mozilla.org/ja/docs/Web/API/window.opener
7367     if( win == null ){
7368         console.log("jump to window.opener("+win+") (Error)\n");
7369     }else{
7370         console.log("jump to window.opener("+win+")\n");
7371         win.focus();
7372     }
7373 }
7374
7375 // 0.2.9 2020-0902 created chekcsom of HTML
7376 CRC32UNIX = 0x04C11DB7 // Unix cksum
7377 function byteCRC32add(bigcrc,octstr,octlen){
7378     var crc = new Int32Array(1)
7379     crc[0] = bigcrc
7380
7381     let oi = 0
7382     for( ; oi < octlen; oi++){
7383         var oct = new Int8Array(1)
7384         oct[0] = octstr[oi]
7385         for( bi = 0; bi < 8; bi++){
7386             //console.log("--CRC32 "+crc[0]+" "+oct[0].toString(16)+" ["+oi+"."+bi+"]\n")
7387             ovf1 = crc[0] < 0 ? 1 : 0
7388             ovf2 = oct[0] < 0 ? 1 : 0
7389             ovf = ovf1 ^ ovf2
7390             oct[0] <<= 1
7391             crc[0] <<= 1
7392             if( ovf ){ crc[0] ^= CRC32UNIX }
7393         }
7394     }
7395     //console.log("--CRC32 byteAdd return crc="+crc[0]+" "+oi+"/"+"octlen+"\n")
7396     return crc[0];
7397 }
7398 function strCRC32add(bigcrc,stri,strlen){
7399     var crc = new Uint32Array(1)
7400     crc[0] = bigcrc
7401     var code = new Uint8Array(strlen);
7402     for( i = 0; i < strlen; i++){
7403         code[i] = stri.charCodeAt(i) // not charAt() !!!!
7404         //console.log("=== "+code[i].toString(16)+" <<= "+stri[i]+"")
7405     }
7406     crc[0] = byteCRC32add(crc,code,strlen)
7407     //console.log("--CRC32 strAdd return crc="+crc[0]+"")
7408     return crc[0]
7409 }
7410 function byteCRC32end(bigcrc,len){
7411     var crc = new Uint32Array(1)
7412     crc[0] = bigcrc
7413     var slen = new Uint8Array(4)
7414     let li = 0
7415     for( ; li < 4; ){
7416         slen[li] = len
7417         li += 1
7418         len >>= 8
7419         if( len == 0 ){
7420             break
7421         }
7422     }
7423     crc[0] = byteCRC32add(crc[0],slen,li)
7424     crc[0] ^= 0xFFFFFFFF
7425     return crc[0]
7426 }
7427 function strCRC32(stri,len){
7428     var crc = new Uint32Array(1)
7429     crc[0] = 0
7430     crc[0] = strCRC32add(0,stri,len)
7431     crc[0] = byteCRC32end(crc[0],len)
7432     //console.log("--CRC32 "+crc[0]+" "+len+"\n")
7433     return crc[0]
7434 }
7435
7436 DestroyGJLink = null; // to be replaced
7437 DestroyFooter = null; // to be defined
7438 DestroyEventSharingCodeview = function dummy({})
7439 Destroy_WirtualDesktop = function(){}
7440 DestroyNaviButtons = function(){}
```

```

7441
7442 function getSourceText(){
7443     if( DestroyFooter != null ) DestroyFooter();
7444     version = document.getElementById('GshVersion').innerHTML
7445     sfavico = document.getElementById('GshFaviconURL').href;
7446     sbanner = document.getElementById('GshBanner').style.backgroundImage;
7447     spositi = document.getElementById('GshBanner').style.backgroundPosition;
7448
7449     if( document.getElementById('GJC_1') != null ){ GJC_1.remove() }
7450     if( DestroyGJLink != null ) DestroyGJLink();
7451     DestroyEventSharingCodeview();
7452     DestroyWirtualDesktop();
7453     GshTopbar.innerHTML = "";
7454     DestroyIndexBar();
7455     DestroyNaviButtons();
7456     ResetPerfMon();
7457     ResetAffView();
7458     Reset_ShadingCanvas();
7459
7460     // these should be removed by CSS selector or class, after seveda to non-printed attribute
7461     GshBanner.removeAttribute("style");
7462     document.getElementById('GshMenuSign').removeAttribute("style");
7463     styleGMenu = GMenu.getAttribute("style")
7464     GMenu.removeAttribute("style");
7465     styleGStat = GStat.getAttribute("style")
7466     GStat.removeAttribute("style");
7467     styleGTop = GTop.getAttribute("style")
7468     GTop.removeAttribute("style");
7469     styleGshGrid = GshGrid.getAttribute("style")
7470     GshGrid.removeAttribute("style");
7471     //styleGPos = GPos.getAttribute("style");
7472     //GPos.removeAttribute("style");
7473     //GPos.innerHTML = "";
7474     //styleGLog = GLog.getAttribute("style");
7475     //GLog.removeAttribute("style");
7476     //GLog.innerHTML = "";
7477     styleGShellPlane = GShellPlane.getAttribute("style")
7478     GShellPlane.removeAttribute("style")
7479     styleRawTextViewer = RawTextViewer.getAttribute("style")
7480     RawTextViewer.removeAttribute("style")
7481     styleRawTextViewerClose = RawTextViewerClose.getAttribute("style")
7482     RawTextViewerClose.removeAttribute("style")
7483
7484     GshFaviconURL.href = "";
7485     if( iselem('ConfigIcon') ) ConfigIcon.src = "";
7486
7487     //it seems that interHTML and outerHTML generate style="" for these (??)
7488     //GshBanner.removeAttribute('style');
7489     //GshFooter.removeAttribute('style');
7490     //GshMenuSign.removeAttribute('style');
7491     GshBanner.style=""
7492     GshMenuSign.style=""
7493
7494     textarea = document.createElement("textarea")
7495     srchtml = document.getElementById("gsh").outerHTML;
7496     //textarea = document.createElement("textarea")
7497     // 2020-0910 ?? ... this causes inserting style="" to Banner and Footer,
7498     // with Chromium?/ after reloading from file:///
7499     textarea.innerHTML = srchtml
7500     // <a href="https://stackoverflow.com/questions/5796718/html-entity-decode">Thanks</a>
7501     var rawtext = textarea.value
7502     //textarea.destroy()
7503     //rawtext = gsh.textContent // this removes #include <FILENAME> too
7504     var orgtext = ""
7505     + /*<+"html>\n" // lost preamble text
7506     + rawtext
7507     + "<+"html>\n" // lost trail text
7508     ;
7509
7510     tlen = orgtext.length
7511     //console.log("getSourceText: length="+tlen+"\n")
7512     document.getElementById('GshFaviconURL').href = sfavico;
7513
7514     document.getElementById('GshBanner').style.backgroundImage = sbanner;
7515     document.getElementById('GshBanner').style.backgroundPosition = spositi;
7516
7517     GStat.setAttribute("style",styleGStat)
7518     GMenu.setAttribute("style",styleGMenu)
7519     GTop.setAttribute("style",styleGTop)
7520     //GLog.setAttribute("style",styleGLog)
7521     //GPos.setAttribute("style",styleGPos)
7522     GshGrid.setAttribute("style",styleGshGrid)
7523     GShellPlane.setAttribute("style",styleGShellPlane)
7524     RawTextViewer.setAttribute("style",styleRawTextViewer)
7525     RawTextViewerClose.setAttribute("style",styleRawTextViewerClose)
7526     canontext = orgtext.replace(' style=""','')
7527     // open="" too
7528     return canontext
7529 }
7530 function getDigest(){
7531     var text = ""
7532     text = getSourceText()
7533     var digest = ""
7534     tlen = text.length
7535     digest = strCRC32(text,tlen) + " " + tlen
7536     return { text, digest }
7537 }
7538 function html_digest(){
7539     version = document.getElementById('GshVersion').innerHTML
7540     let {text, digest} = getDigest()
7541     alert("cksum: " + digest + " " + version)
7542 }
7543 function charsin(stri,char){
7544     ln = 0;
7545     for( i = 0; i < stri.length; i++){
7546         if( stri.charCodeAt(i) == char.charCodeAt(0) )
7547             ln++;
7548     }
7549     return ln;
7550 }
7551
7552 //class digestElement extends HTMLElement { }
7553 //< script>customElements.define('digest',digestElement)< /script>
7554 function showDigest(e){
7555     result = 'version=' + GshVersion.innerHTML + '\n'
7556     result += 'lines=' + e.dataset.lines + '\n'
7557     + 'length=' + e.dataset.length + '\n'
7558     + 'crc32u=' + e.dataset.crc32u + '\n'
7559     + 'time=' + e.dataset.time + '\n';
7560
7561     alert(result)
7562 }
7563
7564 function html_sign(e){

```

```

7565     if( RawTextViewer.style.zIndex == 1000 ){
7566         hideRawTextViewer()
7567         return
7568     }
7569     GshTopbar.innerHTML = "";
7570     ResetPerfMon();
7571     ResetAffView();
7572     Reset_ShadingCanvas();
7573     DestroyIndexBar();
7574     DestroyNaviButtons();
7575     DestroyEventSharingCodeview();
7576     Destroy_WirtualDesktop();
7577     GJFactory_Destroy()
7578     if( DestroyGJLink != null ) DestroyGJLink();
7579     //gsh_digest_.innerHTML = "";
7580     text = getSourceText() // the original text
7581     tlen = text.length
7582     digest = strCRC32(text,tlen)
7583     //gsh_digest_.innerHTML = digest + " " + tlen
7584     //text = getSourceText() // the text with its digest
7585     Lines = charsin(text, '\n')
7586
7587     name = "gsh"
7588     sid = name + "-digest"
7589     d = new Date()
7590     signedAt = d.getTime()
7591
7592     sign = '/'+'*'+ '<span>'
7593         + ' id="' + sid + '"\n'
7594         + ' class=" digest_"\n'
7595         + ' data-target-id="' + name + '"\n'
7596         + ' data-crc32u="' + digest + '"\n'
7597         + ' data-length="' + tlen + '"\n'
7598         + ' data-lines="' + Lines + '"\n'
7599         + ' data-time="' + signedAt + '"\n'
7600         + '>' + '</span>\n*'+ '\n'
7601
7602     text = sign + text
7603
7604     txthtml = '<' + '<table id="LineNumbered"><' + '<tr><' + '<td>'
7605         + '<' + '<textarea cols=5 rows=' + Lines + ' class="LineNumber">'
7606     for( i = 1; i <= Lines; i++ ){
7607         txthtml += i.toString() + '\n'
7608     }
7609     txthtml += ""
7610         + '<' + '</textarea>'
7611         + '<' + '</td><' + '<td>'
7612         + '<' + '<textarea cols=150 rows=' + Lines + ' spellcheck="false"'
7613         + ' class="LineNumber">'
7614         + text + '<' + '</textarea>'
7615         + '<' + '</td><' + '</tr><' + '</table>'
7616
7617     for( i = 1; i <= 30; i++ ){
7618         txthtml += '<br>\n'
7619     }
7620     RawTextViewer.innerHTML = txthtml
7621     RawTextViewer.spellcheck = false // (spellcheck above seems ineffective)
7622
7623     btn = e
7624     e.style.color = "rgba(128,128,255,0.9)";
7625     y = e.getBoudingClientRect().top.toFixed(0)
7626     //h = e.getBoudingClientRect().height.toFixed(0)
7627     RawTextViewer.style.top = Number(y) + 30
7628     RawTextViewer.style.left = 100;
7629     RawTextViewer.style.height = window.innerHeight - 20;
7630     //RawTextViewer.style.Opacity = 1.0;
7631     //RawTextViewer.style.backgroundColor = "rgba(0,0,0,0.0)";
7632     RawTextViewer.style.backgroundColor = "rgba(255,255,255,0.8)";
7633     RawTextViewer.style.zIndex = 1000;
7634     RawTextViewer.style.display = true;
7635
7636     if( RawTextViewerClose.style == null ){
7637         RawTextViewerClose.style = "";
7638     }
7639     RawTextViewerClose.style.top = Number(y) + 10
7640     RawTextViewerClose.style.left = 100;
7641     RawTextViewerClose.style.zIndex = 1001;
7642
7643     ScrollToElement(CurElement,RawTextViewerClose)
7644 }
7645 function hideRawTextViewer(){
7646     RawTextViewer.style.left = 10000;
7647     RawTextViewer.style.zIndex = -100;
7648     RawTextViewer.style.Opacity = 0.0;
7649     RawTextViewer.style = null
7650     RawTextViewer.innerHTML = "";
7651
7652     GshMenuSign.style.color = "rgba(255,128,128,1.0)";
7653     RawTextViewerClose.style.top = 0;
7654     RawTextViewerClose.style = null
7655 }
7656
7657 // source code viewr
7658 function frame_close(){
7659     srcframe = document.getElementById("src-frame");
7660     srcframe.innterHTML = "";
7661     //srcframe.style.cols = 1;
7662     srcframe.style.rows = 1;
7663     srcframe.style.height = 0;
7664     srcframe.style.display = false;
7665     src = document.getElementById("SrcTextarea");
7666     src.innerHTML = ""
7667     //src.cols = 0
7668     src.rows = 0
7669     src.display = false
7670     //alert("--closed--")
7671 }
7672 //<!-- | <span onclick="html_view();">Source</span> -->
7673 //<!-- | <span onclick="frame_close();">SourceClose</span> -->
7674 //<!--| <span>Download</span> -->
7675 function frame_open(){
7676     GshTopbar.innerHTML = "";
7677     ResetPerfMon();
7678     ResetAffView();
7679     Reset_ShadingCanvas();
7680     DestroyIndexBar();
7681     DestroyNaviButtons();
7682     if( DestroyFooter != null ) DestroyFooter();
7683     document.getElementById('GshFaviconURL').href = "";
7684     oldsrc = document.getElementById("GENSRC");
7685     if( oldsrc != null ){
7686         //alert("--I--(erasing old text)")
7687         oldsrc.innterHTML = "";
7688         return

```

```

7689     }else{
7690         //alert("--I--(no old text)")
7691     }
7692     styleBanner = GshBanner.getAttribute("style")
7693     GshBanner.removeAttribute("style")
7694     if( document.getElementById('GJC_1') ){ GJC_1.remove() }
7695
7696     GshFaviconURL.href = "";
7697     if( iselem('ConfigIcon') ) ConfigIcon.src = "";
7698     GStat.removeAttribute('style')
7699     GshGrid.removeAttribute('style')
7700     GshMenuSign.removeAttribute('style')
7701     //GPos.removeAttribute('style')
7702     //GPos.innerHTML = "";
7703     //GLog.removeAttribute('style')
7704     //GLog.innerHTML = "";
7705     GMenu.removeAttribute('style')
7706     GTop.removeAttribute('style')
7707     GShellPlane.removeAttribute('style')
7708     RawTextViewer.removeAttribute('style')
7709     RawTextViewerClose.removeAttribute('style')
7710
7711     if( DestroyGJLink != null ) DestroyGJLink();
7712     GJFactory_Destroy()
7713     Destroy_WirtualDesktop();
7714     DestroyEventSharingCodeview();
7715
7716     src = document.getElementById("gsh");
7717     srchtml = src.outerHTML
7718     srcframe = document.getElementById("src-frame");
7719     srcframe.innerHTML = ""
7720     + "<"+<cite id="GENSRC">\n"
7721     + "<"+<style>\n"
7722     + "#GENSRC textarea{tab-size:4;}\n"
7723     + "#GENSRC textarea{-o-tab-size:4;}\n"
7724     + "#GENSRC textarea{-moz-tab-size:4;}\n"
7725     + "#GENSRC textarea{spellcheck:false;}\n"
7726     + "</"+<style>\n"
7727     + "<"+<textarea id="SrcTextarea" cols=100 rows=20 class="gsh-code" spellcheck="false">"
7728     + "/<"+<html>\n" // lost preamble text
7729     + srchtml
7730     + "<"+</html>\n" // lost trail text
7731     + "</"+<textarea>\n"
7732     + "</"+<cite><!-- GENSRC -->\n";
7733
7734     //srcframe.style.cols = 80;
7735     //srcframe.style.rows = 80;
7736
7737     GshBanner.setAttribute('style',styleBanner)
7738 }
7739 function fill_CSSView(){
7740     part = document.getElementById('GshStyleDef')
7741     view = document.getElementById('gsh-style-view')
7742     view.innerHTML = ""
7743     + "<"+<textarea cols=100 rows=20 class="gsh-code">"
7744     + part.innerHTML
7745     + "<"+</textarea>"
7746 }
7747 function fill_JavaScriptView(){
7748     jspart = document.getElementById('gsh-script')
7749     view = document.getElementById('gsh-script-view')
7750     view.innerHTML = ""
7751     + "<"+<textarea cols=100 rows=20 class="gsh-code">"
7752     + jspart.innerHTML
7753     + "<"+</textarea>"
7754 }
7755 function fill_DataView(){
7756     part = document.getElementById('gsh-data')
7757     view = document.getElementById('gsh-data-view')
7758     view.innerHTML = ""
7759     + "<"+<textarea cols=100 rows=20 class="gsh-code">"
7760     + part.innerHTML
7761     + "<"+</textarea>"
7762 }
7763 function jumpto_StyleView(){
7764     jsview = document.getElementById('html-src')
7765     jsview.open = true
7766     jsview = document.getElementById('gsh-style-frame')
7767     jsview.open = true
7768     fill_CSSView()
7769 }
7770 function jumpto_JavaScriptView(){
7771     jsview = document.getElementById('html-src')
7772     jsview.open = true
7773     jsview = document.getElementById('gsh-script-frame')
7774     jsview.open = true
7775     fill_JavaScriptView()
7776 }
7777 function jumpto_DataView(){
7778     jsview = document.getElementById('html-src')
7779     jsview.open = true
7780     jsview = document.getElementById('gsh-data-frame')
7781     jsview.open = true
7782     fill_DataView()
7783 }
7784 function jumpto_WholeView(){
7785     jsview = document.getElementById('html-src')
7786     jsview.open = true
7787     jsview = document.getElementById('gsh-whole-view')
7788     jsview.open = true
7789     frame_open()
7790 }
7791 function html_view(){
7792     html_stop();
7793
7794     banner = document.getElementById('GshBanner').style.backgroundImage;
7795     footer = document.getElementById('GshFooter').style.backgroundImage;
7796     document.getElementById('GshBanner').style.backgroundImage = "";
7797     document.getElementById('GshBanner').style.backgroundPosition = "";
7798     document.getElementById('GshFooter').style.backgroundImage = "";
7799
7800     //srcwin = window.open("", "CodeView2", "");
7801     srcwin = window.open("", "", "");
7802     srcwin.document.write("<span id="gsh">\n");
7803
7804     src = document.getElementById("gsh");
7805     srcwin.document.write("<"+<style>\n");
7806     srcwin.document.write("<textarea{tab-size:4;}\n");
7807     srcwin.document.write("<textarea{-o-tab-size:4;}\n");
7808     srcwin.document.write("<textarea{-moz-tab-size:4;}\n");
7809     srcwin.document.write("</style>\n");
7810     srcwin.document.write("<h2>\n");
7811     srcwin.document.write("<"+<span onclick="window.close();\>Close</span> | \n");
7812     //srcwin.document.write("<"+<span onclick="html_stop();\>Run</span>\n");

```

```

7813 srcwin.document.write("</h2>\n");
7814 srcwin.document.write("<+>textarea id=\"gsh-src-src\" cols=100 rows=60>");
7815 srcwin.document.write("/<+>html>\n");
7816 srcwin.document.write("<+>span id=\"gsh\">");
7817 srcwin.document.write(src.innerHTML);
7818 srcwin.document.write("<+>/span><+>/html>\n");
7819 srcwin.document.write("</+>textarea>\n");
7820
7821 document.getElementById('GshBanner').style.backgroundImage = banner;
7822 document.getElementById('GshFooter').style.backgroundImage = footer
7823
7824 sty = document.getElementById("GshStyleDef");
7825 srcwin.document.write("<+>style>\n");
7826 srcwin.document.write(sty.innerHTML);
7827 srcwin.document.write("<+>/style>\n");
7828
7829 run = document.getElementById("gsh-script");
7830 srcwin.document.write("<+>script>\n");
7831 srcwin.document.write(run.innerHTML);
7832 srcwin.document.write("<+>/script>\n");
7833
7834 srcwin.document.write("<+>/span><+>/html>\n"); // gsh span
7835 srcwin.document.close();
7836 srcwin.focus();
7837 }
7838 GSH = document.getElementById("gsh")
7839
7840 //GSH.onclick = "alert('Ouch!')"
7841 //GSH.css = "{background-color:#eef;}"
7842 //GSH.style = "background-color:#eef;"
7843 //GSH.style.display = false;
7844 //alert('Ouch0!')
7845 //GSH.style.display = true;
7846
7847 // 2020-0904 created, tentative
7848 //document.addEventListener('keydown',jgshCommand);
7849 //CurElement = GshStatement
7850 CurElement = GshMenu
7851 MemElement = GshMenu
7852
7853 function nextSib(e){
7854     n = e.nextSibling;
7855     for( i = 0; i < 100; i++ ){
7856         if( n == null ){
7857             break;
7858         }
7859         if( n.nodeName == "DETAILS" ){
7860             return n;
7861         }
7862         n = n.nextSibling;
7863     }
7864     return null;
7865 }
7866 function prevSib(e){
7867     n = e.previousSibling;
7868     for( i = 0; i < 100; i++ ){
7869         if( n == null ){
7870             break;
7871         }
7872         if( n.nodeName == "DETAILS" ){
7873             return n;
7874         }
7875         n = n.previousSibling;
7876     }
7877     return null;
7878 }
7879 function setColor(e,eName,eColor){
7880     if( e.hasChildNodes() ){
7881         s = e.childNodes;
7882         if( s != null ){
7883             for( ci = 0; ci < s.length; ci++ ){
7884                 if( s[ci].nodeName == eName ){
7885                     s[ci].style.color = eColor;
7886                     //s[ci].style.backgroundColor = eColor;
7887                     break;
7888                 }
7889             }
7890         }
7891     }
7892 }
7893
7894 // https://docs.microsoft.com/en-us/previous-versions//hh781509(v=vs.85)
7895 function showCurElementPosition(ev){
7896     // if( document.getElementById("GPos") == null ){
7897     //     return;
7898     // }
7899     // if( GPos == null ){
7900     //     return;
7901     // }
7902     e = CurElement
7903     y = e.getBoundingClientRect().top.toFixed(0)
7904     x = e.getBoundingClientRect().left.toFixed(0)
7905
7906     h = ev + " "
7907     h += 'y'+y+", "+ 'x'+x+" -- "
7908     h += "w=" + window.innerWidth + ", h=" + window.innerHeight + " -- "
7909     //GPos.test = h
7910     //GPos.innerHTML = h
7911     // GPos.innerHTML = h
7912 }
7913
7914 function zero2(n){
7915     if( n < 10 ){
7916         return '0' + n;
7917     }else{
7918         return n;
7919     }
7920 }
7921 function DateHourMin(){
7922     d = new Date();
7923     //return '%02d:%02d'.sprintf(d.getHours(),d.getMinutes())
7924     return zero2(d.getHours()) + ":" + zero2(d.getMinutes());
7925 }
7926 function DateShort0(d){
7927     return d.getFullYear()
7928         + '/' + zero2(d.getMonth())
7929         + '/' + zero2(d.getDate())
7930         + ' ' + zero2(d.getHours())
7931         + ':' + zero2(d.getMinutes())
7932         + ':' + zero2(d.getSeconds())
7933 }
7934 function DateShort(){
7935     return DateShort0(new Date());
7936 }

```



```

7937 function DateLong0(ms){
7938     d = new Date();
7939     d.setTime(ms);
7940     return DateShort0(d)
7941         + ' ' + d.getMilliseconds()
7942         + ' ' + d.getTimezoneOffset()/60
7943         + ' ' + d.getTime() + '.' + d.getMilliseconds()
7944 }
7945 function DateLong(){
7946     return DateLong0(new Date());
7947 }
7948 function GShellMenu(e){
7949     //GLog.innerHTML = "Hello, World! (" + DateLong() + ")"
7950     //showGShellPlane()
7951     ConfigClick();
7952 }
7953 // placements of planes
7954 function GShellResizeX(ev){
7955     //if( document.getElementById("GMenu") != null ){
7956         GMenu.style.left = window.innerWidth - 100
7957         GMenu.style.top = window.innerHeight - 90 - 200
7958         //console.log("place GMENU "+GMenu.style.left+" "+GMenu.style.top)
7959     }
7960     //}
7961     GStat.style.width = window.innerWidth
7962     //if( document.getElementById("GPos") != null ){
7963         //GPos.style.width = window.innerWidth
7964         //GPos.style.top = window.innerHeight - 30; //GPos.style.height
7965     //}
7966     //if( document.getElementById("GLog") != null ){
7967         // GLog.style.width = window.innerWidth
7968         //GLog.innerHTML = ""
7969     //}
7970     //if( document.getElementById("GLog") != null ){
7971         //GLog.innerHTML = "Resize: w=" + window.innerWidth +
7972         //", h=" + window.innerHeight
7973     //}
7974     showCurElementPosition(ev)
7975 }
7976 function GShellResize(){
7977     GShellResizeX("[RESIZE]")
7978 }
7979 window.onresize = GShellResize
7980 var prevNode = null
7981 var LogMouseMoveOverElement = false;
7982 function GJSH_OnMouseMove(ev){
7983     if( LogMouseMoveOverElement == false ){
7984         return;
7985     }
7986     x = ev.clientX
7987     y = ev.clientY
7988     d = new Date()
7989     t = d.getTime() / 1000
7990     if( document.elementFromPoint ){
7991         e = document.elementFromPoint(x,y)
7992         if( e != null ){
7993             if( e == prevNode ){
7994                 }else{
7995                     console.log('Mo-'+t+'+'+x+'+'+y+' '
7996                                 +e.nodeType+' '+e.tagName+'#'+e.id)
7997                     prevNode = e
7998                 }
7999             }else{
8000                 console.log(t+'('+x+', '+y+') no element')
8001             }
8002         }else{
8003             console.log(t+'('+x+', '+y+') no elementFromPoint')
8004         }
8005     }
8006     window.addEventListener('mousemove',GJSH_OnMouseMove);
8007 }
8008 function GJSH_OnMouseMoveScreen(ev){
8009     x = ev.screenX
8010     y = ev.screenY
8011     d = new Date()
8012     t = d.getTime() / 1000
8013     console.log(t+'('+x+', '+y+') no elementFromPoint')
8014 }
8015 //screen.addEventListener('mousemove',GJSH_OnMouseMoveScreen);
8016 }
8017 function ScrollToElement(oe,ne){
8018     ne.scrollIntoView()
8019     ny = ne.getBoundingClientRect().top.toFixed(0)
8020     nx = ne.getBoundingClientRect().left.toFixed(0)
8021     //GLog.innerHTML = "["+ny+", "+nx+"]"
8022     //window.scrollTo(0,0)
8023 }
8024 GTop.style.backgroundColor = "rgba(0,0,0,0.0)"
8025 GShGrid.style.left = '250px';
8026 GShGrid.style.zIndex = 0
8027 if( false ){
8028     oy = oe.getBoundingClientRect().top.toFixed(0)
8029     ox = oe.getBoundingClientRect().left.toFixed(0)
8030     y = e.getBoundingClientRect().top.toFixed(0)
8031     x = e.getBoundingClientRect().left.toFixed(0)
8032     window.scrollTo(x,y)
8033     ny = e.getBoundingClientRect().top.toFixed(0)
8034     nx = e.getBoundingClientRect().left.toFixed(0)
8035     //GLog.innerHTML = "["+oy+", "+ox+"]->["+y+", "+x+"]->["+ny+", "+nx+"]"
8036 }
8037 }
8038 function showGShellPlane(){
8039     if( GShellPlane.style.zIndex == 0 ){
8040         GShellPlane.style.zIndex = 1000;
8041         GShellPlane.style.left = 30;
8042         GShellPlane.style.height = 320;
8043         GShellPlane.innerHTML = DateLong() + "<br>" +
8044         "-- History --<br>" + MyHistory;
8045     }else{
8046         GShellPlane.style.zIndex = 0;
8047         GShellPlane.style.left = 0;
8048         GShellPlane.style.height = 50;
8049         GShellPlane.innerHTML = "";
8050     }
8051 }
8052 var SuppressGJShell = false
8053 function jgshCommand(kevent){
8054     if( SuppressGJShell ){
8055         return
8056     }
8057     key = kevent
8058     keycode = key.code
8059     //GStat.style.width = window.innerWidth
8060     GStat.style.backgroundColor = "rgba(0,0,0,0.4)"

```

```

8061
8062 console.log("JSGsh-Key:"+keycode+"(^-^)//")
8063 if( keycode == "Slash"){
8064     console.log('('+x+', '+y+') ')
8065     e = document.elementFromPoint(x,y)
8066     console.log('('+x+', '+y+') '+e.nodeType+ ' '+e.tagName+'#'+e.id)
8067 }else
8068 if( keycode == "Digit0" ){ // fold side-bar
8069     // "Zero page"
8070     showGShellPlane();
8071 }else
8072 if( keycode == "Digit1" ){ // fold side-bar
8073     primary.style.width = "94%"
8074     secondary.style.width = "0%"
8075     secondary.style.opacity = 0
8076     GStat.innerHTML = "[Single Column View]"
8077 }else
8078 if( keycode == "Digit2" ){ // unfold side-bar
8079     primary.style.width = "58%"
8080     secondary.style.width = "36%"
8081     secondary.style.opacity = 1
8082     GStat.innerHTML = "[Double Column View]"
8083 }else
8084 if( keycode == "KeyU" ){ // fold/unfold all
8085     html_fold(GshMenuFold);
8086     location.href = "#"+CurElement.id;
8087 }else
8088 if( keycode == "KeyO" || keycode == "ArrowRight" ){ // fold the element
8089     CurElement.open = !CurElement.open;
8090 }else
8091 if( keycode == "ArrowRight" ){ // unfold the element
8092     CurElement.open = true
8093 }else
8094 if( keycode == "ArrowLeft" ){ // unfold the element
8095     CurElement.open = false
8096 }else
8097 if( keycode == "KeyI" ){ // inspect the element
8098     e = CurElement
8099     //GLog.innerHTML =
8100     GJLog_append("Current Element: " + e + "<br>"
8101         + "name="+e.nodeName + ", "
8102         + "id="+e.id + ", "
8103         + "children="+e.childNodes.length + ", "
8104         + "parent="+e.parentNode.id + "<br>"
8105         + "text="+e.textContent)
8106     GStat.style.backgroundColor = "rgba(0,0,0,0.8)"
8107     return
8108 }else
8109 if( keycode == "KeyM" ){ // memory the position
8110     MemElement = CurElement
8111 }else
8112 if( keycode == "KeyN" || keycode == "ArrowDown" ){ // next element
8113     e = nextSib(CurElement)
8114     if( e != null ){
8115         setColor(CurElement,"SUMMARY","#fff")
8116         setColor(e,"SUMMARY","#8f8") // should be complement ?
8117         oe = CurElement
8118         CurElement = e
8119         //location.href = "#"+e.id;
8120         ScrollToElement(oe,e)
8121     }
8122 }else
8123 if( keycode == "KeyP" || keycode == "ArrowUp" ){ // previous element
8124     oe = CurElement
8125     e = prevSib(CurElement)
8126     if( e != null ){
8127         setColor(CurElement,"SUMMARY","#fff")
8128         setColor(e,"SUMMARY","#8f8") // should be complement ?
8129         CurElement = e
8130         //location.href = "#"+e.id;
8131         ScrollToElement(oe,e)
8132     }else{
8133         e = document.getElementById("GshBanner")
8134         if( e != null ){
8135             setColor(CurElement,"SUMMARY","#fff")
8136             CurElement = e
8137             ScrollToElement(oe,e)
8138         }else{
8139             e = document.getElementById("primary")
8140             if( e != null ){
8141                 setColor(CurElement,"SUMMARY","#fff")
8142                 CurElement = e
8143                 ScrollToElement(oe,e)
8144             }
8145         }
8146     }
8147 }else
8148 if( keycode == "KeyR" ){
8149     location.reload()
8150 }else
8151 if( keycode == "KeyJ" ){
8152     GshGrid.style.top = '120px';
8153     GshGrid.innerHTML = '(>_<){Down}';
8154 }else
8155 if( keycode == "KeyK" ){
8156     GshGrid.style.top = '0px';
8157     GshGrid.innerHTML = '^_^){Up}';
8158 }else
8159 if( keycode == "KeyH" ){
8160     GshGrid.style.left = '0px';
8161     GshGrid.innerHTML = "( '_ ) {Left}";
8162 }else
8163 if( keycode == "KeyL" ){
8164     //GLog.innerHTML +=
8165     GJLog_append(
8166         "screen="+screen.width+'px'+<br>'+
8167         "window="+window.innerWidth+'px'+<br>'+
8168     )
8169     GshGrid.style.left = (document.documentElement.clientWidth-160).toString(10)+'px';
8170     GshGrid.innerHTML = '{@_@}{Right}';
8171 }else
8172 if( keycode == "KeyS" ){
8173     html_stop(GshMenuStop,true)
8174 }else
8175 if( keycode == "KeyF" ){
8176     html_fork()
8177 }else
8178 if( keycode == "KeyC" ){
8179     window.close()
8180 }else
8181 if( keycode == "KeyD" ){
8182     html_digest()
8183 }else
8184 if( keycode == "KeyV" ){

```

```

8185     e = document.getElementById('gsh-digest')
8186     if( e != null ){
8187         showDigest(e)
8188     }
8189 }
8190
8191 showCurElementPosition("[+key.code+] --");
8192 //if( document.getElementById("GPos") != null ){
8193 //GPos.innerHTML += "[+key.code+] --"
8194 //}
8195 //GShellResizeX("[+key.code+] --");
8196 }
8197 var initGSKC = false;
8198 function GShell_initKeyCommands(){
8199     if( initGSKC ){ return; } initGSKC = true;
8200
8201     GShellResizeX("[INIT]");
8202     DisplaySize = '-- Display: '
8203     + 'screen='+screen.width+'px, '+window.innerWidth+'px';
8204
8205     let {text, digest} = getDigest()
8206     //GLog.innerHTML +=
8207     GJLog.append(
8208         '-- GShell: ' + GshVersion.innerHTML + '\n' +
8209         '-- Digest: ' + digest + '\n' +
8210         DisplaySize
8211         //+ "<br>" + "-- LastVisit:<br>" + MyHistory
8212     )
8213     GShellResizeX(null);
8214 }
8215 //GShell_initKeyCommands();
8216
8217 // <a href="https://www.w3.org/TR/WebCryptoAPI/">Web Cryptography API</a>
8218 //Convert a string into an ArrayBuffer
8219 //from https://developers.google.com/web/updates/2012/06/How-to-convert-ArrayBuffer-to-and-from-String
8220 function str2ab(str) {
8221     const buf = new ArrayBuffer(str.length);
8222     const bufView = new Uint8Array(buf);
8223     for (let i = 0, strLen = str.length; i < strLen; i++) {
8224         bufView[i] = str.charCodeAt(i);
8225     }
8226     return buf;
8227 }
8228 function importPrivateKey(pem) {
8229     const binaryDerString = window.atob(pemContents);
8230     const binaryDer = str2ab(binaryDerString);
8231     return window.crypto.subtle.importKey(
8232         "pkcs8",
8233         binaryDer,
8234         {
8235             name: "RSA-PSS",
8236             modulusLength: 2048,
8237             publicExponent: new Uint8Array([1, 0, 1]),
8238             hash: "SHA-256",
8239         },
8240         true,
8241         ["sign"]
8242     );
8243 }
8244 //importPrivateKey(ppem)
8245
8246 //key = {}
8247 //buf = "abc"
8248 //enc = "xyzxxxxx"; //crypto.publicEncrypt(key,buf)
8249 //b64 = btoa(enc)
8250 //dec = atob(b64)
8251 //GLog.innerHTML = "enc:" + b64 + ", dec:" + dec
8252 </script>
8253 */
8254
8255 /*
8256 <!-- ----- GJConsole BEGIN { ----- -->
8257 <span id="gjc" data-title="GJConsole" data-author="sato@its-more.jp">
8258 <details><summary>GJ Console</summary>
8259 <p>
8260 <span id="GJE_RootNode0"></span>
8261 <span id="GJCL_Container"></span>
8262 </p>
8263 <style id="GJConsoleStyle">
8264 .GJConsole {
8265     z-index:1000;
8266     width:400; height:200px;
8267     margin:2px;
8268     color:#fff; background-color:#66a;
8269     font-size:12px; font-family:monospace,Courier New;
8270 }
8271 </style>
8272 <script id="GJConsoleScript" class="GJConsole">
8273 var PS1 = "$ "
8274 function GJC_KeyDown(keyevent){
8275     key = keyevent.code
8276     if( key == "Enter" ){
8277         GJC_Command(this)
8278         this.value += "\n" + PS1 // prompt
8279     }else
8280     if( key == "Escape"){
8281         SuppressGJShell = false
8282         GshMenu.focus() // should be previous focus
8283     }
8284 }
8285
8286 var GJC_SessionId
8287 function GJC_SetSessionId(){
8288     var xd = new Date()
8289     GJC_SessionId = xd.getTime() / 1000
8290 }
8291 GJC_SetSessionId()
8292 function GJC_Memory(mem,args,text){
8293     argv = args.split(' ')
8294     cmd = argv[0]
8295     argv.shift()
8296     args = argv.join(' ')
8297     ret = ""
8298
8299     if( cmd == 'clear' ){
8300         Permanent.setItem(mem,'')
8301     }else
8302     if( cmd == 'read' ){
8303         ret = Permanent.getItem(mem)
8304     }else
8305     if( cmd == 'save' ){
8306         val = Permanent.getItem(mem)
8307         if( val == null ){ val = "" }
8308         d = new Date()

```

```

8309     val += d.getTime()/1000+ " "+GJC_SessionId+ " "+document.URL+ " "+args+"\n"
8310     val += text.value
8311     Permanent.setItem(mem,val)
8312 }else
8313 if( cmd == 'write' ){
8314     val = Permanent.getItem(mem)
8315     if( val == null ){ val = "" }
8316     d = new Date()
8317     val += d.getTime()/1000+ " "+GJC_SessionId+ " "+document.URL+ " "+args+"\n"
8318     Permanent.setItem(mem,val)
8319 }else{
8320     ret = "Commands: write | read | save | clear"
8321 }
8322 return ret
8323 }
8324 // -- 2020-09-14 added TableEditor
8325 var GJE_CurElement = null; //GJE_RootNode
8326 GJE_NodeSaved = null
8327 GJE_TableNo = 1
8328 function GJE_StyleKeyCommand(kev){
8329     keycode = kev.code
8330     console.log('GJE-Key: '+keycode)
8331     if( keycode == 'Escape' ){
8332         GJE_SetStyle(this);
8333     }
8334     kev.stopPropagation()
8335     // https://developer.mozilla.org/en-US/docs/Web/API/Event/stopPropagation
8336 }
8337 var GJE_CommandMode = false
8338 function GJE_TableKeyCommand(kev,tab){
8339     wasCmdMode = GJE_CommandMode
8340     key = kev.code
8341     if( key == 'Escape' ){
8342         console.log("To command mode: "+tab.nodeName+"#" +tab.id)
8343         //tab.setAttribute('contenteditable','false')
8344         tab.style.caretColor = "blue"
8345         GJE_CommandMode = true
8346     }else
8347     if( key == "KeyA" ){
8348         tab.style.caretColor = "red"
8349         GJE_CommandMode = false
8350     }else
8351     if( key == "KeyI" ){
8352         tab.style.caretColor = "red"
8353         GJE_CommandMode = false
8354     }else
8355     if( key == "KeyO" ){
8356         tab.style.caretColor = "red"
8357         GJE_CommandMode = false
8358     }else
8359     if( key == "KeyJ" ){
8360         console.log("ROW-DOWN")
8361     }else
8362     if( key == "KeyK" ){
8363         console.log("ROW-UP")
8364     }else
8365     if( key == "Keyw" ){
8366         console.log("COL-FORW")
8367     }else
8368     if( key == "Keyb" ){
8369         console.log("COL-BACK")
8370     }
8371     kev.stopPropagation()
8372     if( wasCmdMode ){
8373         kev.preventDefault()
8374     }
8375 }
8376 function GJE_DragEvent(ev,elem){
8377     x = ev.clientX
8378     y = ev.clientY
8379     console.log("Dragged: "+this.nodeName+'#' +this.id+' x='+x+' y='+y)
8380 }
8381 // https://developer.mozilla.org/en-US/docs/Web/API/Event/stopPropagation
8382 // https://www.w3.org/TR/uievents/#events-mouseevents
8383 function GJE_DropEvent(ev,elem){
8384     x = ev.clientX
8385     y = ev.clientY
8386     this.style.x = x
8387     this.style.y = y
8388     this.style.position = 'absolute' // 'fixed'
8389     this.parentNode = gsh // just for test
8390     console.log("Dropped: "+this.nodeName+'#' +this.id+' x='+x+' y='+y
8391     +' parent='+this.parentNode.id)
8392 }
8393 function GJE_SetTableStyle(ev){
8394     this.innerHTML = this.value; // sync. for external representation?
8395     if(false){
8396         stid = this.parentNode.id+this.id
8397         // and remove " span" at the end
8398         e = document.getElementById(stid)
8399         //alert('SetTableStyle #' +e.id+' \n'+this.value)
8400         if( e != null ){
8401             e.innerHTML = this.value
8402         }else{
8403             console.log('Style Not found: '+stid)
8404         }
8405         //alert('event StopPropagation: '+ev)
8406     }
8407 }
8408 function setCSSofClass(cclass,cstyle){
8409     const ss = document.styleSheets[3]; // 0, 1, 2, 3, ... ?
8410     rlen = ss.cssRules.length;
8411     let tabrule = null;
8412     rulex = -1
8413     // should skip white space at the top of cstyle
8414     sel = cstyle.charAt(0);
8415     selector = sel+cclass;
8416     console.log('-- search style rule for '+selector)
8417     for(let i = 0; i < rlen; i++){
8418         cr = ss.cssRules[i];
8419         console.log('CSS rule [' +i+'/' +rlen+' ] '+cr.selectorText);
8420         if( cr.selectorText === selector ){ // css class selector
8421             tabrule = ss.cssRules[i];
8422             console.log('CSS rule found for: [' +i+'/' +rlen+' ] '+selector);
8423             ss.deleteRule(i);
8424             //rlen = ss.cssRules.length;
8425             rulex = i
8426             // should search and replace the property here
8427         }
8428     }
8429 }
8430 // https://developer.mozilla.org/en-US/docs/Web/API/CSSStyleSheet/insertRule

```

```

8433 if( tabrule == null ){
8434     console.log('CSS rule NOT found for:['+rlen+' '+selector);
8435     ss.insertRule(cstyle,rlen);
8436     ss.insertRule(cstyle,0); // override by 0?
8437     console.log('CSS rule inserted:['+(rlen+1)+' '\n'+cstyle);
8438 }else{
8439     ss.insertRule(cstyle,rlen);
8440     ss.insertRule(cstyle,0);
8441     console.log('CSS rule replaced:['+(rlen+1)+' '\n'+cstyle);
8442 }
8443 }
8444 function GJE_SetStyle(te){
8445     console.log('Apply the style to:'+te.id+'\n');
8446     console.log('Apply the style to:'+te.parentNode.id+'\n');
8447     console.log('Apply the style to:'+te.parentNode.class+'\n');
8448     cclass = te.parentNode.class;
8449     setCSSofClass(cclass,te.value); // should get selector part from
8450     // selector { rules }
8451 }
8452 if(false){
8453     //console.log('Apply the style:')
8454     //stid = this.parentNode.id+this.id+
8455     //stid = this.id+".style"
8456     css = te.value
8457     stid = te.parentNode.id+".style"
8458     e = document.getElementById(stid)
8459     if( e != null ){
8460         //console.log('Apply the style:'+e.id+'\n'+te.value);
8461         console.log('Apply the style:'+e.id+'\n'+css);
8462         // e.innerHTML = css; //te.value;
8463         //ncss = e.sheet;
8464         //ncss.insertRule(te.value,ncss.cssRules.length);
8465     }else{
8466         console.log('No element to Apply the style: '+stid)
8467     }
8468     tblid = te.parentNode.id+".table";
8469     e = document.getElementById(tblid);
8470     if( e != null ){
8471         //e.setAttribute('style',css);
8472         e.setProperty('style',css,'!important');
8473     }
8474 }
8475 }
8476 function makeTable(argv){
8477     //tid = ''
8478     //cwe = GJE_CurElement
8479     cwe = GJCL_Container;
8480     //cwd = GJFactory;
8481     tid = 'table_' + GJE_TableNo
8482
8483     nt = new Text('\n')
8484     cwe.appendChild(nt)
8485
8486     ne = document.createElement('span'); // the container
8487     cwe.appendChild(ne)
8488     ne.id = tid + 'span'
8489     ne.setAttribute('contenteditable',true)
8490
8491     htspan = document.createElement('span'); // html part
8492     //htspan.id = tid + '.html'
8493     //ne.innerHTML = '\n'
8494     nt = new Text('\n')
8495     ne.appendChild(nt)
8496     ne.appendChild(htspan)
8497
8498     htspan.id = tid
8499     htspan.setAttribute('class',tid)
8500
8501     ne.setAttribute('draggable','true')
8502     ne.addEventListener('drag',GJE_DragEvent);
8503     ne.addEventListener('dragend',GJE_DropEvent);
8504
8505     var col = 3
8506     var row = 2
8507     if( argv[0] != null ){
8508         col = argv[0]
8509         argv.shift()
8510     }
8511     if( argv[0] != null ){
8512         row = argv[0]
8513         argv.shift()
8514     }
8515 }
8516 //ne.setAttribute('class',tid)
8517 ht = "\n"
8518 //ht += '<'+table' + 'id="'+tid+'"' + ' class="'+tid+'"'
8519 ht += '<'+table'
8520 + ' onkeydown="GJE_TableKeyCommand(event,this)"'
8521 //+ ' ondrag="GJE_DragEvent(event,this)"\n'
8522 //+ ' ondragend="GJE_DropEvent(event,this)"\n'
8523 //+ ' draggable="true"\n'
8524 //+ ' contenteditable="true"'
8525 + '>\n'
8526 ht += '<'+tbody>\n';
8527 for( r = 0; r < row; r++ ){
8528     ht += "<"+tr>\n"
8529     for( c = 0; c < col; c++ ){
8530         ht += "<"+td>"
8531         ht += "ABCDEFGHJKLMNPQRSTUVWXYZ".charAt(c) + r
8532         ht += "<"+td>\n"
8533     }
8534     ht += "<"+tr>\n"
8535 }
8536 ht += '<'+tbody>\n';
8537 ht += '<'+table>\n';
8538 htspan.innerHTML = ht;
8539 nt = new Text('\n')
8540 ne.appendChild(nt)
8541
8542 st = '#'+tid+' *{\n' // # for instanse specific
8543 + 'border:1px solid #aaa;\n'
8544 + 'background-color:#efe;\n'
8545 + 'color:#222;\n'
8546 + 'font-size:#14pt !important;\n'
8547 + 'font-family:monospace,Courier New !important;\n'
8548 + '}' //+ hit ESC to apply *+/\n'
8549
8550 // wish script to be included
8551 //nj = document.createElement('script')
8552 //ne.appendChild(nj)
8553 //ne.innerHTML = 'function SetStyle(e){'
8554
8555 // selector seems lost in dynamic style appending
8556 if(false){

```

```

8557     ns = document.createElement('style')
8558     ne.appendChild(ns)
8559     ns.id = tid + '.style'
8560     ns.innerHTML = '\n'+st
8561     nt = new Text('\n')
8562     ne.appendChild(nt)
8563 }
8564 setCSSofClass(tid,st); // should be in JavaScript script?
8565
8566 nx = document.createElement('textarea')
8567 ne.appendChild(nx)
8568 nx.id = tid + '-style_def'
8569 nx.setAttribute('class','GJ_StyleEditor')
8570 nx.spellcheck = false
8571 nx.cols = 60
8572 nx.rows = 10
8573 nx.innerHTML = '\n'+st
8574 nx.addEventListener('change',GJE_SetTableStyle);
8575 nx.addEventListener('keydown',GJE_StyleKeyCommand);
8576 //nx.addEventListener('click',GJE_SetTableStyle);
8577
8578 nt = new Text('\n')
8579 cwe.appendChild(nt)
8580
8581 GJE_TableNo += 1
8582 return 'created TABLE id="'+tid+"'
8583 }
8584 function GJE_NodeEdit(argv){
8585     cwe = GJE_CurElement
8586     cmd = argv[0]
8587     argv.shift()
8588     args = argv.join(' ')
8589     ret = ""
8590
8591     if( cmd == '.u' || cmd == '.un' || cmd == 'undo' ){
8592         if( GJE_NodeSaved != null ){
8593             xn = GJE_RootNode
8594             GJE_RootNode = GJE_NodeSaved
8595             GJE_NodeSaved = xn
8596             ret = '-- did undo'
8597         }else{
8598             ret = '-- could not undo'
8599         }
8600         return ret
8601     }
8602     GJE_NodeSaved = GJE_RootNode.cloneNode()
8603     if( cmd == '.c' || cmd == '.cd' || cmd == 'cd' ){
8604         if( argv[0] == null ){
8605             ne = GJE_RootNode
8606         }else
8607         if( argv[0] == '..' ){
8608             ne = cwe.parentNode
8609         }else{
8610             ne = document.getElementById(argv[0])
8611         }
8612         if( ne != null ){
8613             GJE_CurElement = ne
8614             ret = "-- current node: " + ne.id
8615         }else{
8616             ret = "-- not found: " + argv[0]
8617         }
8618     }else
8619     if( cmd == '.mkt' || cmd == '.mktable' ){
8620         makeTable(argv)
8621     }else
8622     if( cmd == '.m' || cmd == '.mk' || cmd == 'mk' ){
8623         ne = document.createElement(argv[0])
8624         //ne.id = argv[0]
8625         ret = "-- created " + ne + " under " + cwe.tagName + "#" + cwe.id
8626         cwe.appendChild(ne)
8627         if( cmd == '.m' || cmd == '.mk' ){
8628             GJE_CurElement = ne
8629         }
8630     }else
8631     if( cmd == '.n' || cmd == '.nm' || cmd == 'nm' ){
8632         cwe.id = argv[0]
8633     }else
8634     if( cmd == '.r' || cmd == '.rm' || cmd == 'rm' ){
8635     }else
8636     if( cmd == '.h' || cmd == '.sh' || cmd == 'sh' ){
8637         s = argv.join(' ')
8638         cwe.innerHTML = s
8639     }else
8640     if( cmd == '.a' || cmd == '.sa' || cmd == 'sa' ){
8641         cwe.setAttribute(argv[0],argv[1])
8642     }else
8643     if( cmd == '.l' ){
8644     }else
8645     if( cmd == '.i' || cmd == '.ih' || cmd == 'ih' ){
8646         ret = cwe.innerHTML
8647     }else
8648     if( cmd == '.p' || cmd == '.pw' || cmd == 'pw' ){
8649         ret = cwe.nodeType + " " + cwe.tagName + " " + cwe.id
8650         for( we = cwe.parentNode; we != null; ){
8651             ret += "\n" + " " + we.nodeType + " " + we.tagName + " " + we.id
8652             we = we.parentNode
8653         }
8654     }else
8655     {
8656         ret = "Command: mk | rm \n"
8657         ret += "    pw -- print current node\n"
8658         ret += "    mk type -- make node with name and type\n"
8659         ret += "    nm name -- set the id #name of current node\n"
8660         ret += "    rm name -- remove named node\n"
8661         ret += "    cd name -- change current node\n"
8662     }
8663     //alert(ret)
8664     return ret
8665 }
8666 function GJC_Command(text){
8667     lines = text.value.split('\n')
8668     line = lines[lines.length-1]
8669     argv = line.split(' ')
8670     text.value += '\n'
8671     if( argv[0] == '#' ){ argv.shift() }
8672     args0 = argv.join(' ')
8673     cmd = argv[0]
8674     argv.shift()
8675     args = argv.join(' ')
8676
8677     if( cmd == 'nolog' ){
8678         StopConsoleLog = true
8679     }else
8680     if( cmd == 'new' ){

```

```

8681     if( argv[0] == 'table' ){
8682         argv.shift()
8683         console.log('argv='+argv)
8684         text.value += makeTable(argv)
8685     }else
8686     if( argv[0] == 'console' ){
8687         text.value += GJ_NewConsole('GJ_Console')
8688     }else{
8689         text.value += '-- new { console | table }'
8690     }
8691 }else
8692 if( cmd == 'strip' ){
8693     //text.value += GJF_StripClass()
8694 }else
8695 if( cmd == 'css' ){
8696     sel = '#table_1'
8697     if(argv[0]==''0')
8698     rule1 = sel+'{color:#000 !important; background-color:#fff !important;}';
8699     else
8700     rule1 = sel+'{color:#f00 !important; background-color:#eef !important;}';
8701     document.styleSheets[3].deleteRule(0);
8702     document.styleSheets[3].insertRule(rule1,0);
8703     text.value += 'CSS rule added: '+rule1
8704 }else
8705 if( cmd == 'print' ){
8706     e = null;
8707     if( e == null ){
8708         e = document.getElementById('GJFactory_0')
8709     }
8710     if( e == null ){
8711         e = document.getElementById('GJFactory_1')
8712     }
8713     if( argv[0] != null ){
8714         id = argv[0]
8715         if( id == 'f' ){
8716             //e = document.getElementById('GJE_RootNode');
8717         }else{
8718             e = document.getElementById(id)
8719         }
8720         if( e != null ){
8721             text.value += e.outerHTML
8722         }else{
8723             text.value += "Not found: " + id
8724         }
8725     }else{
8726         text.value += GJE_RootNode.outerHTML
8727         //text.value += e.innerHTML
8728     }
8729 }else
8730 if( cmd == 'destroy' ){
8731     text.value += GJFactory_Destroy()
8732 }else
8733 if( cmd == 'save' ){
8734     e = document.getElementById('GJFactory')
8735     Permanent.setItem('GJFactory-1',e.innerHTML)
8736     text.value += "-- Saved GJFactory"
8737 }else
8738 if( cmd == 'load' ){
8739     gjf = Permanent.getItem('GJFactory-1')
8740     e = document.getElementById('GJFactory')
8741     e.innerHTML = gjf
8742     // must restore EventListener
8743     text.value += "-- EventListener was not restored"
8744 }else
8745 if( cmd.charAt(0) == '.' ){
8746     argv0 = args0.split(' ')
8747     text.value += GJE_NodeEdit(argv0)
8748 }else
8749 if( cmd == 'cont' ){
8750     bannerIsStopping = false
8751     GshMenuStop.innerHTML = "Stop"
8752 }else
8753 if( cmd == 'date' ){
8754     text.value += DateLong()
8755 }else
8756 if( cmd == 'echo' ){
8757     text.value += args
8758 }else
8759 if( cmd == 'fork' ){
8760     html_fork()
8761 }else
8762 if( cmd == 'last' ){
8763     text.value += MyHistory
8764     //h = document.createElement("span")
8765     //h.innerHTML = MyHistory
8766     //text.value += h.innerHTML
8767     //tx = MyHistory.replace("\n","")
8768     //text.value += tx.replace("<"+br>","&#10;") + "xxxx<"+br>yyyy"
8769 }else
8770 if( cmd == 'ne' ){
8771     text.value += GJE_NodeEdit(argv)
8772 }else
8773 if( cmd == 'reload' ){
8774     location.reload()
8775 }else
8776 if( cmd == 'mem' ){
8777     text.value += GJC_Memory('GJC_Storage',args,text)
8778 }else
8779 if( cmd == 'stop' ){
8780     bannerIsStopping = true
8781     GshMenuStop.innerHTML = "Start"
8782 }else
8783 if( cmd == 'who' ){
8784     text.value += "SessionId="+GJC_SessionId+" "+document.URL
8785 }else
8786 if( cmd == 'wall' ){
8787     text.value += GJC_Memory('GJC_Wall','write',text)
8788 }else
8789 {
8790     text.value += "Commands: help | echo | date | last \n"
8791     + '          new | save | load | mem \n'
8792     + '          who | wall | fork | nife'
8793 }
8794 }
8795
8796 function GJC_Input(){
8797     if( this.value.endsWith("\n") ){ // remove NL added by textarea
8798         this.value = this.value.slice(0,this.value.length-1)
8799     }
8800 }
8801
8802 var GCJ_Id = null
8803 function GJC_Resize(){
8804     GJC_Id.style.zindex = 20000

```

```

8805 //GJC_Id.style.width = window.innerWidth - 16
8806 GJC_Id.style.width = '100%';
8807 GJC_Id.style.height = 300;
8808 GJC_Id.style.backgroundColor = "rgba(0,64,16,1.0)" // blackboard color
8809 GJC_Id.style.color = "rgba(255,255,255,1.0)"
8810 }
8811 function GJC_FocusIn(){
8812 this.spellCheck = false
8813 SuppressGJShell = true
8814 this.onkeydown = GJC_Keydown
8815 GJC_Resize()
8816 }
8817 function GJC_FocusOut(){
8818 SuppressGJShell = false
8819 this.removeEventListener('keydown',GJC_Keydown);
8820 }
8821 window.addEventListener('resize',GJC_Resize);
8822
8823 function GJC_OnStorage(e){
8824 //alert('Got Message')
8825 //GJC.value += "\n((ReceivedMessage))\n"
8826 }
8827 window.addEventListener('storage',GJC_OnStorage);
8828 //window.addEventListener('storage',()=>alert('GotMessage'))
8829
8830 function GJC_Setup(gjcId){
8831 //gjcId.style.width = gsh.getBoundingClientRect().width
8832 gjcId.style.width = '100%';
8833 gjcId.value = "GJShell Console // " + GshVersion.innerHTML + "\n"
8834 //gjcId.value += "Date: " + DateLong() + "\n"
8835 gjcId.value += PS1
8836 gjcId.onfocus = GJC_FocusIn
8837 gjcId.addEventListener('input',GJC_Input);
8838 gjcId.addEventListener('focusout',GJC_FocusOut);
8839 GJC_Id = gjcId
8840 }
8841 function GJC_Clear(id){
8842 }
8843 function GJConsole_initConsole(){
8844 if( document.getElementById("GJC_0") != null ){
8845 GJC_Setup(GJC_0)
8846 }else{
8847 GJCl_Container.innerHTML = '<
8848 +<textarea id="GJC_1" class="GJConsole"><'+'/>textarea>';
8849 GJC_Setup(GJC_1)
8850 factory = document.createElement('span');
8851 gsh.appendChild(factory)
8852 GJE_RootNode = factory;
8853 GJE_CurElement = GJE_RootNode;
8854 }
8855 }
8856 var initGJCF = false;
8857 function GJConsole_initFactory(){
8858 if( initGJCF ){ return; } initGJCF = true;
8859 GShell_initKeyCommands();
8860 GJConsole_initConsole();
8861 }
8862 //GJConsole_initFactory();
8863 // TODO: focus handling
8864 </script>
8865 <style>
8866 .GJ_StyleEditor {
8867 font-size:9pt !important;
8868 font-family:Courier New, monospace !important;
8869 }
8870 </style>
8871
8872 </details>
8873 </span>
8874 <!-- ----- GJConsole END ) ----- -->
8875 */
8876
8877 /*
8878 <span id="BlinderText">
8879 <style id="BlinderTextStyle">
8880 #GJLinkView {
8881 xxposition:absolute; z-index:5000;
8882 position:relative;
8883 display:block;
8884 left:8px;
8885 color:#fff;
8886 width:800px; height:300px; resize:both;
8887 margin:0px; padding:4px;
8888 background-color:rgba(200,200,200,0.5) !important;
8889 }
8890 .MsgText {
8891 width:578px !important;
8892 resize:both !important;
8893 color:#000 !important;
8894 }
8895 .GjNote {
8896 font-family:Georgia !important;
8897 font-size:13pt !important;
8898 color:#22a !important;
8899 }
8900 .textField {
8901 display:inline;
8902 border:0.5px solid #444;
8903 border-radius:3px;
8904 color:#000; background-color:#fff;
8905 width:106pt; height:18pt;
8906 margin:2px;
8907 padding:2px;
8908 resize:none;
8909 vertical-align:middle;
8910 font-size:10pt; font-family:Courier New;
8911 }
8912 .textLabel {
8913 border:0px solid #000 !important;
8914 background-color:rgba(0,0,0,0);
8915 }
8916 .textURL {
8917 width:300pt !important;
8918 border:0px solid #000 !important;
8919 background-color:rgba(0,0,0,0);
8920 }
8921 .VisibleText {
8922 }
8923 .BlinderText {
8924 color:#000; background-color:#eee;
8925 }
8926 .joinButton {
8927 font-family:Georgia !important;
8928 font-size:11pt;

```



```

8929     line-height:1.1;
8930     height:18pt;
8931     width:50pt;
8932     padding:3px !important;
8933     text-align:center !important;
8934     border-color:#aaa !important;
8935     border-radius:5px;
8936     color:#fff; background-color:#4a4 !important;
8937     vertical-align:middle !important;
8938 }
8939 .SendButton {
8940     vertical-align:top;
8941 }
8942 .ws0_log {
8943     font-size:10pt;
8944     color:#000 !important;
8945     line-height:1.0;
8946     background-color:rgba(255,255,255,0.7) !important;
8947     font-family:Courier New,monospace !important;
8948     width:99.3%;
8949     white-space:pre;
8950 }
8951 </style>
8952
8953 <!-- Form autofill test
8954 Location: <input id="xxserv" type="text" value="https://192.168.10.1/boafm/formLogin" size="80">
8955 <form method="POST" id="xxform" action="https://192.168.10.1/boafm/formLogin">
8956 dest? <input id="XDS" name="dest" type="text" size="80" value="/index_contents.html">
8957 -->
8958 <details><summary>Form Auto. Filling</summary>
8959 <style>
8960 .xxinput { width:260pt !important; line-height:1.1 !important; margin:1px;
8961     display:inline !important; font-size:10pt !important; padding:1px !important;
8962 }
8963 </style>
8964 <span style="font-family:Courier New;color:black;font-size:12pt;" onactive="">
8965 <form method="POST" id="xxform" action="https://192.168.10.1/" style="white-space:pre;">
8966 Location: <input id="xxserv" class="xxinput" type="text" value="https://192.168.10.1/" size="80">
8967 Username: <input id="xxuser" class="xxinput" name="user" type="text" autocomplete="on">
8968 Password: <input id="xxpass" class="xxinput" name="pass" type="password" autocomplete="on">
8969 SessionId: <input id="xxssid" class="xxinput" name="SESSION_ID" type="text" size="80">
8970 <input id="xxsubm" class="xxinput" type="submit" value="Submit"></form>
8971 </span>
8972 <script>
8973 function XXSetFormAction(){
8974     xxform.setAttribute('action',xxserv.value);
8975 }
8976 xxform.setAttribute('action',xxserv.value);
8977 xxserv.addEventListener('change',XXSetFormAction);
8978 //xxserv.value = location.href;
8979 </script>
8980 </details>
8981 */
8982 /*
8983 */
8984 <details id="BlinderTextClass"><summary>BlinderText</summary>
8985 <span class="gsh-src">
8986 <span id="BlinderTextScript">
8987 // https://w3c.github.io/uievents/#event-type-keydown
8988 //
8989 // 2020-09-21 class BlinderText - textarea element not to be readable
8990 //
8991 // BlinderText attributes
8992 // bl_plainText - null
8993 // bl_hideChecksum - [false]
8994 // bl_showLength - [false]
8995 // bl_visible - [false]
8996 // data-bl_config - []
8997 // - min. length
8998 // - max. length
8999 // - acceptable charset in generate text
9000 //
9001 function BlinderChecksum(text){
9002     plain = text.bl_plainText;
9003     return strCRC32(plain,plain.length).toFixed(0);
9004 }
9005 function BlinderKeydown(ev){
9006     pass = ev.target
9007     if( ev.code == 'Enter' ){
9008         ev.preventDefault();
9009     }
9010     ev.stopPropagation()
9011 }
9012 function BlinderKeyUp1(ev){
9013     blind = ev.target
9014     if( ev.code == 'Backspace'){
9015         blind.bl_plainText = blind.bl_plainText.slice(0,blind.bl_plainText.length-1)
9016     }else
9017     if( and(ev.code == 'KeyV', ev.ctrlKey) ){
9018         blind.bl_visible = !blind.bl_visible;
9019     }else
9020     if( and(ev.code == 'KeyL', ev.ctrlKey) ){
9021         blind.bl_showLength = !blind.bl_showLength;
9022     }else
9023     if( and(ev.code == 'KeyU', ev.ctrlKey) ){
9024         blind.bl_plainText = "";
9025     }else
9026     if( and(ev.code == 'KeyR', ev.ctrlKey) ){
9027         checksum = BlinderChecksum(blind);
9028         blind.bl_plainText = checksum; // .toString(32);
9029     }else
9030     if( ev.code == 'Enter' ){
9031         ev.stopPropagation();
9032         ev.preventDefault();
9033         return;
9034     }else
9035     if( ev.key.length != 1 ){
9036         console.log('KeyUp: '+ev.code+'/'+ev.key);
9037         return;
9038     }else{
9039         blind.bl_plainText += ev.key;
9040     }
9041 }
9042 leng = blind.bl_plainText.length;
9043 //console.log('KeyUp: '+ev.code+'/'+blind.bl_plainText);
9044 checksum = BlinderChecksum(blind) % 10; // show last one digit only
9045
9046 visual = '';
9047 if( !blind.bl_hideChecksum || blind.bl_showLength ){
9048     visual += '[';
9049 }
9050 if( !blind.bl_hideChecksum ){
9051     visual += '#'+checksum.toString(10);
9052 }

```

```

9053     if( blind.bl_showLength ){
9054         visual += '/' + leng;
9055     }
9056     if( !blind.bl_hideChecksum || blind.bl_showLength ){
9057         visual += ']' ;
9058     }
9059     if( blind.bl_visible ){
9060         visual += blind.bl_plainText;
9061     }else{
9062         visual += '*'.repeat(leng);
9063     }
9064     blind.value = visual;
9065 }
9066 function BlinderKeyUp(ev){
9067     BlinderKeyUp1(ev);
9068     ev.stopPropagation();
9069 }
9070 // https://w3c.github.io/uievents/#keyboardevent
9071 // https://w3c.github.io/uievents/#uievent
9072 // https://dom.spec.whatwg.org/#event
9073 function BlinderTextEvent(){
9074     ev = event;
9075     blind = ev.target;
9076     console.log('Event '+ev.type+'@'+blind.nodeName+'#'+blind.id)
9077     if( ev.type == 'keyup' ){
9078         BlinderKeyUp(ev);
9079     }else
9080     if( ev.type == 'keydown' ){
9081         BlinderKeyDown(ev);
9082     }else{
9083         console.log('thru-event '+ev.type+'@'+blind.nodeName+'#'+blind.id)
9084     }
9085 }
9086 <!-- textarea hidden id="BlinderTextClassDef" class="textField"
9087 // onkeydown="BlinderTextEvent()" onkeyup="BlinderTextEvent()"
9088 // spellcheck="false"></textarea>
9089 <!-- textarea hidden id="gj_pass1"
9090 // class="textField BlinderText"
9091 // placeholder="Password"
9092 // onkeydown="BlinderTextEvent()"
9093 // onkeyup="BlinderTextEvent()"
9094 // spellcheck="false"></textarea>
9095 function SetupBlinderText(parent,txa,phold){
9096     if( txa == null ){
9097         txa = document.createElement('textarea');
9098         //txa.id = id;
9099     }
9100     txa.setAttribute('class','textField BlinderText');
9101     txa.setAttribute('placeholder',phold);
9102     txa.setAttribute('onkeydown','BlinderTextEvent()');
9103     txa.setAttribute('onkeyup','BlinderTextEvent()');
9104     txa.setAttribute('spellcheck','false');
9105     //txa.setAttribute('bl_plainText','false');
9106     txa.bl_plainText = '';
9107     //parent.appendChild(txa);
9108 }
9109 function DestroyBlinderText(txa){
9110     txa.removeAttribute('class');
9111     txa.removeAttribute('placeholder');
9112     txa.removeAttribute('onkeydown');
9113     txa.removeAttribute('onkeyup');
9114     txa.removeAttribute('spellcheck');
9115     txa.bl_plainText = '';
9116 }
9117 //
9118 // visible textarea like Username
9119 //
9120 function VisibleTextEvent(){
9121     if( event.code == 'Enter' ){
9122         if( event.target.NoEnter ){
9123             event.preventDefault();
9124         }
9125     }
9126     event.stopPropagation();
9127 }
9128 function SetupVisibleText(parent,txa,phold){
9129     if( false ){
9130         txa.setAttribute('class','textField VisibleText');
9131     }else{
9132         newclass = txa.getAttribute('class');
9133         if( and(newclass != null, newclass != '') ){
9134             newclass += ' ';
9135         }
9136         newclass += 'VisibleText';
9137         txa.setAttribute('class',newclass);
9138     }
9139     //console.log('SetupVisibleText class='+txa.class);
9140     txa.setAttribute('placeholder',phold);
9141     txa.setAttribute('onkeydown','VisibleTextEvent()');
9142     txa.setAttribute('onkeyup','VisibleTextEvent()');
9143     txa.setAttribute('spellcheck','false');
9144     cols = txa.getAttribute('cols');
9145     if( cols != null ){
9146         txa.style.width = '580px';
9147         //console.log('VisualText#'+txa.id+' cols='+cols)
9148     }else{
9149         //console.log('VisualText#'+txa.id+' NO cols')
9150     }
9151     rows = txa.getAttribute('rows');
9152     if( rows != null ){
9153         txa.style.height = '30px';
9154         txa.style.resize = 'both';
9155         txa.NoEnter = false;
9156     }else{
9157         txa.NoEnter = true;
9158     }
9159 }
9160 function DestroyVisibleText(txa){
9161     txa.removeAttribute('class');
9162     txa.removeAttribute('placeholder');
9163     txa.removeAttribute('onkeydown');
9164     txa.removeAttribute('onkeyup');
9165     txa.removeAttribute('spellcheck');
9166     cols = txa.removeAttribute('cols');
9167 }
9168 </span>
9169 <script>
9170 js = document.getElementById('BlinderTextScript');
9171 eval(js.innerHTML);
9172 //js.outerHTML = ""
9173 </script>
9174
9175 </span><!-- end of class="gsh-src" -->
9176 </details>

```

```

9177 </span>
9178 */
9179
9180 /*
9181 <script id="GJLinkScript">
9182 function gjkey_hash(text){
9183     return strCRC32(text,text.length) % 0x10000;
9184 }
9185 function gj_addlog(e,msg){
9186     now = (new Date().getTime() / 1000).toFixed(3);
9187     tstp = '['+now+'] '
9188     e.value += tstp + msg;
9189     e.scrollTop = e.scrollHeight;
9190 }
9191 function gj_addlog_cl(msg){
9192     ws0_log.value += '(console.log) ' + msg + '\n';
9193 }
9194 var GJ_Channel = null;
9195 var GJ_Log = null;
9196 var gjx; // the global variable
9197 function GJ_Join(){
9198     target = gj_join;
9199     if( target.value == 'Leave' ){
9200         GJ_Channel.close();
9201         GJ_Channel = null;
9202         target.value = 'Join';
9203         return;
9204     }
9205
9206     var ws0;
9207     var ws0_log;
9208
9209     sav_console_log = console.error
9210     console.error = gj_addlog_cl
9211     ws0 = new WebSocket(gj_serv.innerHTMLHTML);
9212     console.error = sav_console_log
9213
9214     GJ_Channel = ws0;
9215     ws0_log = document.getElementById('ws0_log');
9216     GJ_Log = ws0_log;
9217
9218     now = (new Date().getTime() / 1000).toFixed(3);
9219     const wsstats = ["CONNECTING","OPEN","CLOSING","CLOSED"];
9220     cst = wsstats[ws0.readyState];
9221     gj_addlog(ws0_log,'stat '+ws0.readyState+'('+cst+') : GJ Linked\n');
9222
9223     ws0.addEventListener('error', function(event){
9224         gj_addlog(ws0_log,'stat error : transport error?\n');
9225     });
9226     ws0.addEventListener('open', function(event){
9227         GJLinkView.style.zIndex = 10000;
9228         //console.log('#'+GJLinkView.id+'.zIndex='+GJLinkView.style.zIndex);
9229         datel = new Date().getTime();
9230         date2 = (datel / 1000).toFixed(3);
9231         seed = datel.toString(16);
9232
9233         // user name and key
9234         user = document.getElementById('gj_user').value;
9235         if( user.length == 0 ){
9236             gj_user.value = 'nemo';
9237             user = 'nemo';
9238         }
9239         key1 = document.getElementById('gj_ukey').bl_plainText;
9240         ukey = gjkey_hash(seed+user+key1).toString(16);
9241
9242         // session name and key
9243         chan = document.getElementById('gj_chan').value;
9244         if( chan.length == 0 ){
9245             gj_chan.value = 'main';
9246             chan = 'main';
9247         }
9248         key2 = document.getElementById('gj_ckey').bl_plainText;
9249         ckey = gjkey_hash(seed+chan+key2).toString(16);
9250
9251         msg = date2 + ' JOIN ' + user + '|' + chan + ' ' + ukey + ' ' + ckey;
9252         gj_addlog(ws0_log,'send '+msg+'\n');
9253         ws0.send(msg);
9254
9255         target.value = 'Leave';
9256         //console.log(['+date2+'] #'+target.id+' '+target.value+'\n');
9257         //gj_addlog(ws0_log,'label '+target.value+'\n');
9258     });
9259     ws0.addEventListener('message', function(event){
9260         now = (new Date().getTime() / 1000).toFixed(3);
9261         msg = event.data;
9262         if( false ){
9263             gj_addlog(ws0_log,'recv '+msg+'\n');
9264         }
9265         argv = msg.split(' ')
9266         tstamp = argv[0];
9267         argv.shift();
9268         if( argv[0] == 'reload' ){
9269             location.reload()
9270         }
9271         gjcmd = argv[0];
9272         otstamp = '';
9273         if( gjcmd == 'CAST' ){ // from reflector
9274             otstamp = argv[0];
9275             argv.shift(); // original time stamp
9276             ofrom = argv[0];
9277             argv.shift(); // original from
9278         }
9279         argv.shift(); // command
9280         from = argv[0];
9281         argv.shift(); // from|to
9282         cmdl = argv[0];
9283         argv.shift(); // xxxx command
9284
9285         if( false ){
9286             gj_addlog(ws0_log,'--'
9287                 +' tstamp='+tstamp
9288                 +' ,gjcmd='+gjcmd
9289                 +' ,from='+from
9290                 +' ,cmd=[ '+cmdl+' ]'
9291                 +' '+argv+'\n');
9292         }
9293         if( cmdl == 'auth' ){
9294             // doing authorization required
9295         }
9296         if( cmdl == 'echo' ){
9297             now = (new Date().getTime() / 1000).toFixed(3);
9298             msg = now+ ' '+RESP '+argv.join(' ');
9299             gj_addlog(ws0_log,'send '+msg+'\n');
9300             ws0.send(msg);

```

```

9301     }
9302     if( cmdl == 'eval' ){
9303         argv.shift();
9304         js = argv.join(' ');
9305         ret = eval(js); // <----- eval()
9306         gj_addlog(ws0_log,'eval '+js+ ' '+ret+'\n');
9307         now = (new Date().getTime() / 1000).toFixed(3);
9308         msg = now + ' ' + 'RESP ' + ret;
9309         ws0.send(msg);
9310         gj_addlog(ws0_log,'send '+msg+'\n')
9311     }
9312     if( cmdl == 'DRAW' ){
9313         if( false ){
9314             gj_addlog(ws0_log,'DRAW '+argv[0]+'\n')
9315         }
9316         Pointillism_RemoteDraw(argv[0]);
9317     }
9318 });
9319 ws0.addEventListener('close', function(event){
9320     if( GJ_Channel == null ){
9321         gj_addlog(ws0_log,'stat OK : GJ UnLinked\n');
9322         return;
9323     }
9324     GJ_Channel.close();
9325     GJ_Channel = null;
9326     target.value = 'Join';
9327     gj_addlog(ws0_log,'stat error : close : GJ UnLinked unexpectedly\n');
9328 });
9329 }
9330 function GJ_BcastMessageUserPass(user,chan,msgbody){
9331     now = (new Date().getTime() / 1000).toFixed(3);
9332     msg = now + ' BCAST '+user+' '+chan+' '+ msgbody;
9333     if( false ){
9334         gj_addlog(GJ_Log,'send '+msg+'\n');
9335     }
9336     GJ_Channel.send(msg);
9337 }
9338 function GJ_BcastMessage(msgbody){
9339     if( GJ_Channel == null ){
9340         gj_addlog(ws0_log,'stat error : send : GJ not Linked\n');
9341         return;
9342     }
9343     //target = event.target;
9344     user = document.getElementById('gj_user').value;
9345     chan = document.getElementById('gj_chan').value;
9346     GJ_BcastMessageUserPass(user,chan,msgbody);
9347 }
9348 function GJ_SendMessageUserPass(user,chan,msgbody){
9349     now = (new Date().getTime() / 1000).toFixed(3);
9350     msg = now + ' ISAY '+user+' '+chan+' '+ msgbody;
9351     gj_addlog(GJ_Log,'send '+msg+'\n');
9352     GJ_Channel.send(msg);
9353 }
9354 function GJ_SendMessage(msgbody){
9355     if( GJ_Channel == null ){
9356         gj_addlog(ws0_log,'stat error : send : GJ not Linked\n');
9357         return;
9358     }
9359     //target = event.target;
9360     user = document.getElementById('gj_user').value;
9361     chan = document.getElementById('gj_chan').value;
9362     GJ_SendMessageUserPass(user,chan,msgbody);
9363 }
9364 function GJ_Send(){
9365     msgbody = gj_sendText.value;
9366     GJ_SendMessage(msgbody);
9367 }
9368 </script>
9369
9370 <!-- ----- GJLINK ----- -->
9371 <!--
9372 - User can subscribe to a channel
9373 - A channel will be broadcasted
9374 - A channel can be a pattern (regular expression)
9375 - User is like From:(me) and channel is like To: or Recipient:
9376 - like VIABUS
9377 - watch message with SENDME, WATCH, CATCH, HEAR, or so
9378 - routing with path expression or name pattern (with routing with DNS like system)
9379 -->
9380 */
9381
9382 <<span id="GJLinkGolang">
9383 <<details id="GshWebSocket"><summary>Golang / JavaScript Link</summary>
9384 <<span class="gsh-src"><!-- { -->
9385 // 2020-0920 created
9386 // <a href="https://pkg.go.dev/golang.org/x/net/websocket">WS</a>
9387 // <a href="https://godoc.org/golang.org/x/net/websocket">WS</a>
9388 // INSTALL: go get golang.org/x/net/websocket
9389 // INSTALL: sudo {apt,yum} install git (if git is not installed yet)
9390 // import "golang.org/x/net/websocket"
9391 const gshws_origin = "http://localhost:9999"
9392 const gshws_server = "localhost:9999"
9393 const gshws_port = 9999
9394 const gshws_path = "gjlink1"
9395 const gshws_url = "ws://" + gshws_server + "/" + gshws_path
9396 const GSHWS_MSGSIZE = (8*1024)
9397 func fmtstring(fmts string, params ...interface{})(string){
9398     return fmt.Sprintf(fmts,params...)
9399 }
9400 func GSHWS_MARK(what string)(string){
9401     now := time.Now()
9402     us := fmtstring("%06d",now.Nanosecond() / 1000)
9403     mark := ""
9404     if( !AtConsoleLineTop ){
9405         mark += "\n"
9406         AtConsoleLineTop = true
9407     }
9408     mark += "[" + now.Format(time.Stamp) + "." + us + "] -GJ- " + what + ": "
9409     return mark
9410 }
9411 func gchk(what string,err error){
9412     if( err != nil ){
9413         panic(GSHWS_MARK(what)+err.Error())
9414     }
9415 }
9416 func glog(what string, fmts string, params ...interface{}){
9417     fmt.Print(GSHWS_MARK(what))
9418     fmt.Printf(fmts+"\n",params...)
9419 }
9420
9421 var WSV = []*websocket.Conn{}
9422 func jsend(argv []string){
9423     if len(argv) <= 1 {
9424         fmt.Printf("--Ij %v [-m] command arguments\n",argv[0])

```

```

9425     return
9426 }
9427 argv = argv[1:]
9428 if( len(WSV) == 0 ){
9429     fmt.Printf("--Bj-- No link now\n")
9430     return
9431 }
9432 if( 1 < len(WSV) ){
9433     fmt.Printf("--Ij-- multiple links (%v)\n",len(WSV))
9434 }
9435
9436 multicast := false // should be filtered with regexp
9437 if( 0 < len(argv) && argv[0] == "-m" ){
9438     multicast = true
9439     argv = argv[1:]
9440 }
9441 args := strings.Join(argv, " ")
9442
9443 now := time.Now()
9444 msec := now.UnixNano() / 1000000;
9445 tstamp := fmtstring("%.3f",float64(msec)/1000.0)
9446 msg := fmtstring("%v SEND gshell" + %v",tstamp,args)
9447
9448 if( multicast ){
9449     for i,ws := range WSV {
9450         wn,werr := ws.Write([]byte(msg))
9451         if( werr != nil ){
9452             fmt.Printf("[%v] wn=%v, werr=%v\n",i,wn,werr)
9453         }
9454         glog("SQ",fmtstring("(%v) %v",wn,msg))
9455     }
9456 }else{
9457     i := 0
9458     ws := WSV[i]
9459     wn,werr := ws.Write([]byte(msg))
9460     if( werr != nil ){
9461         fmt.Printf("[%v] wn=%v, werr=%v\n",i,wn,werr)
9462     }
9463     glog("SQ",fmtstring("(%v) %v",wn,msg))
9464 }
9465 }
9466 func ws_broadcast(msg string)(wn int,werr error){
9467     for i,ws := range WSV {
9468         wn,werr = ws.Write([]byte(msg))
9469         if( werr != nil ){
9470             fmt.Printf("[%v] wn=%v, werr=%v\n",i,wn,werr)
9471         }
9472         glog("SQ",fmtstring("(%v) %v",wn,msg))
9473     }
9474     return wn,werr;
9475 }
9476 func serv1(ws *websocket.Conn) {
9477     WSV = append(WSV,ws)
9478     //fmt.Println("\n")
9479     glog("CO", "accepted connections[%v]",len(WSV))
9480     //remoteAddr := ws.RemoteAddr
9481     //fmt.Printf("-- accepted %v\n",remoteAddr)
9482     //fmt.Printf("-- accepted %v\n",ws.Config())
9483     //fmt.Printf("-- accepted %v\n",ws.Config().Header)
9484     //fmt.Printf("-- accepted %v // %v\n",ws,serv1)
9485
9486     var reqb = make([]byte,GSHWS_MSGSIZE)
9487     for {
9488         rn, rerr := ws.Read(reqb)
9489         if( rerr != nil || rn < 0 ){
9490             glog("SQ",fmtstring("(%v,%v)",rn,rerr))
9491             break
9492         }
9493         req := string(reqb[0:rn])
9494         glog("SQ",fmtstring("(%v) %v",rn,req))
9495
9496         margv := strings.Split(req, " ");
9497         margv = margv[1:]
9498         if( 0 < len(margv) ){
9499             if( margv[0] == "RESP" ){
9500                 // should forward to the destination
9501                 continue;
9502             }
9503         }
9504         now := time.Now()
9505         msec := now.UnixNano() / 1000000;
9506         tstamp := fmtstring("%.3f",float64(msec)/1000.0)
9507         res := fmtstring("%v "+CAST+" %v",tstamp,req)
9508
9509         wn := 0;
9510         werr := error(nil);
9511         if( 0 < len(margv) && margv[0] == "BCAST" ){
9512             wn, werr = ws_broadcast(res);
9513         }else{
9514             wn, werr = ws.Write([]byte(res))
9515         }
9516         gchk("SE",werr)
9517         glog("SR",fmtstring("(%v) %v",wn,string(res)))
9518     }
9519     glog("SF","WS response finish")
9520
9521     wsv := []*websocket.Conn{}
9522     wsx := 0
9523     for i,v := range WSV {
9524         if( v != ws ){
9525             wsx = i
9526             wsv = append(wsv,v)
9527         }
9528     }
9529     WSV = wsv
9530     //glog("CO","closed %v",ws)
9531     glog("CO","closed connection [%v/%v]",wsx+1,len(WSV)+1)
9532     ws.Close()
9533 }
9534 // url := [scheme://host[:port]][/path]
9535 func decomp_URL(url string){
9536 }
9537 func full_wsURL(){
9538 }
9539 func gj_server(argv []string) {
9540     gjserv := gshws_url
9541     gjport := gshws_server
9542     gjpath := gshws_path
9543     gjscheme := "ws"
9544
9545     //cmd := argv[0]
9546     argv = argv[1:]
9547     if( 1 <= len(argv) ){
9548         serv := argv[0]

```

```

9549     if( 0 < strings.Index(serv,":/") ){
9550         schemev := strings.Split(serv,"://")
9551         gjscheme = schemev[0]
9552         serv = schemev[1]
9553     }
9554     if( 0 < strings.Index(serv,"/") ){
9555         pathv := strings.Split(serv,"/")
9556         serv = pathv[0]
9557         gjpath = pathv[1]
9558     }
9559     servv := strings.Split(serv,":")
9560     host := "localhost"
9561     port := 9999
9562     if( servv[0] != "" ){
9563         host = servv[0]
9564     }
9565     if( len(servv) == 2 ){
9566         fmt.Sscanf(servv[1],"%d",&port)
9567     }
9568     //glog("LC", "hostport=%v (%v : %v)",servv,host,port)
9569     gjport = fmt.Sprintf("%v:%v",host,port)
9570     gjserv = gjscheme + "://" + gjport + "/" + gjpath
9571 }
9572 glog("LS",fmtstring("listening at %v",gjserv))
9573 http.Handle("/"+gjpath,websocket.Handler(serv1))
9574 err := error(nil)
9575 if( gjscheme == "wss" ){
9576     // https://golang.org/pkg/net/http/#ListenAndServeTLS
9577     //err = http.ListenAndServeTLS(gjport,nil)
9578 }else{
9579     err = http.ListenAndServe(gjport,nil)
9580 }
9581 gchk("LE",err)
9582 }
9583
9584 func gj_client(argv []string) {
9585     glog("CS",fmtstring("connecting to %v",gshws_url))
9586     ws, err := websocket.Dial(gshws_url,"",gshws_origin)
9587     gchk("C",err)
9588
9589     var resb = make([]byte, GSHWS_MSGSIZE)
9590     for qi := 0; qi < 3; qi++ {
9591         req := fmtstring("Hello, GShell! (%v)",qi)
9592         wn, werr := ws.Write([]byte(req))
9593         glog("QM",fmtstring("(%v) %v",wn,req))
9594         gchk("QE",werr)
9595         rn, rerr := ws.Read(resb)
9596         gchk("RE",rerr)
9597         glog("RM",fmtstring("(%v) %v",rn,string(resb)))
9598     }
9599     glog("CF","WS request finish")
9600 }
9601 //</span><!-- end of class="gsh-src" -->
9602 //</details></span>
9603
9604 /*
9605 <details id="GJLink_Section"><summary>GJ Link</summary>
9606 <span id="GJLinkView" class="GJLinkView">
9607 <p>
9608 <note class="GjNote">Execute command "gsh gj server" on the localhost and push the Join button:</note>
9609 </p>
9610
9611 <span id="GJLink_1">
9612 <div id="GJLink_ServerSet"></div>
9613
9614 <div>
9615 <input id="gj_join" type="button" class="joinButton" onclick="GJ_Join()" value="Join">
9616 <span id="GJLink_Account"></span>
9617 </div>
9618
9619 <div>
9620 <input id="gj_sendButton" type="button" class="joinButton SendButton" onclick="GJ_Send()" value="Send" data-bodyid="gj_sendText">
9621 <span id="GJLink_SendArea"></span>
9622 </div>
9623
9624 <div id="ws0_log_container"></div>
9625 </span>
9626 </span>
9627
9628 <script>
9629 function setupGJLinkArea(){
9630     GJLink_ServerSet.innerHTML = '<'+<span id="gj_serv_label"
9631 + ' class="textField textLabel">Server: <'+</span>'
9632 + '<'+<span id="gj_serv" class="textField textURL" contenteditable><'+</span>';
9633
9634     GJLink_Account.innerHTML = '<'+<textarea id="gj_user" class="textField"><'+</textarea>'
9635 + '<'+<textarea id="gj_ukey" class="textField"><'+</textarea>'
9636 + '<'+<textarea id="gj_chan" class="textField"><'+</textarea>'
9637 + '<'+<textarea id="gj_ckekey" class="textField"><'+</textarea>';
9638
9639     GJLink_SendArea.innerHTML =
9640 '<'+<textarea id="gj_sendText" class="textField MsggText" cols=60 rows=2><'+</textarea>';
9641
9642     ws0_log_container.innerHTML = '<'+<textarea id="ws0_log" class="ws0_log"
9643 + ' cols=100 rows=10 spellcheck="false"><'+</textarea>';
9644 }
9645 function clearGJLinkArea(){
9646     GJLink_ServerSet.innerHTML = "";
9647     GJLink_Account.innerHTML = "";
9648     GJLink_SendArea.innerHTML = "";
9649     ws0_log_container.innerHTML = "";
9650 }
9651 </script>
9652
9653 <script>
9654 function SetupGJLink(){
9655     setupGJLinkArea();
9656     SetupVisibleText(GJLink_1,gj_serv,'GJLinkSv');
9657     SetupVisibleText(GJLink_1,gj_user,'UserName');
9658     SetupBlinderText(GJLink_1,gj_ukey,'UserKey');
9659     SetupVisibleText(GJLink_1,gj_chan,'ChannelName');
9660     SetupBlinderText(GJLink_1,gj_ckekey,'ChannelKey');
9661     SetupVisibleText(GJLink_1,gj_sendText,'Message');
9662     gj_serv.innerHTML = 'ws://localhost:9999/gjlink1'
9663 }
9664 function GJLink_init(){
9665     SetupGJLink();
9666 }
9667 function iselem(eid){
9668     return document.getElementById(eid);
9669 }
9670 function DestroyGJLink1(){
9671     clearGJLinkArea();
9672     if( !iselem('gj_user') ){

```

```

9673     return;
9674 }
9675 if( gj_serv_label.parentNode != gj_user ){
9676     return;
9677 }
9678 if( iselem('gj_serv_label') ) gj_user.parentNode.removeChild(gj_serv_label);
9679 if( iselem('gj_serv') ) gj_user.parentNode.removeChild(gj_serv);
9680 if( iselem('gj_user') ) gj_user.parentNode.removeChild(gj_user);
9681 if( iselem('gj_ukey') ) gj_ukey.parentNode.removeChild(gj_ukey);
9682 if( iselem('gj_chan') ) gj_chan.parentNode.removeChild(gj_chan);
9683 if( iselem('gj_ckey') ) gj_ckey.parentNode.removeChild(gj_ckey);
9684 if( iselem('gj_sendText') ) gj_sendText.parentNode.removeChild(gj_sendText);
9685 if( iselem('ws0_log') ) ws0_log.parentNode.removeChild(ws0_log);
9686 }
9687 DestroyGJLink = DestroyGJLink1;
9688 </script>
9689 </details>
9690 */
9691 /*
9692 <style>
9693 .GJDigest {
9694     display:none;
9695 }
9696 </style>
9697 <script id="HtmlCodeview-script">
9698 function showNodeAsHtmlSourceX(otxa,code,prefix,postfix,sign){
9699     txa = document.createElement('textare');
9700     txa.id = otxa.id;
9701     txa.setAttribute('class','HtmlCodeviewText');
9702     otxa.parentNode.replaceChild(txa,otxa);
9703     txa.setAttribute('spellcheck','false');
9704     //txa.value = code.innerHTML;
9705     //txa.innerHTML = code.innerHTML;
9706     txa.innerHTML = prefix + code.outerHTML + postfix;
9707     if( sign ){
9708         text = txa.value;
9709         tlen = txa.value.length;
9710         digest = strCRC32(text,tlen) + ' ' + tlen
9711         + ' ' + code.id + ' (' + DateShort() + ')';
9712         //alert('digest: '+digest);
9713         console.log('digest: '+digest);
9714         txa.innerHTML += '<'+span class="GJDigest">'+digest+'<'+span>\n';
9715     }
9716     txa.style.display = "block";
9717     txa.style.width = "100%";
9718     txa.style.height = "300px";
9719 }
9720 }
9721 function showHtmlCodeX(otxa,code,prefix,postfix,sign){
9722     if( event.target.value == 'ShowCode' ){
9723         showNodeAsHtmlSourceX(otxa,code,prefix,postfix,sign);
9724         event.target.value = 'HideCode';
9725     }else{
9726         otxa.style.display = "none";
9727         event.target.value = 'ShowCode';
9728     }
9729 }
9730 function showNodeAsHtmlSource(otxa,code){
9731     showNodeAsHtmlSourceX(otxa,code,'','');
9732 }
9733 function showHtmlCode(otxa,code){
9734     if( event.target.value == 'ShowCode' ){
9735         showNodeAsHtmlSource(otxa,code);
9736         event.target.value = 'HideCode';
9737     }else{
9738         otxa.style.display = "none";
9739         event.target.value = 'ShowCode';
9740     }
9741 }
9742 </script>
9743 <style id="HtmlCodeview-style">
9744 .HtmlCodeviewText {
9745     font-size:10pt;
9746     font-family:Courier New;
9747     white-space:pre;
9748 }
9749 .HtmlCodeViewButton {
9750     padding:2pt !important;
9751     line-height:1.1 !important;
9752     border:2px inset #bbb !important;
9753     font-size:11pt !important;
9754     font-weight:normal !important;
9755     font-family:Georgia !important;
9756     border-radius:3px !important;
9757     color:#ddd; background-color:#228 !important;
9758 }
9759 </style>
9760 */
9761 /*
9762 <details><summary>Live HTML Snapshot</summary>
9763 <span id="LiveHTML">
9764 <!-- ----- HTML Snapshot: Edit, save and load // 2020-0924 SatoxITS { -->
9765 <input class="HtmlCodeViewButton" type="button" value="ShowCode" onclick="showLiveHTMLCode()">
9766 <span id="LiveHTML_Codeview"></span>
9767 <script id="LiveHtmlScript">
9768 function showLiveHTMLCode(){
9769     showHtmlCode(LiveHTML_Codeview,LiveHTML);
9770 }
9771 }
9772 var _editable = false;
9773 var savSuppressGJShell = false;
9774 function ToggleEditMode(){
9775     _editable = !_editable;
9776     if( _editable ){
9777         savSuppressGJShell = SuppressGJShell;
9778         SuppressGJShell = true;
9779         gsh.setAttribute('contenteditable','true');
9780         GshMenuEdit.innerHTML = 'Lock';
9781         GshMenuEdit.style.color = 'rgba(255,0,0,1)';
9782         GshMenuEdit.style.backgroundColor = 'rgba(255,255,255,1)';
9783     }else{
9784         SuppressGJShell = savSuppressGJShell;
9785         gsh.setAttribute('contenteditable','false');
9786         GshMenuEdit.innerHTML = 'Edit';
9787         GshMenuEdit.style.color = 'rgba(16,160,16,1)';
9788         GshMenuEdit.style.backgroundColor = 'rgba(255,255,255,1)';
9789     }
9790 }
9791 function html_edit(){
9792     ToggleEditMode();
9793 }
9794 }
9795 // Live HTML (DOM) Snapshot onto browser's localStorage
9796 // 2020-0923 SatoxITS

```

```

9797 var htRoot = gsh // -- Element-ID, should be selectable
9798 const snappedHTML = 'SnappedHTML'; // Item-ID of the HTML data in localStogate
9799 // -- should be a [map] of URL
9800 // -- should be with CSSOM as inline script
9801 const htVersionTag = 'VersionTag'; // VesionTag Element-ID in the HTML (in DOM)
9802 function showVersion(note,w,v,u,t){
9803   w.alert(note+' : ' + v + '\n'
9804     + '-- URL: ' + u + '\n'
9805     + '-- Time: ' + t + ' (' + DateLong0(t*1000) + ') '
9806   );
9807 }
9808 function html_save(){
9809   u = document.URL;
9810   t = new Date().getTime() / 1000;
9811   v = '<' + span id="'+htVersionTag+' data-url="'+u+'" data-time="'+t+'>';
9812   v += '<'+/span>\n';
9813   h += v + htRoot.outerHTML;
9814   localStorage.setItem(snappedHTML,h);
9815   showVersion("Saved",window,v,u,t);
9816 }
9817 function html_load(){
9818   h = localStorage.getItem(snappedHTML);
9819   if( h == null ){
9820     alert('No snapshot taken yet');
9821     return;
9822   }
9823   w = window.open('','');
9824   d = w.document;
9825   d.write(h);
9826   w.focus();
9827   html_ver1("Loaded",w,d);
9828 }
9829 function html_ver1(note,w,d){
9830   if( (v = d.getElementById(htVersionTag)) != null ){
9831     h = v.outerHTML;
9832     u = v.getAttribute('data-url');
9833     t = v.getAttribute('data-time');
9834   }else{
9835     h = 'No version info. in the page';
9836     u = '';
9837     t = 0;
9838   }
9839   showVersion(note,w,v,u,t);
9840 }
9841 function html_ver0(){
9842   html_ver1("Version",window,document);
9843 }
9844 </script>
9845 <!-- LiveHTML -->
9846 </span>
9847 </details>
9848 */
9849 /*
9850 <details><summary>Event sharing</summary>
9851 <span id="EventSharingCodeSpan">
9852 <!-- ----- Event sharing // 2020-0925 SatoxITS { -->
9853
9854 <div id="iftestTemplate" class="iftest" hidden="">
9855 <style> .iftestbody{ color:#f22;font-family:Georgia;font-size:10pt;} </style>
9856 <span id="frameBody" class="iftestbody" onclick="frameClick()"><script>
9857   function docadd(txt){
9858     document.body.append(txt);
9859     window.scrollTo(0,100000);
9860   }
9861   function frameClick(){
9862     xy = '(x='+event.x + ' y='+event.y+')';
9863     //docadd('Got Click on #' +event.target.id+' +xy+ '\n');
9864     docadd('Got Click on #' +Fid.value+' , '+xy+ '\n');
9865     window.scrollTo(0,100000);
9866     window.parent.postMessage('OnClick: '+xy, '*');
9867   }
9868   function frameMouseMove(){
9869     if( false ){
9870       document.body.append('Mousemove on #' +event.target.id+' '
9871         + 'x='+event.x + ' y='+event.y + '\n');
9872       peerWin = window.frames.iframe1;
9873       document.body.append('Send to peer #' +peerWin+ ' ' + '\n');
9874       window.scrollTo(0,100000);
9875       peerWin.postMessage('Hi!', '*');
9876     }
9877   }
9878   function frameKeydown(){
9879     msg = 'Got Keydown: #' +Fid.value+' , (' +event.code+' )';
9880     docadd(msg+'\n');
9881     window.parent.postMessage(msg, '*');
9882   }
9883   function frameOnMessage(){
9884     docadd('Message ' + event.data + '\n');
9885     window.scrollTo(0,100000);
9886   }
9887 }
9888 if( document.getElementById('Fid') ){
9889   frameBody.id = Fid.value;
9890   h = '';
9891   h += '<'+style>*{';
9892   h += 'font-size:10pt;white-space:pre-wrap;';
9893   h += 'font-family:Courier New;';
9894   h += '<'+/style>';
9895   h += 'I am '+Fid.value+'\n';
9896   document.write(h);
9897   window.addEventListener('click',frameClick);
9898   window.addEventListener('keydown',frameKeydown);
9899   window.addEventListener('message',frameOnMessage);
9900   window.addEventListener('mousemove',frameMouseMove);
9901   window.parent.postMessage('Hi parent, I am '+Fid.value, '*');
9902 }
9903 </script></span></div>
9904
9905 <style>.iframeTest{margin:3px;resize:both;width:370px;height:120px;}</style>
9906 <h2>Inter-window communicaiton</h2>
9907 <note>
9908 frame0 >>> frame1 and frame2<br>
9909 frame1 >>> frame0 and frame2<br>
9910 frame2 >>> frame0 and frame1<br>
9911 </note>
9912 <div id="iframe-test">
9913 <pre id="iframeHost" style="border:1px;font-family:Courier New;font-size:10pt;"></pre>
9914 <iframe id="iframe0" title="iframe0" class="iframeTest" style=""></iframe>
9915 <iframe id="iframe1" title="iframe1" class="iframeTest"></iframe>
9916 <iframe id="iframe2" title="iframe2" class="iframeTest"></iframe>
9917 </div>
9918 <script id="if0-test-script">

```



```

9921 function InterFrameComm_init(){
9922     setupFrames0();
9923     setupFrames12();
9924 }
9925 function setFrameSrcdoc(dst,src){
9926     if( true ){
9927         dst.contentWindow.document.write(src);
9928         // this makes browser waite close, and crash if accumulated !?
9929         // so it should be closed after write
9930         dst.contentWindow.document.close();
9931     }else{
9932         // to be erased before source dump
9933         // but should be set for live snapshot
9934         dst.srcdoc = src;
9935     }
9936 }
9937 function setupFrames0(){
9938     ibody = iframe0.contentWindow.document.body;
9939     iframe0.style.width = "755px"
9940     //iframeHost.innerHTML = "Message exchange at iframes' host:\n";
9941     window.addEventListener('message',messageFromChild);
9942
9943     if0 = '';
9944     if0 += '<'+pre style="font-family:Courier New;">';
9945     if0 += '<input id="Fid" value="iframe0">';
9946     if0 += iftestTemplate.innerHTML;
9947     setFrameSrcdoc(iframe0,if0);
9948
9949     function clickOnChild(){
9950         console.log('clickOn #' +this.id);
9951     }
9952     function moveOnChild(){
9953         console.log('moveOn #' +this.id);
9954     }
9955     iframe0.contentWindow.document.body.style.setProperty('white-space','pre');
9956     iframe0.contentWindow.document.body.style.setProperty('font-size','9pt');
9957 }
9958 function setupFrames12(){
9959     if1 = <input id="Fid" value="iframe1">;
9960     if1 += iftestTemplate.innerHTML;
9961     setFrameSrcdoc(iframe1,if1);
9962     //iframe1.name = 'iframe1'; // this seems break contentWindow
9963
9964     if2 = <input id="Fid" value="iframe2">;
9965     if2 += iftestTemplate.innerHTML;
9966     setFrameSrcdoc(iframe2,if2);
9967
9968     iframe1.addEventListener('message',messageFromChild);
9969     //iframe1.addEventListener('mouseover',moveOnChild);
9970     iframe2.addEventListener('message',messageFromChild);
9971     //iframe2.addEventListener('mouseover',moveOnChild);
9972     iframe1.contentWindow.postMessage('[parent0] Hi iframe1 -- from parent.','*');
9973     //iframe1.contentWindow.postMessage('Your peer is '+iframe2.contentWindow,'*');
9974     iframe2.contentWindow.postMessage('[parent0] Hi iframe2 -- from parent.','*');
9975     //iframe2.contentWindow.postMessage('Your peer is '+iframe1.contentWindow,'*');
9976 }
9977 function messageFromChild(){
9978     from = null;
9979     forw = null;
9980     if( event.source == iframe0.contentWindow ){
9981         from = 'iframe0'
9982         forw = 'iframe12';
9983     }else
9984     if( event.source == iframe1.contentWindow ){
9985         from = '[iframe1]'
9986         forw = 'iframe2';
9987     }else
9988     if( event.source == iframe2.contentWindow ){
9989         from = '[iframe2]'
9990         forw = 'iframe1';
9991     }else
9992     {
9993         iframeHost.innerHTML += 'Message [unknown] '
9994         + ' orig=' + event.origin
9995         + ' data=' + event.data
9996         //+ ' from=' + event.source
9997         ;
9998     }
9999     msglog1 = from + event.data + ' -- '
10000     + ' from=' + event.source
10001     + ' orig=' + event.origin
10002     + ' name=' + event.source.name
10003     //+ ' port=' + event.ports
10004     //+ ' evid=' + event.lastEventId
10005     + '\n'
10006     ;
10007     if( true ){
10008         if( forw == 'iframe1' || forw == 'iframe12' ){
10009             iframe1.contentWindow.postMessage(from+event.data);
10010         }
10011         if( forw == 'iframe2' || forw == 'iframe12' ){
10012             iframe2.contentWindow.postMessage(from+event.data);
10013         }
10014     }
10015     txtadd0(msglog1);
10016
10017     function txtadd0(txt){
10018         iframe0.contentWindow.document.body.append(txt);
10019         iframe0.contentWindow.scrollTo(0,100000);
10020     }
10021 }
10022 function es_ShowSelf(){
10023     iframe1.setAttribute('src',document.URL);
10024     iframe2.setAttribute('src',document.URL);
10025 }
10026</script>
10027
10028<input class="HtmlCodeViewButton" type="button" value="ShowSelf" onclick="es_ShowSelf()">
10029<input class="HtmlCodeViewButton" type="button" value="ShowCode" onclick="es_showHtmlCode()">
10030<span id="EventSharingCodeview"></span>
10031<script id="EventSharingScript">
10032function es_showHtmlCode(){
10033     showHtmlCode(EventSharingCodeview,EventSharingCodeSpan);
10034}
10035DestroyEventSharingCodeview = function(){
10036     //EventSharingCodeview.parentNode.removeChild(EventSharingCodeview);
10037     EventSharingCodeview.innerHTML = "";
10038     iframe0.style = "";
10039     //iframe0.srcdoc = "erased";
10040     //iframe1.srcdoc = "erased";
10041     //iframe2.srcdoc = "erased";
10042 }
10043</script>
10044<!-- EventSharing -->

```

```
10045</span>
10046</details>
10047*/
10048
10049/*
10050<!-- ----- "GShell Inside" Notification { -->
10051<script id="script-gshell-inside">
10052var notices = 0;
10053function noticeGShellInside(){
10054    ver = '';
10055    if( ver = document.getElementById('GshVersion') ){
10056        ver = ver.innerHTML;
10057    }
10058    console.log('GJShell Inside (^~^)' + ver);
10059    notices += 1;
10060    if( 2 <= notices ){
10061        document.removeEventListener('mousemove',noticeGShellInside);
10062    }
10063}
10064document.addEventListener('mousemove',noticeGShellInside);
10065noticeGShellInside();
10066
10067const FooterName = 'GshFooter'
10068function DestroyFooter(){
10069    if( (footer = document.getElementById(FooterName)) != null ){
10070        //footer.parentNode.removeChild(footer);
10071        empty = document.createElement('div');
10072        empty.id = 'GshFooter0';
10073        footer.parentNode.replaceChild(empty,footer);
10074    }
10075}
10076function showFooter(){
10077    footer = document.createElement('div');
10078    footer.id = FooterName;
10079    footer.style.backgroundImage = "url("+ITSmoreQR+")";
10080    //GshFooter0.parentNode.appendChild(footer);
10081    GshFooter0.parentNode.replaceChild(footer,GshFooter0);
10082}
10083</script>
10084<!-- } -->
10085
10086<!--
10087    border:20px inset #888;
10088-->
10089
10090//<span id="VirtualDesktopCodeSpan">
10091/*
10092<details id="VirtualDesktopDetails"><summary>Virtual Desktop</summary>
10093<!-- ----- Web Virtual Desktop // 2020-0927 SatoxITS { -->
10094<style>
10095.VirtualSpace {
10096    z-index:0;
10097    xwidth:1280px !important; xheight:720px !important;
10098    width:5120px; height:2880px;
10099    border-width:0px;
10100    xbackground-color:rgba(32,32,160,0.8);
10101    xbackground-image:url("WD-WallPaper03.png");
10102    xbackground-size:100% 100%;
10103    color:#22a;font-family:Georgia;font-size:10pt;
10104    xxoverflow:scroll;
10105}
10106.VirtualGrid {
10107    z-index:0;
10108    position:absolute;
10109    width:800px; height:500px;
10110    border:1px inset #fff;
10111    color:rgba(192,255,192,0.8);
10112    font-family:Georgia, Courier New;
10113    text-align:right;
10114    vertical-align:middle;
10115    font-size:200px;
10116    text-shadow:4px 4px #ccf;
10117}
10118.WD_GridScroll {
10119    z-index:100000;
10120    background-color:rgba(200,200,200,0.1);
10121}
10122.VirtualDesktop {
10123    z-index:0;
10124    position:relative;
10125    resize:both !important;
10126    overflow:scroll;
10127    display:block;
10128    min-width:120px !important; min-height:60px !important;
10129    width:800px;
10130    height:500px;
10131    border:10px inset #228;
10132    border-width:30px; border-radius:20px;
10133    background-image:url("WD-WallPaper03.png");
10134    background-size:100% 100%;
10135    color:#22a;font-family:Georgia;font-size:10pt;
10136}
10137.comment {
10138    // overflow=scroll seems to bound childrens' view in the element span
10139    // specifying overflow seems fix the position of the element
10140}
10141.VirtualBrowserSpan {
10142    z-index:10;
10143    xxxborder:0.5px dashed #fff !important;
10144    border-color:rgba(255,255,255,0.5) !important;
10145    position:relative;
10146    left:100px;
10147    top:100px;
10148    display:block;
10149    resize:both !important;
10150    width:540px;
10151    height:320px;
10152    min-width:40px !important; min-height:20px !important;
10153    max-width:5120px !important; max-height:2880px !important;
10154    background-color:rgba(255,200,255,0.1);
10155    xoverflow:scroll;
10156}
10157.xVirtualBrowserLocationBar:focus {
10158    color:#f00;
10159    background-color:rgba(255,128,128,0.2);
10160}
10161.xVirtualBrowserLocationBar:active {
10162    color:#f00;
10163    background-color:rgba(128,255,128,0.2);
10164}
10165a.VirtualBrowserLocation {
10166    color:#ccc !important;
10167    text-decoration:none !important;
10168}
```

```
10169a.VirtualBrowserLocation:hover {
10170   color:#fff !important;
10171   text-decoration:underline;
10172}
10173.VirtualBrowserLocationBar {
10174   position:absolute;
10175   z-index:100000;
10176   display:block;
10177   width:400px;
10178   height:20px;
10179   padding-left:2px;
10180   line-height:1.1;
10181   vertical-align:middle;
10182   font-size:14px;
10183   color:#fff;
10184   background-color:rgba(128,128,128,0.2);
10185   font-family:Georgia;
10186}
10187.VirtualBrowserCommandBar {
10188   position:absolute;
10189   z-index:200000;
10190   xxxdisplay:inline;
10191   display:block;
10192   width:60px;
10193   height:20px;
10194   line-height:1.1;
10195   vertical-align:middle;
10196   font-size:14px;
10197   color:#fe4;
10198   background-color:rgba(128,128,128,0.1);
10199   font-family:Georgia;
10200   text-align:left;
10201   left:404px;
10202}
10203.VirtualBrowserFrame {
10204   xxposition:relative;
10205   position:absolute;
10206   xxxdisplay:inline;
10207   display:block;
10208   z-index:10;
10209   resize:both !important;
10210   width:480px; height:240px;
10211   min-width:60px; min-height:30px;
10212   max-width:5120px; max-height:2880px;
10213   border-radius:6px;
10214   background-color:rgba(255,255,255,0.9);
10215   border-top:20px solid;
10216   border-right:4px solid;
10217   border-bottom:10px solid;
10218}
10219.WinFavicon {
10220   width:16px;
10221   height:16px;
10222   margin:1px;
10223   margin-right:3px;
10224   vertical-align:middle;
10225   background-color:rgba(255,255,255,1.0);
10226}
10227.VirtualDesktopMenuBar {
10228   xposition:absolute;
10229   color:#fff;
10230   font-size:7pt;
10231   text-align:right;
10232   padding-right:4px;
10233   background-color:rgba(128,128,128,0.7);
10234}
10235.VirtualDesktopCalender {
10236   color:#fff;
10237   font-size:22pt;
10238   text-align:right;
10239   padding-right:4px;
10240   xbackground-color:rgba(255,255,255,0.2);
10241}
10242.xxxinput { width:260pt !important; line-height:1.1 !important; margin:1px;
10243   display:inline !important; font-size:10pt !important; padding:1px !important;
10244}
10245.WD_Config {
10246   display:inline !important;
10247   padding:2px !important;
10248   font-size:10pt !important;
10249   width:60pt !important;
10250   height:12pt !important;
10251   line-height:1.0pt !important;
10252   height:15pt !important;
10253}
10254.WD_Button {
10255   display:inline !important;
10256   padding:2px !important;
10257   color:#fff !important;
10258   background-color:#228 !important;
10259   font-size:10pt !important;
10260   width:60pt !important;
10261   height:12pt !important;
10262   line-height:1.0pt !important;
10263   height:16pt !important;
10264   border:2px inset #44a !important;
10265}
10266.WD_Href {
10267   display:inline !important;
10268   padding:2px !important;
10269   font-size:9pt !important;
10270   width:120pt !important;
10271   height:12pt !important;
10272   line-height:1.0pt !important;
10273   height:15pt !important;
10274}
10275
10276.LiveHtmlCodeviewText {
10277   font-size:10pt;
10278   font-family:Courier New;
10279   xwhite-space:pre;
10280}
10281
10282.WD_Panel {
10283   x-index:100 !important;
10284   color:#000 !important;
10285   margin-left:25px !important;
10286   width:800px !important;
10287   padding:4px !important;
10288   border:1px solid #888 !important;
10289   border-radius:6px !important;
10290   background-color:rgba(220,220,220,0.9) !important;
10291   font-size:9pt;
10292   font-family:Courier New;
```

```

10293}
10294.WD_Help {
10295  font-size:10pt !important;
10296  font-family:Courier New;
10297  line-height:1.2 !important;
10298  color:#000 !important;
10299  width:100% !important;
10300  background-color:rgba(240,240,255,0.8) !important;
10301}
10302
10303.WB_Zoom {
10304}
10305</style>
10306<h2>CosmoScreen 0.0.8</h2>
10307<menu>
10308<span id="WD_Help_1" class="WD_Help"><b>ScopeControl command keys</b><br>
10309 g ... grid on/off<br>
10310 i ... zoom in<br>
10311 o ... zoom out<br>
10312 s ... save current scope<br>
10313 r ... restore saved scope<br>
10314</span>
10315</menu>
10316<div class="WD_Panel" draggable="true">
10317<p><i-- should be on the frame of the WD --></i>
10318Monitor <input id="WD_Resize_1" class="WD_Button" type="button" value="Resize">
10319 < <input id="WD_Width_1" class="WD_Config" type="text">
10320 x <input id="WD_Height_1" class="WD_Config" type="text">
10321 wall-paper: <a href="WD-WallPaper03.png" class="WD_Href" contenteditable="true">WD-WallPaper03.png</a>
10322</p>
10323<p>
10324Desktop <input id="WS_Resize_1" class="WD_Button" type="button" value="Resize">
10325 < <input id="WS_1_Width" class="WD_Config" type="text">
10326 x <input id="WS_1_Height" class="WD_Config" type="text">
10327</p>
10328<p>
10329Display <input id="WD_Zoom_1" class="WD_Button" type="button" value="Zoom">
10330 < <input id="WD_Zoom_1_XY" class="WD_Config" type="text" value="1.0">
10331 x <input id="WD_Zoom_1_MAG" class="WD_Config" type="text" value="1.41421356">
10332 < <input id="WD_Zoom_1_OUT" class="WD_Button" type="button" value="ZoomOut">
10333 < <input id="WD_Zoom_1_IN" class="WD_Button" type="button" value="ZoomIn">
10334</p>
10335<p>
10336Content <input id="WD_Scroll_1" class="WD_Button" type="button" value="Scroll">
10337 X <input id="WD_Left_1" class="WD_Config" type="text" value="0">
10338 Y <input id="WD_Top_1" class="WD_Config" type="text" value="0">
10339shift+wheel for horizontal scroll
10340</p>
10341<p>
10342Scopexx <input id="WD_Zoom_1_Restore" class="WD_Button" type="button" value="Restore">
10343 < <input id="WD_Zoom_1_RestoreScope" class="WD_Config" type="text" value="Scope0">
10344 | <input id="WD_Zoom_1_Save" class="WD_Button" type="button" value="Save">
10345 < <input id="WD_Zoom_1_SaveScope" class="WD_Config" type="text" value="Scope0">
10346</p>
10347<p>
10348Scopeyy <input id="WD_Zoom_1_Forw" class="WD_Button" type="button" value="Forw">
10349 < <input id="WD_Zoom_1_ForwScope" class="WD_Config" type="text" value="Scope0">
10350 | <input id="WD_Zoom_1_Back" class="WD_Button" type="button" value="Back">
10351 > <input id="WD_Zoom_1_BackScope" class="WD_Config" type="text" value="Scope0">
10352</p>
10353<p>
10354Scopezz <input id="WD_Zoom_1_Push" class="WD_Button" type="button" value="Push">
10355 < <input id="WD_Zoom_1_PushScope" class="WD_Config" type="text" value="Scope0">
10356 | <input id="WD_Zoom_1_Pop" class="WD_Button" type="button" value="Pop">
10357 > <input id="WD_Zoom_1_PopScope" class="WD_Config" type="text" value="Scope0">
10358</p>
10359<p>
10360Display <input id="WD_Grid_1" class="WD_Button" type="button" value="GridOn">
10361</p>
10362<p>
10363Overflow <input id="WD_Overflow_1" class="WD_Button" type="button" value="scroll">
10364 "scroll" imprisons windows inside the display
10365</p>
10366</div>
10367
10368<div id="VirtualDesktop_1" class="VirtualDesktop" draggable="true" style="" contenteditable="true">
10369<div id="VirtualDesktop_1_MenuBar" class="VirtualDesktopMenuBar" spellcheck="false">
10370<i>CosmoScreen 0.0.8</i><span id="VirtualDesktop_1_Clock"></span>
10371</div>
10372<div id="VirtualDesktop_1_Calender" class="VirtualDesktopCalender" >00:00</div>
10373<div align="right"><h1>VirtualSpace 1.</h1></div>
10374<div id="VirtualDesktop_1_Content" class="VirtualSpace">
10375
10376<div id="VirtualBrowser_1" class="VirtualBrowserSpan" spellcheck="false" draggable="true">
10377<div id="VirtualBrowser_1_Location" class="VirtualBrowserLocationBar"></div>
10378<span id="VirtualBrowser_1_Command" class="VirtualBrowserCommandBar">Reload</span>
10379<iframe id="VirtualBrowser_1_Frame" class="VirtualBrowserFrame" style=""></iframe>
10380</div>
10381
10382<div id="VirtualBrowser_2" class="VirtualBrowserSpan" spellcheck="false" draggable="true">
10383<div id="VirtualBrowser_2_Location" class="VirtualBrowserLocationBar"></div>
10384<span id="VirtualBrowser_2_Command" class="VirtualBrowserCommandBar">Reload</span>
10385<iframe id="VirtualBrowser_2_Frame" class="VirtualBrowserFrame" style=""></iframe>
10386</div>
10387
10388<div id="VirtualBrowser_3" class="VirtualBrowserSpan" spellcheck="false" draggable="true">
10389<div id="VirtualBrowser_3_Location" class="VirtualBrowserLocationBar"></div>
10390<span id="VirtualBrowser_3_Command" class="VirtualBrowserCommandBar">Reload</span>
10391<iframe id="VirtualBrowser_3_Frame" class="VirtualBrowserFrame" style=""></iframe>
10392</div>
10393
10394<div id="VirtualDesktop_1_GridPlane" class="VirtualSpace"></div>
10395</div>
10396</div>
10397
10398<input class="HtmlCodeViewButton" type="button" value="ShowCode" onclick="vd_showHtmlCode()">
10399<span id="VirtualDesktopCodeview"></span>
10400<script id="VirtualDesktopScript">
10401function vd_showHtmlCode(){
10402  codespan = document.getElementById('VirtualDesktopCodeSpan');
10403  showHtmlCode(VirtualDesktopCodeview,codespan);
10404  VirtualDesktopCodeview.setAttribute('class','LiveHtmlCodeviewText');
10405}
10406DestroyEventSharingCodeview = function(){
10407  VirtualDesktopCodeview.innerHTML = "";
10408}
10409
10410function wdlog(log){
10411  if( GJ_Channel != null ){
10412    GJ_SendMessage('WD '+log);
10413  }
10414  console.log(log);
10415}
10416var topMostWin = 10000;

```

```

10417function onEnterWin(e){
10418    t = e.target;
10419    oindex = t.style.zIndex;
10420    //if( oindex == '' ) oindex = 0;
10421    //t.saved_zIndex = oindex;
10422    //t.style.zIndex = 10000;
10423    topMostWin += 1;
10424    t.style.zIndex = topMostWin;
10425    nindex = t.style.zIndex;
10426    wdlog('Enter '+t+' #' + t.id + '(' + oindex + '->' + nindex + ')');
10427    e.stopPropagation();
10428    e.preventDefault();
10429}
10430function onClickWin(e){ // can detect click on the thick border? t = e.target;
10431    oindex = t.style.zIndex;
10432    topMostWin += 1;
10433    t.style.zIndex = topMostWin;
10434    nindex = t.style.zIndex;
10435    wdlog('Click '+t+' #' + t.id + '(' + oindex + '->' + nindex + ')');
10436    //e.stopPropagation();
10437    //e.preventDefault();
10438}
10439function onLeaveWin(e){
10440    t = e.target;
10441    //oindex = t.style.zIndex;
10442    //nindex = t.saved_zIndex;
10443    //t.style.zIndex = nindex;
10444    //wdlog('Leave '+e.target+' #' + e.target.id + '(' + oindex + '->' + nindex + ')');
10445    e.stopPropagation();
10446    e.preventDefault();
10447}
10448
10449var WinDragstartX; // event
10450var WinDragstartY;
10451var WinDragstartTX; // target
10452var WinDragstartTY;
10453
10454function onWinDragstart(e){
10455    WinDragstartX = e.x;
10456    WinDragstartY = e.y;
10457
10458    t = e.target;
10459
10460    //WinDragstartTX = t.getBoundingClientRect().left.toFixed(0)
10461    //WinDragstartTY = t.getBoundingClientRect().top.toFixed(0)
10462    if( t.style.left == '' ){
10463        WinDragstartTX = x0 = 0;
10464        t.style.left = '0px';
10465    }else{
10466        //WinDragstartTX = x0 = Number(t.style.left);
10467        WinDragstartTX = x0 = parseInt(t.style.left);
10468    }
10469    if( t.style.top == '' ){
10470        WinDragstartTY = y0 = 0;
10471        t.style.top = '0px';
10472    }else{
10473        //WinDragstartTY = y0 = Number(t.style.top);
10474        WinDragstartTY = y0 = parseInt(t.style.top);
10475    }
10476    if( true ){ // to be undo
10477        t.wasAtX = WinDragstartTX;
10478        t.wasAtY = WinDragstartTY;
10479    }
10480    wdlog('DragSTA #' + t.id
10481        + ' event(' + e.x + ', ' + e.y + ')'
10482        + ' position=' + t.style.position
10483        + ' style left,top(' + t.style.left + ', ' + t.style.top + ')'
10484    );
10485    e.stopPropagation();
10486    //e.preventDefault();
10487    return true;
10488}
10489function onWinDragEvent(wh,e,set,dolog){
10490    t = e.target;
10491    dx = e.x - WinDragstartX;
10492    dy = e.y - WinDragstartY;
10493    nx = WinDragstartTX + dx;
10494    ny = WinDragstartTY + dy;
10495    log = 'Drag'+wh+' #' + t.id
10496        + ' event0(' + WinDragstartX + ', ' + WinDragstartY + ')'
10497        + ' event(' + e.x + ', ' + e.y + ')'
10498        + ' diff(' + dx + ', ' + dy + ')'
10499        + ' (' + nx + ', ' + ny + ')'
10500        + ' (' + t.style.left + ', ' + t.style.top + ')'
10501        + ' wasAt(' + t.wasAtX + ', ' + t.wasAtY + ')'
10502    ;
10503    if( e.x != 0 || e.y != 0 ){
10504        if( set == true ){
10505            //t.style.x = nx + 'px'; // not effective
10506            //t.style.y = ny + 'px'; // not effective
10507            t.style.left = nx + 'px';
10508            t.style.top = ny + 'px';
10509            log += ' Set';
10510        }else{
10511            log += ' NotSet';
10512            if( !dolog ){
10513                log = '';
10514            }
10515        }
10516    }else{
10517        log += ' What?'; // the type is event start?
10518        if( !dolog ){
10519            log = '';
10520        }
10521    }
10522    if( and(dolog, log != '' ) ){
10523        wdlog(log);
10524    }
10525    if( true ){
10526        // should be propargeted to parent in FireFox ?
10527        e.stopPropagation();
10528    }
10529    e.preventDefault();
10530    return false;
10531}
10532function onWinDrag(e){
10533    return onWinDragEvent('Ing',e,true,false);
10534}
10535function onWinDragend(e){
10536    return onWinDragEvent('End',e,false,true);
10537}
10538function onWinDragexit(e){
10539    return onWinDragEvent('Exit',e,false,true);
10540}

```

```

10541function onWinDragover(e){
10542    return onWinDragEvent('Over',e,false,true);
10543}
10544function onWinDragenter(e){
10545    return onWinDragEvent('Enter',e,false,true);
10546}
10547function onWinDragleave(e){
10548    return onWinDragEvent('Leave',e,false,true);
10549}
10550function onWinDragdrop(e){
10551    return onWinDragEvent('Drop',e,false,true);
10552}
10553function onFaviconChange(e){
10554    wdlog('--Favicon #' + e.target.id + ' href=' + e.details.href);
10555}
10556var savedSuppressGJShell = false;
10557function stopGShell(e){
10558    //wdlog('enter Gsh STOP');
10559    savedSuppressGJShell = SuppressGJShell;
10560    SuppressGJShell = true;
10561    e.stopPropagation();
10562    e.preventDefault();
10563}
10564function contGShell(e){
10565    //wdlog('leave Gsh STOP');
10566    SuppressGJShell = savedSuppressGJShell;
10567    e.stopPropagation();
10568    e.preventDefault();
10569}
10570
10571function WD_onKeydown(e){
10572    keycode = e.code;
10573    console.log('Keydown #' + e.target.id + ' ' + keycode);
10574    if( keycode == 'KeyG' ){
10575        WD_setGrid1(WD_Grid1);
10576    }else
10577    if( keycode == 'KeyI' ){
10578        WD_doZoomIN();
10579    }else
10580    if( keycode == 'KeyO' ){
10581        WD_doZoomOUT();
10582    }else
10583    if( keycode == 'KeyR' ){
10584        WD_RestoreScope(null);
10585    }else
10586    if( keycode == 'KeyS' ){
10587        WD_SaveScope(null);
10588    }
10589    e.stopPropagation();
10590    e.preventDefault();
10591}
10592function WD_onKeyUp(e){
10593    e.stopPropagation();
10594    e.preventDefault();
10595}
10596function WD_EventSetup1(){
10597    WirtualDesktop_1.addEventListener('keydown', e => { WD_onKeydown(e); });
10598    WirtualDesktop_1_Content.addEventListener('keydown', e => { WD_onKeydown(e); });
10599    WD_Help_1.addEventListener('keydown', e => { WD_onKeydown(e); });
10600    WD_Help_1.addEventListener('keyup', e => { WD_onKeyUp(e); });
10601}
10602function settleWin(s,l,cmd,f,u,w,h,x,y,c,scale){
10603    function WirtualBrowserCommand(e,s,l,cmd,f){
10604        command = cmd.innerHTML
10605        if( command == "Reload" ){
10606            href_id = e.target.href_id;
10607            d = document.getElementById(href_id);
10608            wdlog('href_tag=#'+href_id+'\n elem=#'+href_id+'\n href='+d);
10609            url = d.innerHTML;
10610            wdlog('---- Load href_tag=#'+href_id+'\n elem=#'+href_id+'\n href='+d
10611                + '\n url=' + url);
10612            wdlog('---- Load target #' + f.id + ' with url=' + url;
10613            f.src = url;
10614        }else{
10615            alert('unknown command' + command + ' "' + e.target.id + ', ' + l.id + ', ' + f.id);
10616        }
10617    }
10618    function onKeyDown(e){
10619        if( e.code == 'Enter' ){
10620            e.stopPropagation();
10621            e.preventDefault();
10622        }
10623    }
10624    function onKeyUp(e){
10625        if( e.code == 'Enter' ){
10626            e.stopPropagation();
10627            e.preventDefault();
10628            // should reload immediately ?
10629        }
10630    }
10631    if( false ){
10632        wdlog('start settle WirtualBrowser url=' + u + '\n'
10633            + 'id=' + s.id + '\n'
10634            + 'width=' + s.style.width + '\n'
10635            + 'height=' + s.style.height
10636        );
10637    }
10638    // very important for WordPress ??
10639    s.style.width = f.style.width = 501; // for WordPress ...??
10640    s.style.height = f.style.height = 271; // for WordPress ...??
10641    if( false ){
10642        wdlog('midway settle WirtualBrowser url=' + u + '\n'
10643            + 'id=' + s.id + '\n'
10644            + 'width=' + s.style.width + '\n'
10645            + 'height=' + s.style.height
10646        );
10647    }
10648    s.width = 502; // for WordPress ...??
10649    s.height = 272; // for WordPress ...??
10650    if( false ){
10651        wdlog('midway-2 settle WirtualBrowser url=' + u + '\n'
10652            + 'id=' + s.id + '\n'
10653            + 'span-width=' + s.width + '\n'
10654            + 'span-height=' + s.height
10655        );
10656    }
10657    s.style.width = w + 'px';
10658    s.style.height = h + 'px';
10659    f.style.width = w + 'px';
10660    f.style.height = h + 'px';
10661    //f.style.setProperty('-webkit-transform', 'scale('+scale+')');
10662    f.style.setProperty('transform', 'scale('+scale+')');

```

```

10665 //wlog('--x1-- u='+u+' width s='+s.style.width+'f='+f.style.width);
10666 //wlog('--x2-- u='+u+' width s='+s.style.width+'f='+f.style.width);
10667 s.setAttribute('draggable','true');
10668 f.setAttribute('draggable','false'); // why necessary?
10669 l.setAttribute('draggable','false'); // why necessary?
10670 cmd.setAttribute('draggable','false'); // why necessary?
10671 s.addEventListener('dragstart', e => { onWinDragstart(e); });
10672 s.addEventListener('drag', e => { onWinDrag(e); });
10673 s.addEventListener('exit', e => { onWinDragexit(e); });
10674 s.addEventListener('dragend', e => { onWinDragend(e); });
10675 s.addEventListener('dragexit', e => { onWinDragexit(e); });
10676 s.addEventListener('dragenter', e => { onWinDragenter(e); });
10677 s.addEventListener('dragover', e => { onWinDragover(e); });
10678 s.addEventListener('dragleave', e => { onWinDragleave(e); });
10679 s.addEventListener('drop', e => { onWinDragdrop(e); });
10680
10681 s.addEventListener('mouseenter',e => { onEnterWin(e); });
10682 s.addEventListener('mouseleave',e => { onLeaveWin(e); });
10683
10684 if( false ){
10685     s.style.position = "absolute";
10686     s.style.x = x+'px';
10687     s.style.left = x+'px';
10688     s.style.y = y+'px';
10689     s.style.top = y+'px';
10690 }else{
10691     s.style.setProperty('position','absolute','important');
10692     s.style.setProperty('x',x+'px','important');
10693     s.style.setProperty('left',x+'px','important');
10694     s.style.setProperty('y',y+'px','important');
10695     s.style.setProperty('top',y+'px','important');
10696 }
10697
10698 favicon = './favicon.ico';
10699 uv1 = u.split('/');
10700 if( 2 <= uv1.length ){
10701     uv2 = uv1[1].split('/');
10702     if( 2 <= uv2.length ){
10703         if( uv1[0] == 'file' ){
10704             //favicon = 'file://' + uv2.slice(0,uv2.length-1).join('/');
10705             // + './favicon.ico';
10706             favicon = './favicon.ico';
10707         }else{
10708             favicon = uv1[0] + '://' + uv2[0] + './favicon.ico';
10709         }
10710     }
10711 }
10712 //wlog("----- favicon-url="+favicon);
10713 href_id = l.id + '_href';
10714 l.innerHTML = ''
10715     + '<a id="'+href_id+'" class="WirtualBrowserLocation" href="'+u+'>'+u+'</a>';
10716 //l.addEventListener('click', e => { onClickWin(e); });
10717 l.addEventListener('mouseenter',e => { stopGShell(e); });
10718 l.addEventListener('mouseleave',e => { contGShell(e); });
10719 l.addEventListener('keydown', e => { onKeyDown(e); });
10720 l.addEventListener('keyup', e => { onKeyUp(e); });
10721
10722 cmd.href_id = href_id;
10723 wlog('(0)cmd=#'+cmd.id);
10724 wlog('(1)href_id=#'+href_id);
10725 wlog('(2)href_id=#'+cmd.href_id);
10726 cmd.addEventListener('click', e => { WirtualBrowserCommand(e,s,l,cmd,f); });
10727
10728 f.style.borderColor = c;
10729 f.src = u;
10730 //f.addEventListener('mouseenter',e => { onEnterWin(e); });
10731 //f.addEventListener('mouseleave',e => { onLeaveWin(e); });
10732
10733 //s.addEventListener('click', e => { onClickWin(e); });
10734 //f.addEventListener('click', e => { wlog('click wbl'); });
10735 f.addEventListener('mozbrowsericonchange',onFaviconChange);
10736
10737 wlog('done settle WirtualBrowser url='+u+'\n'
10738     + 'id=' + s.id + ' '
10739     + 'width=' + s.style.width + ' '
10740     + 'height=' + s.style.height + ' '
10741     + 'cmd=' + cmd.id
10742 );
10743
10744 }
10745
10746function WD_EventSetup2(){
10747     dt = WirtualDesktop_1;
10748     dt.style.width = "800px";
10749     dt.style.height = "500px";
10750     dt.addEventListener('dragstart',e => { onWinDragstart(e); });
10751     dt.addEventListener('drag', e => { onWinDrag(e); });
10752     dt.addEventListener('exit', e => { onWinDragexit(e); });
10753 }
10754
10755function GRonClick(){
10756     WD_SaveScope(null); // should be push
10757     t = event.target;
10758     x = t.getAttribute('data-leftx');
10759     y = t.getAttribute('data-topy');
10760     zoom = WD_Zoom_1_XY.value;
10761     x *= zoom;
10762     y *= zoom;
10763     WD_doScrollXY(event,x,y);
10764     //alert('scroll #' + t.id + ' x=' + x + ', y=' + y);
10765 }
10766function WD_setGrid(e){
10767     t = e.target;
10768     WD_setGrid1(t);
10769 }
10770function WD_setGrid1(t){
10771     //ds = WirtualDesktop_1_Content; // should be WirtualSpace_1
10772     ds = WirtualDesktop_1_GridPlane;
10773     if( t.value == 'GridOn' ){
10774         for( col = 0; col < 16; col++ ){
10775             for( row = 0; row < 16; row++ ){
10776                 g1 = document.createElement('span');
10777                 g1.setAttribute('class','WirtualGrid');
10778                 leftx = col * 800;
10779                 topy = row * 500;
10780                 gid = col + '.' + row
10781                 label = '<' + 'span '
10782                     + 'id="'+gid+'"' + 'class="WD_GridScroll" '
10783                     + 'contenteditable="false" onclick="GRonClick()" '
10784                     + 'data-leftx=' + leftx + ' ' + 'data-topy=' + topy + ' '
10785                     + '>'
10786                     + gid + '<'+ '/span>';
10787                 console.log('grid '+label);
10788                 g1.innerHTML = label;

```

```

10789         gl.position = 'relative';
10790         gl.leftx = leftx;
10791         gl.topy = topy;
10792         gl.style.left = gl.leftx + 'px';
10793         gl.style.top = gl.topy + 'px';
10794         if( col % 2 == row % 2 ){
10795             gl.style.backgroundColor = 'rgba(255,255,255,0.3)';
10796         }
10797         ds.appendChild(gl);
10798     }
10799 }
10800 t.value = 'GridOff';
10801 }else{
10802     ds.innerHTML = '';
10803     t.value = 'GridOn';
10804 }
10805 }
10806 }
10807 var sav_scrollLeft;
10808 var sav_scrollTop;
10809 var sav_nscale;
10810 function WD_SaveScope(e){
10811     sav_scrollLeft = WD_Left_1.value;
10812     sav_scrollTop = WD_Top_1.value;
10813     sav_nscale = WD_Zoom_1_XY.value;
10814     //console.log('saved zoom='+sav_oscale+','+sav_nscale);
10815 }
10816 function WD_RestoreScope(e){
10817     WD_Zoom_1_XY.value = sav_nscale;
10818     WD_doZoom();
10819 }
10820 WD_Left_1.value = sav_scrollLeft;
10821 WD_Top_1.value = sav_scrollTop;
10822 WD_doScroll(null);
10823 }
10824 function ignoreEvent(e){
10825     e.stopPropagation();
10826     //e.preventDefault();
10827 }
10828 function zoomMag(){
10829     return WD_Zoom_1_MAG.value;
10830 }
10831 function WD_EventSetup3(){
10832     WD_Grid_1.addEventListener('click', e => { WD_setGrid(e); });
10833     WD_Zoom_1_Save.addEventListener('click', e => { WD_SaveScope(e); });
10834     WD_Zoom_1_Restore.addEventListener('click', e => { WD_RestoreScope(e); });
10835     WD_Width_1.value = dt.style.width;
10836     WD_Width_1.addEventListener('keydown', ignoreEvent);
10837     WD_Width_1.addEventListener('keyup', ignoreEvent);
10838     WD_Height_1.value = dt.style.height;
10839     WD_Height_1.addEventListener('keydown', ignoreEvent);
10840     WD_Height_1.addEventListener('keyup', ignoreEvent);
10841     WD_Zoom_1_MAG.addEventListener('keydown', ignoreEvent);
10842     WD_Zoom_1_MAG.addEventListener('keyup', ignoreEvent);
10843 }
10844 }
10845 function escale1(e,oscale,nscale){
10846     e.style.setProperty('transform','scale('+nscale+')');
10847     rscale = oscale / nscale;
10848     w = parseInt(e.style.width);
10849     h = parseInt(e.style.height);
10850     w = w * rscale; //(oscale/nscale);
10851     h = h * rscale; //(oscale/nscale);
10852     e.style.width = w + 'px';
10853     e.style.height = h + 'px';
10854 }
10855 function scaleWD(ds,oscale,nscale){
10856     if( true ){
10857         escale1(WirtualBrowser_1,oscale,nscale);
10858         escale1(WirtualBrowser_1_Location,oscale,nscale);
10859         escale1(WirtualBrowser_1_Command,oscale,nscale);
10860         escale1(WirtualBrowser_1_Frame,oscale,nscale);
10861     }
10862     escale1(WirtualBrowser_2,oscale,nscale);
10863     escale1(WirtualBrowser_2_Location,oscale,nscale);
10864     escale1(WirtualBrowser_2_Command,oscale,nscale);
10865     escale1(WirtualBrowser_2_Frame,oscale,nscale);
10866 }
10867     escale1(WirtualBrowser_3,oscale,nscale);
10868     escale1(WirtualBrowser_3_Location,oscale,nscale);
10869     escale1(WirtualBrowser_3_Command,oscale,nscale);
10870     escale1(WirtualBrowser_3_Frame,oscale,nscale);
10871 }
10872 }
10873 function WD_doZoom(){
10874     ds = WirtualDesktop_1_Content; // should be WirtualSpace_1
10875     oscale = WD_Zoom_1_XY.ovalue; // hidden value for current zoom
10876     nscale = WD_Zoom_1_XY.value;
10877     ds.style.zoom = nscale;
10878     WD_Zoom_1_XY.value = ds.style.zoom;
10879     WD_Zoom_1_XY.ovalue = ds.style.zoom;
10880     scaleWD(ds,oscale,nscale);
10881 }
10882 function WD_EventSetup4(){
10883     WD_Zoom_1.addEventListener('click',WD_doZoom);
10884     WD_Zoom_1_XY.addEventListener('keydown', ignoreEvent);
10885     WD_Zoom_1_XY.addEventListener('keyup', ignoreEvent);
10886 }
10887 }
10888 function WD_doZoomOUT(){
10889     ds = WirtualDesktop_1_Content; // should be WirtualSpace_1
10890     oscale = WD_Zoom_1_XY.value;
10891     if( oscale == 0 || oscale == '' ){
10892         oscale = 1;
10893     }
10894     nscale = oscale / zoomMag();
10895     ds.style.zoom = nscale;
10896     WD_Zoom_1_XY.value = ds.style.zoom;
10897     WD_Zoom_1_XY.ovalue = ds.style.zoom;
10898     scaleWD(ds,oscale,nscale);
10899 }
10900 function WD_doZoomIN(){
10901     ds = WirtualDesktop_1_Content; // should be WirtualSpace_1
10902     oscale = WD_Zoom_1_XY.value;
10903     if( oscale == 0 || oscale == '' ){
10904         oscale = 1;
10905     }
10906     nscale = oscale * zoomMag();
10907     if( 4 < nscale ){
10908         alert('maybe too large, zoom='+nscale);
10909         return;
10910     }
10911     ds.style.zoom = nscale;
10912     WD_Zoom_1_XY.value = ds.style.zoom;

```



```

10913   WD_Zoom_1_XY.ovalue = ds.style.zoom;
10914   scaleWD(ds,oscale,nscale);
10915 }
10916 function WD_EventSetup5(){
10917   WD_Zoom_1_OUT.addEventListener('click',WD_doZoomOUT);
10918   WD_Zoom_1_IN.addEventListener('click',WD_doZoomIN);
10919 }
10920
10921 function WD_doResize(e){
10922   dt = WirtualDesktop_1;
10923   dt.style.width = WD_Width_1.value;
10924   dt.style.height = WD_Height_1.value;
10925   WD_Width_1.value = dt.style.width;
10926   WD_Height_1.value = dt.style.height;
10927 }
10928 WD_Resize_1.addEventListener('click',e => { WD_doResize(e); });
10929
10930
10931 function WD_dorSResize(e){
10932   //alert('Resize Space');
10933   ds = WirtualDesktop_1_Content; // should be WirtualSpace_1
10934   ds.style.width = WS_1_Width.value;
10935   ds.style.height = WS_1_Height.value;
10936   WS_1_Width.value = ds.style.width;
10937   WS_1_Height.value = ds.style.height;
10938 }
10939 function WD_EventSetup6(){
10940   ds = WirtualDesktop_1_Content; // should be WirtualSpace_1
10941   ds.style.width = '5120px';
10942   ds.style.height = '2880px';
10943   WS_1_Width.value = ds.style.width;
10944   WS_1_Height.value = ds.style.height;
10945   WS_1_Width.addEventListener('keydown',ignoreEvent);
10946   WS_1_Width.addEventListener('keyup',ignoreEvent);
10947   WS_1_Height.addEventListener('keydown',ignoreEvent);
10948   WS_1_Height.addEventListener('keyup',ignoreEvent);
10949   WS_Resize_1.addEventListener('click',e => { WD_dorSResize(e); });
10950 }
10951
10952 function WD_doScrollXY(e,sleft,stop){
10953   dt = WirtualDesktop_1;
10954   dt.scrollLeft = sleft;
10955   dt.scrollTop = stop;
10956   WD_Left_1.value = dt.scrollLeft;
10957   WD_Top_1.value = dt.scrollTop;
10958   console.log('--Scroll #' + dt.id + ' (' + sleft + ', ' + stop + ')');
10959 }
10960 function WD_doScroll(e){
10961   //dt = WirtualDesktop_1_Content;
10962   dt = WirtualDesktop_1;
10963   sleft = parseInt(WD_Left_1.value);
10964   stop = parseInt(WD_Top_1.value);
10965   dt.scrollLeft = sleft;
10966   dt.scrollTop = stop;
10967   WD_Left_1.value = dt.scrollLeft;
10968   WD_Top_1.value = dt.scrollTop;
10969   console.log('--Scroll #' + dt.id + ' (' + sleft + ', ' + stop + ')');
10970 }
10971 function showScrollPosition(){
10972   if( false )
10973     console.log(
10974       'wstop=' + WirtualDesktop_1.style.top + ', ' +
10975       'wsx=' + WirtualDesktop_1.style.y + ', ' +
10976       'wss=' + WirtualDesktop_1.scrollTop + ', ' +
10977       'wdtop=' + WirtualDesktop_1_Content.style.top + ', ' +
10978       'wdx=' + WirtualDesktop_1_Content.style.y + ', ' +
10979       'wds=' + WirtualDesktop_1_Content.scrollTop + ', '
10980     );
10981   WD_Left_1.value = WirtualDesktop_1.scrollLeft;
10982   WD_Top_1.value = WirtualDesktop_1.scrollTop;
10983 }
10984 function WD_EventSetup7(){
10985   WD_Scroll_1.addEventListener('click',e => { WD_doScroll(e); });
10986   WD_Left_1.addEventListener('keydown',ignoreEvent);
10987   WD_Left_1.addEventListener('keyup',ignoreEvent);
10988   WD_Top_1.addEventListener('keydown',ignoreEvent);
10989   WD_Top_1.addEventListener('keyup',ignoreEvent);
10990 }
10991 function WD_EventSetup8(){
10992   WirtualDesktop_1.addEventListener('scroll',showScrollPosition);
10993   WirtualDesktop_1_Content.addEventListener('scroll',showScrollPosition);
10994 }
10995
10996 if( false ){
10997   w = 1000 + 'px';
10998   dt.style.width = w;
10999   dt.style.height = "300px";
11000   dt.style.resize = "both";
11001   dt.style.borderWidth = 50 + 'px';
11002   dt.style.borderRadius = 25 + 'px';
11003   console.log('----- #' + dt.id + ' style=' + dt.style);
11004   console.log('----- #' + dt.id + ' width=' + dt.style.width);
11005   console.log('----- #' + dt.id + ' left=' + dt.style.left);
11006   console.log('----- #' + dt.id + ' border=' + dt.style.border);
11007 }
11008 function onDTResize(e){
11009   dt = e.target;
11010   h = parseInt(dt.style.height);
11011   dt.style.borderWidth = (h * 0.075) + 'px';
11012   console.log('----- borderWidgh=' + dt.style.borderWidth);
11013 }
11014
11015 WirtualDesktopDetails.addEventListener('toggle',WirtualDesktop_init);
11016 function WirtualDesktop_init(){
11017   if( !WirtualDesktopDetails.open ){
11018     return;
11019   }
11020   //GJ_Join();
11021   WirtualDesktop_1.addEventListener('resize', e => { onDTResize(e); });
11022   //console.log('----- #' + dt.id
11023   // + ' borderWidth=' + dt.style.getProperty('border-width'));
11024
11025   settleWin(
11026     WirtualBrowser_1,
11027     WirtualBrowser_1_Location,
11028     WirtualBrowser_1_Command,
11029     WirtualBrowser_1_Frame,
11030     document.URL,
11031     500,280,50,20,'#262',1.0);
11032   settleWin(
11033     WirtualBrowser_2,
11034     WirtualBrowser_2_Location,
11035     WirtualBrowser_2_Command,
11036     WirtualBrowser_2_Frame,

```

```

11037     'https://its-more.jp/ja_jp/',
11038     500,280,150,100,'#448',1.0);
11039     settleWin(
11040         WirtualBrowser_3,
11041         WirtualBrowser_3_Location,
11042         WirtualBrowser_3_Command,
11043         WirtualBrowser_3_Frame,
11044         //'../gshell/gsh.go.html',
11045         //'http://gshell.org/gshell/gsh.go.html',
11046         'https://golang.org',
11047         500,280,250,180,'#444',1.0);
11048         //1000,720,0,0,'#444',0.125);
11049         //1200,720,-100,-50,'#444',0.4);
11050     function WD_ClockUpdate(e){
11051         WirtualDesktop_1_Clock.innerHTML = DateShort();
11052         WirtualDesktop_1_Calender.innerHTML = DateHourMin();
11053     }
11054     window.setInterval(WD_ClockUpdate,500);
11055
11056     WD_EventSetup1();
11057     WD_EventSetup2();
11058     WD_EventSetup3();
11059     WD_EventSetup4();
11060     WD_EventSetup5();
11061     WD_EventSetup6();
11062     WD_EventSetup7();
11063     WD_EventSetup8();
11064 }
11065 //WirtualDesktop_init();
11066
11067 Destroy_WirtualDesktop = function(){
11068     WirtualDesktop_1.style = "";
11069
11070     WirtualBrowser_1.removeAttribute('style');
11071     WirtualBrowser_1_Location.innerHTML = '';
11072     WirtualBrowser_1_Frame.removeAttribute('src');
11073     WirtualBrowser_1_Frame.removeAttribute('style');
11074     WirtualBrowser_1_Frame.style="";
11075
11076     WirtualBrowser_2.removeAttribute('style');
11077     WirtualBrowser_2_Location.innerHTML = '';
11078     WirtualBrowser_2_Frame.removeAttribute('src');
11079     WirtualBrowser_2_Frame.style="";
11080
11081     WirtualBrowser_3.removeAttribute('style');
11082     WirtualBrowser_3_Location.innerHTML = '';
11083     WirtualBrowser_3_Frame.removeAttribute('src');
11084     WirtualBrowser_3_Frame.style="";
11085
11086     GJFactory_1.style = "";
11087     iframe0.style = "";
11088     WirtualDesktop_1.style = "";
11089 }
11090
11091 </script>
11092 <!-- WirtualDesktop -->
11093 </details>
11094 *//</span>
11095
11096 <!-- ===== Work { ===== -->
11097 </span id="SBSidebar_WorkCodeSpan">
11098 /*
11099 <details><summary>SBSidebar</summary>
11100 <!-- ----- SBSidebar // 2020-0928 SatoxITS { -->
11101 <h2>SBSidebar</h2>
11102 <input id="SBSidebar_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
11103 <input id="SBSidebar_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
11104 <input id="SBSidebar_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
11105 <span id="SBSidebar_WorkCodeView"></span>
11106 <script id="SBSidebar_WorkScript">
11107 function SBSidebar_openWorkCodeView(){
11108     function SBSidebar_showWorkCode(){
11109         showHtmlCode(SBSidebar_WorkCodeView,SBSidebar_WorkCodeSpan);
11110     }
11111     SBSidebar_WorkCodeViewOpen.addEventListener('click',SBSidebar_showWorkCode);
11112 }
11113 SBSidebar_openWorkCodeView(); // should be invoked by an event
11114
11115 console.log('-- Sbslider // 2020-1006-01 SatoxITS --');
11116 function SetSidebar(){
11117     sidebar = document.getElementById('secondary');
11118     console.log('primary='+primary+' + 'secondary'+sidebar+' +main'+main+' ');
11119     wrap = sidebar.parentNode;
11120     console.log('-- Sbslider parent is '+wrap+', #' +wrap.id+' .' +wrap.class);
11121     //wrap = wrap.parentNode;
11122     //console.log('-- Sbslider parent is '+wwrap+', #' +wwrap.id+' .' +wwrap.class);
11123     //wwrap = wwrap.parentNode;
11124     //console.log('-- Sbslider parent is '+wwwrap+', #' +wwwrap.id+' .' +wwwrap.class);
11125     nsb = sidebar.cloneNode();
11126     nsb = sidebar;
11127     nsb.style.width = '100%';
11128     slider = document.createElement('div');
11129     slider.id = 'SbSlider';
11130     slider.appendChild(nsb);
11131     slider.setAttribute('class','SbSlider');
11132     nsb.style.position = 'relative';
11133     slider.style.position = 'fixed';
11134     slider.style.display = 'block'; //inline;
11135     slider.style.zIndex = 100000;
11136     // nsb.style.zIndex = 200000;
11137     nsb.style.position = 'absolute';
11138     nsb.style.minWidth = '80px';
11139     nsb.style.left = '0px';
11140     nsb.style.top = '0px';
11141
11142     w = window.innerWidth;
11143     console.log('sliderWidth '+w+': ',(w/3)+'px');
11144     if( w < 640 ){
11145         slider.style.setProperty('width',(w/3) + 'px','important');
11146     }
11147     main.style.marginLeft = (parseInt(slider.style.width)/2 - 20) + 'px';
11148
11149     slider.style.resize = "both";
11150     slider.draggable = "true";
11151     wrap.appendChild(slider);
11152     console.log('-- added Sbslider');
11153     //nsb.addEventListener('scroll',SbScrolled);
11154
11155     buttons = document.createElement('div');
11156     buttons.id = 'NaviButtons';
11157     buttons.setAttribute('class','NaviButtons');
11158     buttons.align = "center";
11159     buttons.innerHTML += '<+>p><a href="#TopOfPost" draggable="true">TOP<+>/a<+>/p>';
11160     buttons.innerHTML += '<+>p><a href="#EndOfPost" draggable="true">END<+>/p>';

```

```

11161     page.appendChild(buttons);
11162     buttons.style.position = 'fixed';
11163     buttons.style.zIndex = 30000;
11164     buttons.style.width = '180%';
11165     buttons.style.top = '320px';
11166     buttons.style.left = parseInt(w) * 1.0 + 'px';
11167     console.log('-- Sbslider installed (^-^)/ SatoxITS');
11168 }
11169 //window.addEventListener('load',SetSidebar); // after load
11170 DestroyNaviButtons = function(){
11171     nb = document.getElementById('NaviButtons');
11172     if( nb != null ){
11173         nb.parentNode.removeChild(NaviButtons);
11174     }
11175 }
11176 </script>
11177
11178 // 2020-1006 its-more.jp-blog-60000-style.css
11179 <!-- {
11180 <style>
11181 #NaviButtons {
11182     position:fixed;
11183     display:block;
11184     xwidth:100%;
11185     xxtop:100px;
11186     xxleft:10px;
11187     z-index:30000;
11188     font-size:10pt;
11189     color:#2ff !important;
11190     text-align:center;
11191     background-color:rgba(230,230,230,0.01);
11192 }
11193 #NaviButtons a {
11194     color:#2a2 !important;
11195     font-size:20px;
11196     text-align:center;
11197     xtext-shadow:2px 2px #0ff;
11198     resize:both;
11199     padding:6px;
11200     margin:10px;
11201     border:1px solid #288 !important;
11202     border-radius:3px;
11203     background-color:rgba(160,160,160,0.05);
11204 }
11205 #Sbslider {
11206     overflow:auto;
11207     resize:both !important;
11208     xxoverflow-y:hidden !important;
11209     height:100px !important;
11210     display:inline !important;
11211     position:fixed !important;
11212     left:0px;
11213     top:0px;
11214     xxwidth:180px;
11215     width:24%;
11216     min-width:80px;
11217     height:100% !important;
11218     background-color:rgba(100,100,200,0.1);
11219 }
11220 #secondary {
11221     position:fixed;
11222     left:0px;
11223     top:0px;
11224     xxxz-index:60000;
11225     scroll-behavior: overflow !important;
11226     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxwidth:18% !important;
11227     padding-left:4pt;
11228     color:#fff;
11229     font-size:0.5em;
11230     background-color:rgba(64,160,64,0.6) !important;
11231     white-space:nowrap;
11232 }
11233 #secondary a {
11234     color:#fff !important;
11235     text-decoration:disable !important;
11236 }
11237 #primary {
11238     position:relative;
11239     width:75% !important;
11240     left:25% !important;
11241 }
11242 #main {
11243     position:relative;
11244     width:75% !important;
11245     left:25% !important;
11246 }
11247 #site-navigation {
11248     position:relative;
11249     left:120px;
11250 }
11251 #adswsc_countertext {
11252     color:#4169e1;
11253     font-size:16pt !important;
11254     xxfont-size:10% !important;
11255     font-weight:bold;
11256 }
11257 #nowTime {
11258     color:#0ffa0;
11259     font-size:16pt !important;
11260     xxfont-size:10% !important;
11261     font-weight:bold;
11262     text-shadow:1px 1px #fff;
11263 }
11264 .navigation-top {
11265     color:#22a !important;
11266     border:0px;
11267     background-color:rgba(220,220,220,0.1);
11268 }
11269
11270 .visible-scrollbar, .invisible-scrollbar, .mostly-customized-scrollbar {
11271     display: block;
11272     xxwidth: 1em;
11273     xxoverflow: auto;
11274     xxheight: 1em;
11275 }
11276 .invisible-scrollbar::-webkit-scrollbar {
11277     xdisplay: none;
11278     width:1px !important;
11279     height:1px !important;
11280 }
11281 .mostly-customized-scrollbar::-webkit-scrollbar {
11282     width: 2px;
11283     height: 2px;
11284     xxbackground-color: #aaa; xxx:or add it to the track;

```

```

11285}
11286.mostly-customized-scrollbar::-webkit-scrollbar-thumb {
11287    background: #000;
11288}
11289</style>
11290<!-->
11291
11292
11293</details>
11294<!-- SBSidebar_WorkCodeSpan } -->
11295*//</span>
11296//<!-- ===== Work } ===== -->
11297
11298
11299//<!-- ===== Work { ===== -->
11300//<span id="Affiliate_WorkCodeSpan">
11301/*
11302<div id="AffViewDock">
11303<div id="AffView" class="AffView" draggable="true" style="">
11304<div id="AffISet" class="AffPlate">
11305<iframe id="aff_0" class="AffItem"></iframe>
11306<iframe id="aff_1" class="AffItem"></iframe>
11307<iframe id="aff_2" class="AffItem"></iframe>
11308<iframe id="aff_3" class="AffItem"></iframe>
11309<iframe id="aff_4" class="AffItem"></iframe>
11310<iframe id="aff_5" class="AffItem"></iframe>
11311</div>
11312</div></div>
11313<details id="Affiliate_Test"><summary>Affiliate</summary>
11314<!-- ----- Affiliate // 2020-1010 SatoxITS { -->
11315<h2>Supportive Affiliate</h2>
11316<style>
11317.AffView {
11318    z-index:0;
11319    overflow-x:scroll;
11320    overflow-y:scroll;
11321    position:fixed;
11322    max-width:2560px;
11323    max-height:100%;
11324    width:270px;
11325    left:75%;
11326    height:95%;
11327    resize:both;
11328    xleft:-10%;
11329    margin-top:40px;
11330    xleft:0;
11331    xxalign:right;
11332    display:block;
11333    border:4px inset rgba(255,255,255,0.1);
11334    background-color:rgba(255,255,255.0.1);
11335}
11336.AffView:hover {
11337    z-index:1;
11338    width:300px;
11339    overflow:scroll;
11340    border:4px inset #ffc;
11341    background-color:rgba(80,80,255.0.2);
11342    background-color:#ffc;
11343}
11344.AffPlate:hover {
11345    border-left:4px dashed #888;
11346}
11347.AffPlate{
11348    overflow-x:visible;
11349    border-left:4px dashed rgba(255,255,255,0.1);
11350    max-width:2560px;
11351    max-height:2880px;
11352    margin-top:10px;
11353    margin-bottom:10px;
11354    margin-left:4px;
11355    width:300px;
11356    xheight:1440px;
11357}
11358.AffItem {
11359    overflow-x:visible;
11360    xoverflow-y:scroll;
11361    max-width:2560px;
11362    max-height:1440px;
11363    z-index:0;
11364    display:block;
11365    xposition:fixed;
11366    xposition:absolute;
11367    position:relative;
11368    //left:300px;
11369    xresize:both;
11370    padding:0px;
11371    width:600px;
11372    height:400px;
11373    max-height:800px !important;
11374    margin-top:0%;
11375    margin-left:0%;
11376    margin-right:0% !important;
11377    border:16px inset #ccc;
11378    transform:scale(0.5);
11379    background-color:rgba(255,255,255,0.2);
11380    xxalign:right;
11381}
11382.AffItem:hover {
11383    border:16px inset #bbf;
11384}
11385</style>
11386<input id="Affiliate_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
11387<input id="Affiliate_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
11388<input id="Affiliate_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
11389<span id="Affiliate_WorkCodeView"></span>
11390<script id="Affiliate_WorkScript">
11391function Affiliate_openWorkCodeView(){
11392    function Affiliate_showWorkCode(){
11393        showHtmlCode(Affiliate_WorkCodeView,Affiliate_WorkCodeSpan);
11394    }
11395    Affiliate_WorkCodeViewOpen.addEventListener('click',Affiliate_showWorkCode);
11396}
11397Affiliate_openWorkCodeView(); // should be invoked by an event
11398
11399//<iframe id="aff_8" xxsrc="https://stackoverflow.com/tags" class="AffItem"></iframe>
11400//<iframe id="aff_9" xxsrc="https://developer.mozilla.org/en-US/docs/Web" class="AffItem"></iframe>
11401var Aff_isSetup = false;
11402Affiliate_Test.addEventListener('click',Aff_Setup);
11403function Aff_Setup(){
11404    if( Aff_isSetup ){ return; } Aff_isSetup = true;
11405    parent = document.documentElement;
11406    parent.appendChild(AffView);
11407    AffView.style.top = '0px';
11408    AffView.style.left = (window.innerWidth - 280) + 'px';

```

```

11409
11410 var off = 100;
11411 zoom = 0.5;
11412 ozoom = 0.3;
11413 leftx = window.innerWidth - 300;
11414 left = leftx + 'px';
11415 left = -130 + 'px';
11416 console.log('aff-init window.innerWidth='+window.innerWidth);
11417 w = 1000;
11418 h = 560;
11419
11420 aff_0.src = "../gshell/gsh.go.html";
11421 aff_1.src = "https://golang.org";
11422 aff_2.src = "https://drafts.csswg.org/";
11423 aff_3.src = "https://html.spec.whatwg.org/dev/";
11424 aff_4.src = "https://wikipedia.org";
11425 aff_5.src = "https://www.bing.com/translator";
11426
11427 //parent.appendChild(aff_0);
11428 aff_0.style.width = zoom*w+'px';
11429 aff_0.style.height = zoom*h+'px';
11430 aff_0.style.left = left;
11431 //aff_0.style.top = off+'px'; off += ozoom*h;
11432 aff_0.draggable = 'true';
11433
11434 //parent.appendChild(aff_1);
11435 aff_1.style.width = zoom*w+'px';
11436 aff_1.style.height = zoom*h+'px';
11437 aff_1.style.left = left;
11438 //aff_1.style.top = off+'px'; off += ozoom*h;
11439 aff_1.style.top = '-150px';
11440 aff_1.draggable = 'true';
11441
11442 //parent.appendChild(aff_2);
11443 aff_2.style.width = zoom*w+'px';
11444 aff_2.style.height = zoom*h+'px';
11445 aff_2.style.left = left;
11446 //aff_2.style.top = off+'px'; off += ozoom*h;
11447 aff_2.style.top = '-300px';
11448 aff_2.draggable = 'true';
11449
11450 //parent.appendChild(aff_3);
11451 aff_3.style.transform = 'scale(0.25)';
11452 aff_3.style.width = 2*zoom*w+'px';
11453 aff_3.style.height = 2*zoom*h+'px';
11454 aff_3.style.border = '32px inset #ccc';
11455 //aff_3.style.left = -390 + 'px'; //left*2;
11456 //aff_3.style.left = (leftx - 265) + 'px';
11457 aff_3.style.left = -395 + 'px';
11458 //aff_3.style.top = (-155+off)+'px'; off += ozoom*h;
11459 aff_3.style.top = '-600px';
11460 aff_3.draggable = 'true';
11461
11462 //parent.appendChild(aff_4);
11463 aff_4.style.width = zoom*w+'px';
11464 aff_4.style.height = zoom*h+'px';
11465 aff_4.style.left = left;
11466 //aff_4.style.top = off+'px'; off += ozoom*h;
11467 aff_4.style.top = '-900px';
11468 aff_4.draggable = 'true';
11469
11470 //parent.appendChild(aff_5);
11471 aff_5.style.transform = 'scale(0.300)';
11472 aff_5.style.width = zoom*(w*1.67)+'px';
11473 aff_5.style.height = zoom*(h*1.67)+'px';
11474 aff_5.style.border = '25px inset #ccc';
11475 aff_5.style.left = -308+'px';
11476 //aff_5.style.left = (-175+leftx)+'px';
11477 //aff_5.style.top = (-95+off)+'px'; off += ozoom*h;
11478 //aff_5.style.left = '0px';
11479 //aff_5.style.top = '0px';
11480 aff_5.style.top = '-1150px';
11481 //aff_5.style.align = 'right';
11482 aff_5.draggable = 'true';
11483
11484 function affresize(){
11485     AffView.style.left = (window.innerWidth - 280) + 'px';
11486     leftx = window.innerWidth - 400;
11487     left = leftx + 'px';
11488     console.log('aff-resize window.innerWidth='+window.innerWidth);
11489 }
11490 window.addEventListener('resize', affresize);
11491 //document.addEventListener('resize', affresize);
11492 //gsh.addEventListener('resize', affresize);
11493
11494 function ResetAffView(){
11495     AffViewDock.appendChild(AffView);
11496     AffView.removeAttribute('style');
11497     aff_0.removeAttribute('src');
11498     aff_0.removeAttribute('style'); aff_0.removeAttribute('draggable');
11499     aff_1.removeAttribute('src');
11500     aff_1.removeAttribute('style'); aff_1.removeAttribute('draggable');
11501     aff_2.removeAttribute('src');
11502     aff_2.removeAttribute('style'); aff_2.removeAttribute('draggable');
11503     aff_3.removeAttribute('src');
11504     aff_3.removeAttribute('style'); aff_3.removeAttribute('draggable');
11505     aff_4.removeAttribute('src');
11506     aff_4.removeAttribute('style'); aff_4.removeAttribute('draggable');
11507     aff_5.removeAttribute('src');
11508     aff_5.removeAttribute('style'); aff_5.removeAttribute('draggable');
11509 }
11510 </script>
11511 </details>
11512 <!-- Affiliate_WorkCodeSpan } -->
11513 *//</span>
11514 //<!-- ===== Work } ===== -->
11515
11516
11517
11518 //<!-- ===== Work { ===== -->
11519 //<span id="FontSelector_WorkCodeSpan">
11520 /*
11521 <details><summary>Font Selector</summary>
11522 <!-- ----- FontSelector // 2020-1013 SatoxITS { -->
11523 <h2>Font Selection</h2>
11524 <div>
11525 <div id="FontList"></div>
11526 <div>
11527 <style>
11528 #FontList {
11529     overflow:visible;
11530     background-color:rgba(240,245,255,1.0) !important;
11531 }
11532 #FontList td {

```

```

11533     font-size:20px;
11534     padding:0px;
11535     padding-left:2px;
11536     padding-right:2px;
11537     margin:0px;
11538     line-height:1.2;
11539     border:0px;
11540 }
11541 #FontList tr:hover {
11542     color:#fff;
11543     background-color:#000;
11544     xborder:1px solid #000;
11545 }
11546 .xcourier { colr:#000; font-size:16px; font-family:courier; }
11547 .xcursive { colr:#000; font-size:16px; font-family:cursive; }
11548 .xfantasy { colr:#000; font-size:16px; font-family:fantasy; }
11549 .xgeorgia { colr:#000; font-size:16px; font-family:georgia; }
11550 .xmonospace { colr:#000; font-size:16px; font-family:monospace; }
11551 </style>
11552 <script>
11553 function fontstr(name,text){
11554     //tr = '<'+'tr style='font-family:'+'name+'>\n';
11555     tr = '<'+'tr style='font-family:'+'name+'>\n';
11556     tr += '<'+'td style="font-family:Arial;font-size:12pt;">'+name+'<'+'>';
11557     tr += '<'+'td>'+text+'<'+'>';
11558     tr += '<'+'td><'+'b>'+text+'<'+'>';
11559     tr += '<'+'td><'+'i>'+text+'<'+'>';
11560     tr += '<'+'td><'+'b><'+'i>'+text+'<'+'>';
11561     tr += '<'+'tr>';
11562     return tr;
11563 }
11564 function lsfont(){
11565     text = 'GShell-Go012';
11566
11567     fl = '';
11568     fl += '<table>\n';
11569     fl += fontstr('Arial',text);
11570     fl += fontstr('Courier',text);
11571     fl += fontstr('Courier New',text);
11572     fl += fontstr('Georgia',text);
11573     fl += fontstr('Helvetica',text);
11574     fl += fontstr('Verdana',text);
11575     fl += fontstr('Times',text);
11576
11577     fl += fontstr('Osaka',text);
11578     fl += fontstr('Meiryo',text);
11579     fl += fontstr('YuMincho',text);
11580
11581     //fl += fontstr('Roman',text);
11582     //document.fonts.load("30px cursive");
11583     fl += fontstr('Serif',text);
11584     fl += fontstr('Sans-Serif',text);
11585     fl += fontstr('System-UI',text);
11586     fl += fontstr('Monospace',text);
11587     fl += fontstr('Cursive',text);
11588     fl += fontstr('Fantasy',text);
11589     fl += '</table>\n';
11590
11591     if( false ){
11592         fss = document.fonts.entries(); // FontFaceSet
11593         console.log('FSS='+fss);
11594         while( true ){
11595             font = fss.next();
11596             if( font.done ){
11597                 break;
11598             }
11599             fl += font.value[0] + '<br>';
11600         }
11601     }
11602     FontList.innerHTML = fl;
11603 }
11604 lsfont();
11605 document.fonts.onloadingdone = function(fsse){
11606     //alert('font-loaded '+fsse.fontfaces.length);
11607 }
11608 </script>
11609
11610
11611 <h2>Drawing Text on Canvas</h2>
11612 <!-- 2020-1012 -- Drawing Text on Canvas // SatoxITS ( -->
11613 <div id="TextCanvas_1_Panel" class="TextCanvasPanel">
11614 <input id="TextCanvas_1_Clear" class="PanelButton" type="button" value="Clear">
11615 <input id="TextCanvas_1_Draw" class="PanelButton" type="button" value="Draw">
11616 <input id="TextCanvas_1_Font" class="CanvasPanel" type="text" size="14" value="Georgia">
11617 <input id="TextCanvas_1_Size" class="CanvasPanel" type="text" size="3" value="64">Pixels
11618 <input id="TextCanvas_1_Bold" class="CanvasBox" type="checkbox" checked="">Bold
11619 <input id="TextCanvas_1_Italic" class="CanvasBox" type="checkbox" checked="">Italic
11620 <p>
11621 <input id="TextCanvas_1_Text" type="text" size="50" value="GShell">
11622 </p>
11623 </div>
11624 <p>
11625 <canvas id="TextCanvas_1" class="TextCanvas" width="400px" height="100px" draggable="true"></canvas>
11626 </p>
11627 <style>
11628 .CommandUsageText {
11629     font-family:Courier New;
11630 }
11631 .TextCanvas {
11632     border:1px solid #000;
11633     resize:both;
11634     display:inline !important;
11635 }
11636 .CanvasBox {
11637     vertical-align:middle;
11638     margin-left:4px !important;
11639     margin-right:2px !important;
11640 }
11641 .CanvasPanel {
11642     vertical-align:middle !important;
11643     height:14pt !important;
11644     width:inherit !important;
11645     padding:2px !important;
11646     margin:4px !important;
11647     margin-left:4px !important;
11648     margin-right:2px !important;
11649     font-size:10pt !important;
11650     font-family:Georgia !important;
11651     color:#000;
11652     display:inline !important;
11653 }
11654 .TextCanvas1Panel {
11655     vertical-align:middle;
11656     font-size:10pt !important;

```

```

11657 font-family:Georgia !important;
11658 color:#000;
11659 display:inline !important;
11660}
11661.PanelButton {
11662 font-size:10pt !important;
11663 font-family:Georgia !important;
11664 vertical-align:middle;
11665 width:45pt !important;
11666 height:14pt !important;
11667 line-height:1.2 !important;
11668 padding:2px !important;
11669 margin:1px !important;
11670 display:inline !important;
11671 padding:1px !important;
11672 color:#fff !important;
11673 background-color:#228 !important;
11674}
11675</style>
11676<script>
11677function DrawTextCanvas(){
11678 ctx = TextCanvas_1.getContext('2d');
11679 var textfont = '';
11680 if( TextCanvas_1_Italic.checked ) textfont += ' italic';
11681 if( TextCanvas_1_Bold.checked ) textfont += ' bold';
11682 textfont += ' '+TextCanvas_1_Size.value+'px';
11683 textfont += ' '+TextCanvas_1_Font.value;
11684 //ctx.font = 'italic bold 64px Georgia';
11685 //console.log('TxFont='+textfont);
11686 ctx.font = textfont;
11687 ctx.fillText(TextCanvas_1_Text.value,10,80);
11688}
11689DrawTextCanvas();
11690TextCanvas_1_Draw.addEventListener('click',DrawTextCanvas);
11691function ClearTextCanvas(){
11692 cv = TextCanvas_1;
11693 ctx = cv.getContext('2d');
11694 ctx.clearRect(0,0,cv.width,cv.height);
11695}
11696TextCanvas_1_Clear.addEventListener('click',ClearTextCanvas);
11697</script>
11698<!-- } -->
11699
11700<script>
11701//TextCanvas_1_Panel.addEventListener('mouseover',OffGJShell);
11702//TextCanvas_1_Panel.addEventListener('mouseout',OnGJShell);
11703</script>
11704</details>
11705<!-- FontSelector_WorkCodeSpan } -->
11706*/ </span>
11707<!-- ===== Work } ===== -->
11708
11709
11710<!-- ===== Work { ===== -->
11711<span id="Shading_WorkCodeSpan">
11712/*
11713<details><summary>Shading Canvas</summary>
11714<!-- ----- Shading Canvas // 2020-1011 SatoxITS { -->
11715<h2>Shading Canvas</h2>
11716<note class="CommandUsageText">
11717<b>Commands</b><br>
11718Placement Mode<br>
11719a ... apply (into absolute position)<br>
11720j ... bring down (ArrowDown)<br>
11721k ... bring up (ArrowUp)<br>
11722h ... bring left (ArrowLeft)<br>
11723l ... bring right (ArrowRight)<br>
11724o ... z-index = 0<br>
11725+ ... z-index += 1<br>
11726- ... z-index -= 1<br>
11727r ... return to here (relative position)<br>
11728c ... clear the log text<br>
11729Note: the HTML text must be contenteditable to catch Key Event.<br>
11730</note>
11731
11732
11733<div id="Shading_1" class="ShadingPlate" draggable="true" contenteditable="true">
11734<div id="Shading_1_Html" class="ShadingHtml" draggable="true"></div>
11735<div id="Shading_1_Log" class="ShadingLog" draggable="true" style=""></div>
11736
11737<canvas id="Shading_1_Canvas" class="ShadingCanvas" width="300px" height="300px" draggable="true" style="">My Canvas.</canvas></div>
11738</style>
11739.ShadingPlate {
11740 z-index:0;
11741 position:static;
11742 overflow:scroll;
11743 display:block;
11744 width:100%;
11745 height:400px;
11746 font-size:9pt;
11747 font-family:Courier New;
11748 border:1px dashed #000;
11749 color:#444;
11750}
11751.ShadingLog {
11752 z-index:0;
11753 position:relative;
11754 display:block;
11755 top:0px;
11756 left:0px;
11757 overflow:scroll;
11758 width:100%;
11759 font-size:9pt;
11760 font-family:Courier New;
11761 color:#666;
11762 overflow:scroll;
11763 background:rgba(200,255,200,0.4);
11764 height:400px;
11765}
11766.ShadingHtml {
11767 z-index:2;
11768 position:relative;
11769 display:block;
11770 top:0px;
11771 left:0px;
11772 overflow:scroll;
11773 width:100%;
11774 font-size:12pt;
11775 font-family:Courier New;
11776 color:#666;
11777 overflow:scroll;
11778 background:rgba(200,255,200,0.4);
11779 height:400px;
11780}

```

```

11781.ShadingCanvas {
11782   z-index:3;
11783   position:relative;
11784   xdisplay:block;
11785   top:0px;
11786   left:100px;
11787   resize:both;
11788   border:1px solid #000;
11789}
11790</style>
11791<input id="Shading_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
11792<input id="Shading_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
11793<input id="Shading_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
11794<span id="Shading_WorkCodeView"></span>
11795<script id="Shading_WorkScript">
11796function Shading_openWorkCodeView(){
11797   function Shading_showWorkCode(){
11798     showHtmlCode(Shading_WorkCodeView,Shading_WorkCodeSpan);
11799   }
11800   Shading_WorkCodeViewOpen.addEventListener('click',Shading_showWorkCode);
11801}
11802const BR = '<'+br>';
11803Shading_openWorkCodeView(); // should be invoked by an event
11804function sh_onClick(e){
11805   Shading_1_Log.innerHTML += 'Click '+e.target.nodeName+'#'+e.target.id
11806   +' offset('+e.offsetX+', '+e.offsetY+')'
11807   +' client('+e.clientX+', '+e.clientY+')'
11808   +' page('+e.pageX+', '+e.pageY+')'
11809   +' screen('+e.screenX+', '+e.screenY+')'
11810   +BR;
11811   e.stopPropagation();
11812   e.preventDefault();
11813 }
11814function sh_onKeyUp(e){
11815   if( Shading_1.style.zIndex == '' ){
11816     Shading_1.style.zIndex = 0;
11817   }
11818   zi = parseInt(Shading_1.style.zIndex);
11819
11820   if( e.key.length == 1 ){
11821     Shading_1_Html.innerHTML += e.key;
11822   }
11823
11824   if( e.key == '0' ){ zi = 0; }else
11825   if( e.key == '+' ){ zi += 1; }else
11826   if( e.key == '-' ){ zi -= 1; }else
11827   if( e.key == 'c' ){
11828     Shading_1_Log.innerHTML = '';
11829   }else
11830   if( e.key == 'r' ){
11831     Shading_1.style.position = "relative";
11832     Shading_1.style.top = '0px';
11833     Shading_1.style.left = '0px';
11834     zi = 0;
11835   }else
11836   if( e.key == 'j' || e.code == 'ArrowDown' ){
11837     topx = parseInt(Shading_1.style.top) + 50;
11838     Shading_1.style.top = topx + 'px';
11839   }else
11840   if( e.key == 'k' || e.code == 'ArrowUp' ){
11841     topx = parseInt(Shading_1.style.top) - 50;
11842     Shading_1.style.top = topx + 'px';
11843   }else
11844   if( e.key == 'l' || e.code == 'ArrowRight' ){
11845     lefty = parseInt(Shading_1.style.left) + 50;
11846     Shading_1.style.left = lefty + 'px';
11847   }else
11848   if( e.key == 'h' || e.code == 'ArrowLeft' ){
11849     lefty = parseInt(Shading_1.style.left) - 50;
11850     Shading_1.style.left = lefty + 'px';
11851   }else
11852   if( e.key == 'a' ){
11853     Shading_1.style.position = "absolute";
11854     Shading_1.style.top = '0px';
11855     Shading_1.style.left = '0px';
11856   }else{
11857   }
11858   Shading_1.style.zIndex = zi;
11859   Shading_1_Log.innerHTML += 'Keyup..' + e.target.nodeName + '#' + e.target.id
11860   +'Up('+e.key+'/' + e.code+')'
11861   +'z-index:'+zi+'/' + Shading_1.style.zIndex
11862   +'top:'+Shading_1.style.top
11863   +BR;
11864   e.stopPropagation();
11865   e.preventDefault();
11866 }
11867function sh_onKeyDown(e){
11868   Shading_1_Log.innerHTML += 'Keydown'+e.target.nodeName+'#'+e.target.id
11869   +'Down('+e.key+'/' + e.code+')'+BR;
11870   e.stopPropagation();
11871   e.preventDefault();
11872 }
11873function Shading_Setup(){
11874   Shading_1_Log.innerHTML += '<'+h4>Click here<'+/h4>';
11875   Shading_1.addEventListener('keydown',sh_onKeyDown);
11876   Shading_1.addEventListener('keyup',sh_onKeyUp);
11877   Shading_1.addEventListener('click',sh_onClick);
11878   Shading_1.addEventListener('click',sh_onClick);
11879
11880   Shading_1_Log.style.top = "-400px";
11881   Shading_1_Log.style.left = "200px";
11882
11883   Shading_1.appendChild(Shading_1_Canvas);
11884   Shading_1_Canvas.style.width = "300px";
11885   Shading_1_Canvas.style.height = "300px";
11886   Shading_1_Canvas.style.position = "relative";
11887   Shading_1_Canvas.style.top = "-750px";
11888   Shading_1_Canvas.style.left = "100px";
11889
11890   const ctx = Shading_1_Canvas.getContext('2d');
11891   ctx.fillStyle = 'rgba(160,0,0,0.9)';
11892   ctx.fillRect(50,50,40,40);
11893   ctx.fillStyle = 'rgba(0,160,0,0.9)';
11894   ctx.fillRect(60,60,40,40);
11895   ctx.fillStyle = 'rgba(0,0,160,0.9)';
11896   ctx.fillRect(70,70,40,40);
11897}
11898function Reset_ShadingCanvas(){
11899   Shading_1_Log.removeAttribute('style');
11900   Shading_1_Log.innerHTML = '';
11901   Shading_1_Canvas.style = "";
11902   //Shading_1_Canvas.removeAttribute('style');
11903}
11904

```



```

11905</script>
11906</details>
11907<!-- Shading_WorkCodeSpan } -->
11908*/ //</span>
11909//<!-- ===== Work } ===== -->
11910
11911
11912//<!-- ===== Work { ===== -->
11913//</span id="Charmap_WorkCodeSpan">
11914/*
11915<details id="Charmap_Work"><summary>Character Map</summary>
11916<!-- ----- UnicodeCharmap // 2020-1015 SatoxITS { -->
11917<h2>Unicode Character Map</h2>
11918<note>code 0x0000 - 0xFFFF / 16px / 3200 x 3200 px / zoom:0.25</note>
11919<div id="Charmap_1_Frame">
11920<div id="Charmap_1_Text" class="Charmap">
11921</div>
11922</div>
11923<br>
11924
11925<style>
11926#Charmap_1_Frame {
11927  overflow:scroll;
11928  height:3200px; width:3200px;
11929  xtransform:scale(0.5);
11930  zoom:0.25;
11931  resize:both;
11932  background-color:#fff;
11933}
11934.Charmap {
11935  xzoom:0.25;
11936  font-size:16px;
11937  line-height:1.0;
11938  xxfont-family:Georgia;
11939  color:#000;
11940}
11941</style>
11942<script>
11943function charmapgen(){
11944  text = '';
11945  for( cc = 0; cc < 0x10000; cc++){
11946    text += String.fromCharCode(cc);
11947  }
11948  Charmap_1_Text.innerHTML = text;
11949}
11950Charmap_Work.addEventListener('click', charmapgen);
11951//charmapgen();
11952</script>
11953
11954<input id="Charmap_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
11955<input id="Charmap_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
11956<input id="Charmap_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
11957<span id="Charmap_WorkCodeView"></span>
11958<script id="Charmap_WorkScript">
11959function Charmap_openWorkCodeView(){
11960  function Charmap_showWorkCode(){
11961    showHtmlCode(Charmap_WorkCodeView, Charmap_WorkCodeSpan);
11962  }
11963  Charmap_WorkCodeViewOpen.addEventListener('click', Charmap_showWorkCode);
11964}
11965Charmap_openWorkCodeView(); // should be invoked by an event
11966</script>
11967</details>
11968<!-- Charmap_WorkCodeSpan } -->
11969*/ //</span>
11970//<!-- ===== Work } ===== -->
11971
11972
11973//<!-- ===== Work { ===== -->
11974//</span id="Pointillism_WorkCodeSpan">
11975/*
11976<details open=""><summary>Collaborated Pointillism</summary>
11977<!-- ----- CollaboratedPointillism // 2020-1016 SatoxITS { -->
11978<a name="Pointillism" class="Pointillism"><h2><a href="#Pointillism">Collaborated Pointillism</a></h2></a>
11979
11980<input id="Pointillism_1_Share" type="checkbox" class="HtmlCodeViewButton" value="Share"> Share
11981<input type="button" class="HtmlCodeViewButton" onclick="Pointillism_1_ResetCanvas()" value="Reset">
11982<input type="button" class="HtmlCodeViewButton" onclick="Pointillism_1_ClearCanvas()" value="Clear">
11983<input type="button" class="HtmlCodeViewButton" onclick="Pointillism_1_ReplayCanvas()" value="Replay">
11984<input type="button" class="HtmlCodeViewButton" onclick="Pointillism_1_RepeatCanvas()" value="Repeat">
11985<input type="button" class="HtmlCodeViewButton" onclick="Pointillism_1_ClearCanvas()" value="Save">
11986<input type="button" class="HtmlCodeViewButton" onclick="Pointillism_1_ClearCanvas()" value="Load">
11987<div id="Pointillism_1" class="Pointillism">
11988
11989<span id="Pointillism_1_Unit_1" class="Pointillism_Unit">
11990<span id="Pointillism_1_XY_1" class="Pointillism_XY">XY1</span>
11991<span id="Pointillism_1_XY_1_Remote" class="Pointillism_XY_Remote">XY1-remote</span>
11992<canvas id="Pointillism_1_Canvas_1" class="Pointillism_Canvas" width="300px" height="300px"></canvas>
11993</span>
11994
11995<span id="Pointillism_1_Unit_2" class="Pointillism_Unit">
11996<span id="Pointillism_1_XY_2" class="Pointillism_XY">XY2</span>
11997<span id="Pointillism_1_XY_2_Remote" class="Pointillism_XY_Remote">XY2-remote</span>
11998<canvas id="Pointillism_1_Canvas_2" class="Pointillism_Canvas" width="300px" height="300px"></canvas>
11999</span>
12000
12001</div>
12002<br>
12003
12004<style>
12005.Pointillism {
12006  xdisplay:block;
12007  resize:both;
12008  width:680px;
12009  height:380px;
12010  min-width:240px;
12011  min-height:270px;
12012  background-color:#eee;
12013  overflow:scroll;
12014  font-size:16px;
12015  font-family:Georgia;
12016  color:#000;
12017  vertical-align:middle;
12018}
12019.Pointillism_Unit {
12020  position:relative;
12021  top:0px;
12022  display:block;
12023  overflow:scroll;
12024  width:300px;
12025  height:350px;
12026  margin:5px;
12027  padding:10px;
12028  background-color:rgba(255,255,127,0.7);

```

```

12029}
12030.Pointillism_XY {
12031  display:block;
12032  vertical-align:middle;
12033  width:290px;
12034  xxheight:20px;
12035  font-size:12px;
12036  line-height:1.2;
12037  padding:5px;
12038  margin:0px;
12039  color:#fff;
12040  background-color:#44c;
12041}
12042.Pointillism_XY_Remote {
12043  display:block;
12044  vertical-align:middle;
12045  width:290px;
12046  xxheight:20px;
12047  font-size:12px;
12048  line-height:1.2;
12049  padding:5px;
12050  color:#fff;
12051  background-color:#4a4;
12052}
12053.Pointillism_Canvas {
12054  display:block;
12055  position:relative;
12056  xpadding:20px;
12057  xleft:20px;
12058  xtop:20px;
12059  background-color:#333;
12060}
12061</style>
12062<script>
12063var points = [];
12064var replay = [];
12065var replayx = 0;
12066function pClearCanvas(can){
12067  ctx = can.getContext('2d');
12068  ctx.clearRect(0,0,can.width,can.height);
12069}
12070function Pointillism_1_ClearCanvas(){
12071  pClearCanvas(Pointillism_1_Canvas_1);
12072  pClearCanvas(Pointillism_1_Canvas_2);
12073}
12074function PointsReset(){
12075  points = [];
12076  replay = [];
12077  inRepeat = false;
12078  inReplay = false;
12079  Pointillism_1_ClearCanvas();
12080}
12081function Pointillism_1_ResetCanvas(){
12082  PointsReset();
12083  if( Pointillism_1_Share.checked ){
12084    //alert('---broad cast reset\n');
12085    GJ_BcastMessage('DRAW RESET');
12086  }
12087}
12088function Pointillism_1_ResetCanvasReceive(){
12089  //alert('---received reset\n');
12090  PointsReset();
12091}
12092function drawPoint(can,x,y,r,g,b){
12093  const ctx = can.getContext('2d');
12094  ctx.fillStyle = 'rgba('+r+', '+g+', '+b+',0.7)';
12095  ctx.fillRect(x,y,8,8);
12096}
12097function waitMs(serno,ms){
12098  console.log('-- wait #' +serno+ ' +ms+ms');
12099  until = new Date();
12100  now = until.getTime();
12101  untilMs = now + ms;
12102  for( wi = 0; ; wi++){
12103    now = new Date();
12104    nowMs = now.getTime();
12105    remMs = untilMs - nowMs;
12106    //console.log('wait '+wi+' : '+remMs+'/'+ms);
12107    if( remMs < 0 ){
12108      break;
12109    }
12110  }
12111}
12112var inReplay = false;
12113function replay1(){
12114  rx = replayx;
12115  if( replay.length <= rx ){
12116    return;
12117  }
12118  replayx += 1;
12119  pl = replay[rx];
12120  if( pl[1] == 1 ){
12121    can = Pointillism_1_Canvas_1;
12122  }else{
12123    can = Pointillism_1_Canvas_2;
12124  }
12125  drawPoint(can,pl[2],pl[3],pl[4],pl[5],pl[6]);
12126  if( inReplay == false ){
12127    console.log('wait '+replayx+' Stopped');
12128    return;
12129  }
12130  if( rx < replay.length-1 ){
12131    prevMs = replay[rx][0].getTime();
12132    nextMs = replay[rx+1][0].getTime();
12133    delayMs = nextMs - prevMs;
12134    //console.log('wait '+replayx+' '+delayMs+'ms');
12135    window.setTimeout(replay1,delayMs);
12136  }else{
12137    console.log('wait '+replayx+' Finished');
12138    if( inRepeat ){
12139      window.setTimeout(repeat1,1000);
12140    }
12141  }
12142}
12143function Pointillism_1_ReplayCanvas(can){
12144  Pointillism_1_ClearCanvas();
12145  replay = points;
12146  replayx = 0;
12147  inReplay = true;
12148  replay1();
12149}
12150var inRepeat = false;
12151function repeat1(){
12152  Pointillism_1_ClearCanvas();

```

```

12153     replay = points;
12154     replayx = 0;
12155     replay1();
12156     if( inRepeat ){
12157         //window.setTimeout(repeat1,1000);
12158     }
12159 }
12160function Pointillism_1_RepeatCanvas(can){
12161     if( inRepeat ){
12162         inRepeat = false;
12163         inReplay = false;
12164     }else{
12165         inRepeat = true;
12166         inReplay = true;
12167         repeat1();
12168     }
12169 }
12170
12171function CopyLocal(){ return Pointillism_1_Share.checked == false; }
12172function Pointillism_Setup(){
12173     var moveCount1 = 0;
12174     var moveCount2 = 0;
12175
12176     var gjlinked = false;
12177     function GJdraw(msg){
12178         if( gjlinked == false ){
12179             //GJLink_Section.open = true;
12180             GJ_Join();
12181             gjlinked = true;
12182         }
12183         GJ_BcastMessage('DRAW '+msg);
12184     }
12185     function showXY1(e){
12186         moveCount1 += 1;
12187         x = e.offsetX;
12188         y = e.offsetY;
12189         Pointillism_1_XY_1_innerHTML = 'XY1: '+ 'x'+x +', y='+y + '/' +moveCount1+'/' +points.length;
12190         Pointillism_1_XY_2_Remote_innerHTML = 'XY1: '+ 'x'+x +', y='+y + '/' +moveCount1;
12191         if( e.buttons || CopyLocal() ){
12192             points.push([new Date(),1,x,y,64,64,255]);
12193             drawPoint(Pointillism_1_Canvas_1,x,y,128,128,255);
12194             if( CopyLocal() ){
12195                 drawPoint(Pointillism_1_Canvas_2,x,y,160,160,255);
12196             }
12197             GJdraw('1','+x+','+y');
12198         }
12199     }
12200     function showXY2(e){
12201         moveCount2 += 1;
12202         x = e.offsetX;
12203         y = e.offsetY;
12204         Pointillism_1_XY_2_innerHTML = 'XY2: '+ 'x'+x +', y='+y + '/' +moveCount2+'/' +points.length;
12205         Pointillism_1_XY_1_Remote_innerHTML = 'XY2: '+ 'x'+x +', y='+y + '/' +moveCount1;
12206         if( e.buttons || CopyLocal() ){
12207             points.push([new Date(),2,x,y,64,255,64]);
12208             drawPoint(Pointillism_1_Canvas_2,x,y,128,255,128);
12209             if( CopyLocal() ){
12210                 drawPoint(Pointillism_1_Canvas_1,x,y,160,255,160);
12211             }
12212             //GJdraw('2','+x+','+y');
12213         }
12214     }
12215     Pointillism_1_Canvas_1.addEventListener('mousemove',showXY1);
12216     Pointillism_1_Canvas_2.addEventListener('mousemove',showXY2);
12217     Pointillism_1_Unit_2.style.left = '340px';
12218     Pointillism_1_Unit_2.style.top = '-375px';
12219 }
12220function Pointillism_RemoteDraw(arg){
12221     //alert('Draw at '+arg);
12222     //drawPoint(Pointillism_1_Canvas_2,x,y,160,160,255);
12223     if( arg == 'RESET' ){
12224         Pointillism_1_ResetCanvasReceive();
12225     }else{
12226         argv = arg.split(',');
12227         x = argv[1];
12228         y = argv[2];
12229         Pointillism_1_XY_2_Remote_innerHTML = 'XYR: '+ 'x'+x +', y='+y + '/' +points.length;
12230         drawPoint(Pointillism_1_Canvas_2,x,y,255,0,0);
12231     }
12232 }
12233Pointillism_Setup();
12234</script>
12235
12236
12237<input id="Pointillism_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
12238<input id="Pointillism_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
12239<input id="Pointillism_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
12240<span id="Pointillism_WorkCodeView"></span>
12241<script id="Pointillism_WorkScript">
12242function Pointillism_openWorkCodeView(){
12243     function Pointillism_showWorkCode(){
12244         showHtmlCode(Pointillism_WorkCodeView,Pointillism_WorkCodeSpan);
12245     }
12246     Pointillism_WorkCodeViewOpen.addEventListener('click',Pointillism_showWorkCode);
12247 }
12248Pointillism_openWorkCodeView(); /// should be invoked by an event
12249</script>
12250</details>
12251<!-- Pointillism_WorkCodeSpan -->
12252* //</span>
12253//<!-- ===== Work } ===== -->
12254
12255
12256
12257//<!-- ===== Work { ===== -->
12258//<span id="Template_WorkCodeSpan">
12259/*
12260<details><summary>Work Template</summary>
12261<!-- ----- Template of Work// 2020-0928 SatoxITS -->
12262<h2>Template of Work</h2>
12263<input id="Template_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
12264<input id="Template_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
12265<input id="Template_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
12266<span id="Template_WorkCodeView"></span>
12267<script id="Template_WorkScript">
12268function Template_openWorkCodeView(){
12269     function Template_showWorkCode(){
12270         showHtmlCode(Template_WorkCodeView,Template_WorkCodeSpan);
12271     }
12272     Template_WorkCodeViewOpen.addEventListener('click',Template_showWorkCode);
12273 }
12274Template_openWorkCodeView(); /// should be invoked by an event
12275</script>
12276</details>

```

```

12277<!-- Template_WorkCodeSpan } -->
12278*/ //</span>
12279//<!-- ===== Work } ===== -->
12280
12281
12282
12283//<!-- ===== Work { ===== -->
12284//<span id="OriginalSource_WorkCodeSpan">
12285/*
12286<details open=""><summary>Original Source</summary>
12287<!-- ----- OriginalSource // 2020-1009 SatoxITS { -->
12288<h2>Original Source of GShell</h2>
12289<input id="OriginalSource_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
12290<input id="OriginalSource_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
12291<input id="OriginalSource_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
12292<span id="OriginalSourceTextElement"></span>
12293<span id="OriginalSource_WorkCodeView"></span>
12294<script id="OriginalSource_WorkScript">
12295function OriginalSource_openWorkCodeView(){
12296    function OriginalSource_showWorkCode(){
12297        //showHtmlCode(OriginalSource_WorkCodeView,OriginalSource_WorkCodeSpan);
12298        //OriginalSourceTextElement = OriginalSourceNode;
12299        //console.log('src3=\n'+OriginalSourceNode.outerHTML);
12300        showHtmlCodeX(OriginalSource_WorkCodeView,OriginalSourceNode,
12301            /*'\n',
12302            '\n',true);
12303    }
12304    OriginalSource_WorkCodeViewOpen.addEventListener('click',OriginalSource_showWorkCode);
12305}
12306//OriginalSourceNode = document.documentElement.cloneNode();
12307//OriginalSourceNode = gsh.cloneNode(true); //=====
12308    //console.log('src0=\n'+document.documentElement.outerHTML);
12309    //console.log('src1=\n'+gsh.outerHTML);
12310    //console.log('src2=\n'+OriginalSourceNode.innerHTML);
12311OriginalSource_openWorkCodeView(); // should be invoked by an event
12312//showNodeAsHtmlSource(OriginalSource_WorkCodeView,OriginalSourceNode);
12313function SaveOriginalNode(){
12314    if( false ){
12315        m0 = performance.memory;
12316        mu0 = m0.usedJSHeapSize;
12317        console.log('-- heap bef clone: '
12318            +m0.usedJSHeapSize+'/'+m0.totalHeapSize+'/'+m0.jsHeapSizeLimit);
12319    }
12320    OriginalSourceNode = gsh.cloneNode(true);
12321    if( false ){
12322        m1 = performance.memory;
12323        mu1 = m1.usedJSHeapSize;
12324        mu = mu1 - mu0;
12325        //alert('-- clone: used heap '+mu0+' -> '+mu1+' = '+mu+' bytes');
12326        console.log('-- heap aft clone: '
12327            +m1.usedJSHeapSize+'/'+m1.totalHeapSize+'/'+m1.jsHeapSizeLimit);
12328        //OriginalSourceNode = document.documentElement.cloneNode(true);
12329    }
12330}
12331
12332function Gsh_setupPage(){
12333    GshSetImages();
12334    //Indexer_afterLoaded();
12335    //GShell_initKeyCommands();
12336    //GJConsole_initConsole();
12337    GJConsole_initFactory();
12338    GJLink_init();
12339    InterFrameComm_init();
12340    Gshell_initTopbar();
12341    //WirtualDesktop_init();
12342    Banner_init();
12343//    Aff_Setup();
12344    Shading_Setup();
12345    window.setInterval(ShowResourceUsage,1000);
12346    //document.addEventListener("keydown",jgshCommand); // should be applied later?
12347}
12348function OnLoad(){
12349    SaveOriginalNode();
12350    Gsh_setupPage();
12351}
12352document.addEventListener('load',Gsh_setupPage);
12353</script>
12354</details>
12355<!-- OriginalSource_WorkCodeSpan } -->
12356*/ //</span>
12357//<!-- ===== Work } ===== -->
12358
12359
12360
12361//</div>
12362//<br><script>OnLoad();</script></span>
12363

```