

```

1  #include "all103.h"
2
3  char path[260], oldpath[260];
4
5  word Chks = 0x0000;
6  unsigned char huge *buffer = NULL;
7  char buffile[80] = "C:\\BUFF.TMP";
8
9  long BufStart= 0x00000L;
10 long BufEnd  = 0x1FFFFL;
11 long bufsize = 0x20000L;
12
13 static struct fatr {
14     char fname[13];
15     long fsize;
16     char fattr;
17     struct {
18         unsigned day:5;
19         unsigned month:4;
20         unsigned year:7;
21     } fdate;
22 } f[1000];
23 static char filename[260];
24 static char fmatch[15] = "*. *";
25
26 int xgets( char *str )
27 {
28     struct text_info ti;
29     int i;
30
31     gettextinfo( &ti );
32
33     str[1] = str[2] = 0;
34
35     for( ;; )
36     {
37         gotoxy( ti.curx, ti.cury );
38         cprintf( str+2 ); cprintf( " " );
39         gotoxy( ti.curx+str[1], ti.cury );
40
41         switch( i = getch() )
42         {
43             case 0:
44                 getch();
45                 break;
46
47             case 0x09:
48                 str[1] = str[2] = 0;
49                 return 0;
50
51             case 0x1B:
52                 str[1] = str[2] = 0;
53                 return -1;
54
55             case 0x08:
56                 if( str[1] )
57                     str[ 2 + --str[1] ] = 0;
58                 break;
59
60             case '\n':
61             case '\r':
62                 return 1;
63
64             default:
65                 str[ 2 + str[1]++ ] = i;
66                 str[ 2 + str[1] ] = 0;
67                 break;
68         }
69     }
70 }
71
72 unsigned long read_bin( char *filename, unsigned long addr )

```

```

73 {
74     FILE *f, *o = NULL;
75     int i;
76
77     Chks = 0;
78
79     if( ( f = fopen( filename, "rb" ) ) == NULL )
80         return -1;
81
82     if( !buffer && ( o = fopen( buffile, "wb" ) ) == NULL )
83         return -1;
84
85     while( ( i = getc( f ) ) != EOF && addr < bufsize )
86     {
87         if( !buffer ) putc( i, o );
88         else buffer[addr] = i;
89
90         Chks += i;
91         ++addr;
92     }
93
94     if( o ) fclose( o );
95     fclose( f );
96     return addr-1;
97 }
98
99 static char *pos;
100
101 static unsigned char getval( void )
102 {
103     unsigned char c1, c2;
104
105     c1 = toupper( *pos++ );
106     c2 = toupper( *pos++ );
107
108     c1 = ( (c1 > '9') ? c1 - 'A'+10 : c1 - '0' );
109     c2 = ( (c2 > '9') ? c2 - 'A'+10 : c2 - '0' );
110
111     return ( c1 << 4 ) | c2;
112 }
113
114 unsigned long read_intel( char *filename, unsigned long addr )
115 {
116     FILE *f, *o = NULL;
117     char l[160];
118     unsigned a;
119     unsigned char d, chk, num;
120
121     addr = 0;
122     Chks = 0;
123
124     if( ( f = fopen( filename, "r" ) ) == NULL )
125         return -1;
126
127     if( !buffer && ( o = fopen( buffile, "wb" ) ) == NULL )
128         return -1;
129
130     while( ( pos = fgets( l, 160, f ) ) != NULL )
131     {
132         if( *pos++ != ':' )
133             continue;
134
135         chk = ( num = getval() ); // num bytes
136         chk += ( d = getval() ); a = 256 * d; // address high
137         chk += ( d = getval() ); a |= d; // address low
138         chk += getval(); // type
139
140         while( num-- )
141         {
142             chk += ( d = getval() );
143             if( a < bufsize )
144                 {

```

```

145         Chks += d;
146         if( !buffer ) putc( d, o );
147         else buffer[a] = d;
148         ++a;
149     }
150 }
151 if( ( ( chk + getval() ) & 0xFF ) != 0 )
152 {
153     errbeep();
154     if( o ) fclose( o );
155     fclose( f );
156     return 0L;
157 }
158 if( a > addr )
159     addr = a;
160 }
161
162 if( o ) fclose( o );
163 fclose( f );
164 return addr-1;
165 }
166
167 unsigned long read_motorola( char *filename, unsigned long addr )
168 {
169     *filename = 0;
170     delay(2000);
171     return addr+100;
172 }
173
174 static int cmp( const void *f1, const void *f2 )
175 {
176     struct fatr *f1 = (struct fatr *)f1;
177     struct fatr *f2 = (struct fatr *)f2;
178
179     if( f1->fattnr != f2->fattnr ) return f1->fattnr - f2->fattnr;
180     else return strcmp( f1->fname, f2->fname );
181 }
182
183 int load_file( void )
184 {
185     int i, done, left, top, newpath, numfiles, pos;
186     unsigned long addr;
187     char tmp[33], *cptr;
188
189     textattr( ( LIGHTGRAY << 4 ) | YELLOW ); clrscrn( 0,0, 24,39 );
190
191     textattr( ( CYAN << 4 ) | WHITE );
192     _window( 0,0, 22,39 ); // left
193     _window( 2,1, 16,7 ); // drives
194     locate( 13, 41 ); cprintf( "File name: " );
195
196     textattr( ( BLUE << 4 ) + WHITE );
197     locate( 11, 41); cprintf( "  LOAD : " );
198
199 More:
200     textattr( ( CYAN << 4 ) | WHITE );
201     _window( 12,40, 22,79 ); // right
202     locate( 13, 41 ); cprintf( "File name: " );
203     filename[0] = 0;
204
205     newpath = 1;
206     left = 0;
207
208     for( done = 0; !done; )
209     {
210         textattr( ( LIGHTGRAY << 4 ) | YELLOW ); clrscrn( 23,0, 23,79 );
211         textattr( ( BLUE << 4 ) + WHITE );
212         locate( 23, 0 );
213         if( left )
214             cprintf( "Command:Tab PgUp PgDn Up Home End Esc Enter Drive-letter
215                     Fl:files" );
216         else

```

```

216         cprintf( "Command:Tab Esc Enter" );
217
218     textattr( ( CYAN << 4 ) | WHITE );
219     if( left ) dbl = 1; frame( 0,0, 22,39 );
220     dbl ^= 1; frame( 12,40, 22,79 );
221     dbl = 0;
222
223     if( newpath )
224     {
225         struct ffblk ff;
226         char fname[260];
227
228         newpath = numfiles = 0;
229
230         for( i = 0; i < 13; ++i )
231         {
232             if( path[0] == 'A'+i )
233                 textattr( ( BLUE << 4 ) + WHITE );
234             else
235                 textattr( ( CYAN << 4 ) | WHITE );
236             locate( 3+i, 2 ); cprintf( "%c:", 'A'+i );
237
238             if( path[0] == 'N'+i )
239                 textattr( ( BLUE << 4 ) + WHITE );
240             else
241                 textattr( ( CYAN << 4 ) | WHITE );
242             locate( 3+i, 5 ); cprintf( "%c:", 'N'+i );
243         }
244         textattr( ( CYAN << 4 ) | WHITE );
245
246         strcpy( fname, path );
247         strcat( fname, "\\*.*)" );
248
249         done = findfirst( fname, &ff, 0xFFFF );
250         while( !done && numfiles < 1000 )
251         {
252             if( ( ff.ff_attrib & FA_DIRC ) != 0 &&
253                 !( ff.ff_name[0] == '.' && ff.ff_name[1] == 0 ) )
254             {
255                 strcpy( f[numfiles].fname, ff.ff_name );
256                 f[numfiles].fattr = FA_DIRC;
257                 *(unsigned*)&f[numfiles].fdate = ff.ff_fdate;
258                 f[numfiles++].fsize = 0;
259             }
260             done = findnext( &ff );
261         }
262
263         strcpy( fname, path );
264         strcat( fname, "\\\" );
265         strcat( fname, fmatch );
266
267         done = findfirst( fname, &ff, 0xFFFF );
268         while( !done && numfiles < 1000 )
269         {
270             if( ( ff.ff_attrib & ( FA_LABEL | FA_DIRC ) ) == 0 )
271             {
272                 strcpy( f[numfiles].fname, ff.ff_name );
273                 f[numfiles].fattr = ff.ff_attrib;
274                 *(unsigned*)&f[numfiles].fdate = ff.ff_fdate;
275                 f[numfiles++].fsize = ff.ff_fsize;
276             }
277             done = findnext( &ff );
278         }
279
280         textattr( ( BLUE << 4 ) + WHITE );
281         locate( 1, 1 ); cprintf( "%-38.38s", fname );
282         textattr( ( CYAN << 4 ) | WHITE );
283
284         qsort( &f, numfiles, sizeof(f[0]), cmp );
285         done = pos = top = 0;
286     }
287

```

```

288     for( i = 0; i < 18; ++i )
289     {
290         struct fatr *ff = &f[i+top];
291
292         if( i+top == pos )
293             textattr( ( BLUE << 4 ) + WHITE );
294         else
295             textattr( ( CYAN << 4 ) | WHITE );
296         locate( i+3, 8 );
297         if( i+top < numfiles )
298         {
299             cprintf( "%-12s ", ff->fname );
300             if( ff->fattr & FA_DIRECT )
301                 cprintf( " <DIR>" );
302             else
303                 cprintf( "%7ld", ff->fsize );
304             cprintf( " %02d-%02d-%02d",
305                     ff->fdate.month, ff->fdate.day, (ff->fdate.year+80)%100 );
306         }
307         else
308             cprintf( "
309     }
310
311     textattr( ( CYAN << 4 ) | WHITE );
312     locate( 13, 52 ); cprintf( "%s ", filename );
313     locate( 13, 52+strlen(filename) );
314
315     switch( i = getch() )
316     {
317         case 0:
318             if( !left )
319             {
320                 getch();
321                 break;
322             }
323
324             switch( i = getch() )
325             {
326                 case 59: // F1
327                     textattr( ( BLUE << 4 ) + WHITE );
328                     clrscrn( 21, 1, 21, 38 );
329                     locate( 21, 1 ); cprintf( "Files(%-12s):", fmatch );
330                     tmp[0] = 0;
331
332                     for( done = 0; !done; )
333                     {
334                         locate( 21, 21 ); cprintf( "%s ", tmp );
335                         locate( 21, 21+strlen(tmp) );
336
337                         switch( i = getch() )
338                         {
339                             case 0:
340                                 getch();
341                                 break;
342
343                             case 8:
344                                 if( ( i = strlen( tmp ) ) != 0 )
345                                     tmp[ i-1 ] = 0;
346                                 break;
347
348                             case 0x1B:
349                                 done = 1;
350                                 break;
351
352                             case '\n':
353                             case '\r':
354                                 if( ( i = strlen(tmp) ) > 0 && i < 13 )
355                                 {
356                                     strcpy( fmatch, tmp );
357                                     newpath = 1;
358                                 }
359                                 done = 1;

```

```

360             break;
361
362         default:
363             if( strlen( tmp ) < 12 )
364                 strcat( tmp, (char*)&i );
365             break;
366         }
367
368     }
369     textattr( ( CYAN << 4 ) | WHITE );
370     clrscrn( 21, 1, 21, 38 );
371     done = 0;
372     break;
373
374     case 71: // HOME
375         pos = top = 0;
376         break;
377
378     case 79: // END
379         if( ( top = numfiles-18 ) < 0 )
380             top = 0;
381         pos = numfiles - 1;
382         break;
383
384     case 72: // UP
385         if( pos )
386             --pos;
387         if( pos < top )
388             top = pos;
389         break;
390
391     case 80: // DOWN
392         if( pos+1 < numfiles )
393         {
394             ++pos;
395             if( pos >= top+18 )
396                 ++top;
397         }
398         break;
399
400     case 73: // PGUP
401         if( ( pos -= 9 ) < 0 )
402             pos = 0;
403         if( pos < top && ( top -= 9 ) < 0 )
404             top = 0;
405         break;
406
407     case 81: // PGDN
408         if( ( pos += 9 ) >= numfiles )
409             pos = numfiles-1;
410
411         if( pos >= top+18 )
412         {
413             top += 9;
414             if( top + 18 >= numfiles )
415                 top = numfiles - 18;
416             if( top < 0 )
417                 top = 0;
418         }
419         break;
420     }
421     break;
422
423     case 8: // BS
424         if( ( i = strlen( filename ) ) > 0 )
425             filename[i-1] = 0;
426         break;
427
428     case 9:
429         left ^= 1; break;
430
431     case 0x1B:

```

```

432         return 0;
433
434     case '\n':
435     case '\r':
436         if( !left )
437         {
438             strcpy( tmp, filename );
439             sprintf( filename, "%s\\%s", path, tmp );
440             done = 1;
441         }
442         else if( f[pos].fattr & FA_DIREC )
443         {
444             if( !strcmp( f[pos].fname, "." ) )
445                 break;
446
447             newpath = 1;
448             if( !strcmp( f[pos].fname, ".." ) )
449                 *strrchr( path, '\\' ) = 0;
450             else
451             {
452                 strcat( path, "\\" );
453                 strcat( path, f[pos].fname );
454             }
455         }
456         else
457         {
458             sprintf( filename, "%s\\%s", path, f[pos].fname );
459             done = 1;
460         }
461         break;
462
463     default:
464         if( left )
465         {
466             if( isalpha( i ) )
467             {
468                 struct ffblk ff;
469                 char p[260];
470
471                 sprintf( p, "%c:\\*.\"", toupper(i) );
472                 if( !findfirst( p, &ff, 0xFFFF ) )
473                 {
474                     newpath = 1;
475                     sprintf( path, "%c:", toupper(i) );
476                 }
477             }
478             else if( strlen( filename ) < 12 )
479                 strcat( filename, (char*)&i );
480             break;
481         }
482     }
483 }
484
485 textattr( ( CYAN << 4 ) | WHITE );
486 if( ( cptr = strrchr( filename, '\\' ) ) != NULL )
487     ++cptr;
488 else
489     cptr = filename;
490 locate( 13, 52 ); cprintf( "%s ", cptr );
491
492 locate( 14, 41 ); cprintf( "<B>in,<I>ntel HEX,<M>otorola S HEX : " );
493
494 for( done = 0; !done; )
495 {
496     switch( toupper( getch() ) )
497     {
498     case 0:      getch(); break;
499     case 'B':    i = 'B'; done = 1; break;
500     case 'I':    i = 'I'; done = 1; break;
501     case 'S':    i = 'S'; done = 1; break;
502     case 0x1B:   return 0;
503     }

```

```

504     }
505
506     locate( 15, 41 ); cprintf( "Load address(%05X) : ", addr = BufStart );
507
508     tmp[0] = 6; if( xgets( tmp ) == -1 ) return 0;
509     if( tmp[1] )
510         addr = atoi( tmp+2 );
511
512     textattr( BLINK | ( LIGHTGREEN << 4 ) | WHITE );
513     locate( 16, 41 ); cprintf( "Loading..." );
514
515     switch( i )
516     {
517     case 'B':  addr = read_bin( filename, addr ); break;
518     case 'I':  addr = read_intel( filename, addr ); break;
519     case 'S':  addr = read_motorola( filename, addr ); break;
520     }
521
522     textattr( ( CYAN << 4 ) | WHITE );
523     locate( 16, 41 ); cprintf( "Loaded... " );
524
525     putchar( 7 );
526     locate( 17, 41 ); cprintf( "Ok !   END ADDR. : %05X", addr );
527
528     locate( 20, 41 ); cprintf( "Press any key to continue" );
529     locate( 21, 41 ); cprintf( "Or press CR to back to menu" );
530
531     switch( getch() )
532     {
533     case 0:      getch();
534     case '\n':
535     case '\r':   return 1;
536     default :   goto More;
537     }
538 }
539
540 void save_file( void )
541 {
542     char fname[40], tmp[20], done;
543     unsigned char v;
544     unsigned long i, start = BufStart, ende = BufEnd;
545     FILE *f, *o;
546
547     textattr( ( CYAN << 4 ) | WHITE );
548     _window( 11,40, 23,79 );
549
550     textattr( ( BLUE << 4 ) + WHITE );
551     locate( 11, 41 ); cprintf( "  SAVE : " );
552
553     for( done = 0; !done; )
554     {
555         textattr( ( CYAN << 4 ) | WHITE ); clrscrn( 12,41, 22,78 );
556
557         locate( 12, 41 ); cprintf( "File name: " );
558         fname[0] = 30; if( xgets( fname ) != 1 || fname[1] == 0 ) return;
559
560         locate( 13, 41 ); cprintf( "Start address[%05X]:", BufStart );
561         tmp[0] = 6; if( xgets( tmp ) != 1 ) return;
562         if( tmp[1] ) start = atoi( tmp+2 );
563
564         locate( 14, 41 ); cprintf( "End address[%05X]:", BufEnd );
565         tmp[0] = 6; if( xgets( tmp ) != 1 ) return;
566         if( tmp[1] ) ende = atoi( tmp+2 );
567
568         if( ( o = fopen( fname+2, "rb" ) ) != NULL )
569         {
570             fclose( o );
571
572             textattr( ( RED << 4 ) | WHITE );
573             locate( 15, 41 ); cprintf( "FILE EXISTS! OVERWRITE(Y/N)? " );
574             textattr( ( CYAN << 4 ) | WHITE );
575

```

```

576         if( ( done = getch() ) == 0 )
577             getch();
578         else if( toupper(done) != 'Y' )
579         {
580             done = 0;
581             continue;
582         }
583         else
584         {
585             done = 0;
586             clrscrn( 15,41, 15,78 );
587         }
588     }
589
590     locate( 15, 41 ); cprintf( "Saving now..." );
591
592     locate( 16, 41 );
593     if( ( o = fopen( fname+2, "wb" ) ) == NULL )
594     {
595         errbeep();
596         textattr( ( RED << 4 ) | WHITE ); cprintf( "File open error !" );
597         return;
598     }
599
600     if( !buffer && ( f = fopen( buffile, "rb" ) ) == NULL )
601     {
602         errbeep();
603         textattr( ( RED << 4 ) | WHITE ); cprintf( "File open error !" );
604         return;
605     }
606
607     for( i = start; i <= ende; ++i )
608     {
609         v = ( buffer ? buffer[i] : getch(f) );
610         if( putc( v, o ) == EOF )
611         {
612             errbeep();
613             textattr( ( RED << 4 ) | WHITE ); cprintf( "File write error !" );
614             fclose( o );
615             return;
616         }
617     }
618
619     fclose( o );
620     if( !buffer ) fclose( f );
621
622     putchar( 7 );
623     cprintf( " OK !" );
624
625     locate( 20, 41 ); cprintf( "Press any key to continue" );
626     locate( 21, 41 ); cprintf( "Or press CR to back to menu " );
627
628     if( ( done = getch() ) == 0 )
629         getch();
630     else if( ( done = getch() ) == 0x1B || done == '\r' )
631         return;
632     else
633         done = 0;
634 }
635 }
636
637
638

```