



```
0000 1: ;-----
0000 2: ;
0000 3: ;      Copyright 2009 Intergraded Logic Systems
0000 4: ;      Source Code is Copyright Stephen J. Car
0000 5: ;
0000 6: ;
0000 7: ;      Please do not share this source!
0000 8: ;-----
0000 9: ;      Editfile utility for real.dos v1.0)
0000 10: ;
0000 11: ;-----
0000 12: ;      Real.dos file2pc.com
0000 13: ;      This program is to replace
0000 14: ;      file2pc.com
0000 15: ;
0000 16: ;      FILE2PC - REV. 1.1
0000 17: ;
0000 18: ; ATTEMPTING REVISION DUE TO BUG
0000 19: ; REPORTS: 2/25/92
0000 20: ; REVISING 3/23/92 DUE TO ERROR
0000 21: ; SENDING FILE: GOING TO ADD A RETRY
0000 22: ; COUNTER AND 3 SECOND DELAY FOR
0000 23: ; RETRIES: REV 1.1
0000 24: ; A PROGRAM TO TRANSFER A FILE FROM
0000 25: ; THE ATARI TO THE PC VIA SIO2PC
0000 26: ;
0000 27: ; SIO2PC IS ADDRESSED AS DISK #$A
0000 28: ; WHICH IS SIO BUS DEVID $3A
0000 29: ;
0000 30: ;
0000 31: ;      Code History.  Init design, 11/09/2005  sjc
0000 32: ;      cleaned upcode formatting so it is
0000 33: ;
0000 34: ; Updated the screen Print Utility so it will work with the
0000 35: ; This change will also help "Print P:"      8/1
0000 36: ;
0000 37: ;
0000 38: ; Also added code for Control+1 pause of screen to the high
0000 39: ; handling.      8/24/2012  sjc
0000 40: ;
0000 41: ;
0000 42: ;
0000 43: ;
0000 44: ;
0000 45: ;
0000 46: ;-----
0000 47: ; Notes: o This source code MAY NOT be placed for download
0000 48: ;      o "mea" is a macro that loads the address of the
0000 49: ;      into the pointer specified by the second field.
0000 50: ;-----
0000 51: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 52: ;      XASM  st and IBM VERSIONS
0000 53: ;-----
```

```
0000 54:
0000 55: com_file_name: .macro
0000 56: ; dc.b 13,"filename.ext",$9b
0000 57: dc.b 12,"File2pc.com ",$9b
0000 58: .endm
0000 59:
0000 60: ;
0000 61: ;~~~~~
0000 62: ; File revision history. Version Number in hex
0000 63: ;
0000 64: file_ver: equ $15
0000 65: ;~~~~~
0000 66: ; the Month in dec for the first time this revision
0000 67: ;was compiled
0000 68: ;
0000 69: c_month: equ 8
0000 70: ;~~~~~
0000 71: ; the day in dec for the first time this revision
0000 72: ;was compiled
0000 73: c_day: equ 24
0000 74: ;~~~~~
0000 75: ; the year in dec for the first time this revision
0000 76: ;was compiled
0000 77: c_year: equ 2012
0000 78: ;~~~~~
0000 79: ; Type of compiler used in program
0000 80: ;
0000 81: ; 0 = unknown Compiler
0000 82: ; 1 = xasm
0000 83: ; 2 = mac_65
0000 84: ; 3 = basic
0000 85: ; 4 = compiled basic xl
0000 86: ; 5 = C65
0000 87: ; 6 = Action
0000 88: ;
0000 89: ; We can add more as time goes on!
0000 90: ;
0000 91: ;
0000 92: ;
0000 93: xasm: equ 1
0000 94: mac_65: equ 2
0000 95: basic: equ 3
0000 96: basicxl: equ 4
0000 97: c65: equ 5
0000 98: action: equ 6
0000 99:
0000 100: file_compiler: equ xasm
0000 101: ;~~~~~
0000 102: ; is this relocatable code ?
0000 103: ; anyother value other than 1 or 2 would be unknown
0000 104: r.yes: equ 1
0000 105: r.no: equ 2
0000 106: ;
```

```

0000 107: ;~~~~~
0000 108: ; Gotta define if it can be relocated
0000 109: l_relocatable: equ r..no
0000 110: ;
0000 111: ;
0000 112: ;~~~~~
0000 113: ; Gotta define our Header control Byte
0000 114: ;$ff =bypass intro
0000 115: ;$15 = Bypass Dos check
0000 116: r.crl: equ $7d
0000 117: r.crlf: equ $9b
0000 118: r..space: equ $20
0000 119: r.bypassDOS: equ $15
0000 120: r.bypass: equ $ff
0000 121:
0000 122: l_frstscreenbyte: equ r..space
0000 123: ;~~~~~
0000 124: ; The language the output file is in.
0000 125: ;
0000 126: ;
0000 127: ; 0 = Undefined
0000 128: ; 1 = English
0000 129: ; 2 = German
0000 130: ;
0000 131: r..language: equ 1
0000 132: ;~~~~~
0000 133:
0000 134:
0000 135: .include equates
0000 136: .include globals
0000 137: .include macros
0000 138:
0000 139: .6502
0000 140:
0000 141: .org $5000
5000 142: win_start:
5000 143: header_info:
5000 144: .include header
52a7 145:
52a7 146:
52a7 147: INIT:
52a7 148:
52a7 ad3c56 149: LDA i.parm_1
52aa c904 150: CMP #4
52ac b000 151: BCS GO_ON
52ae 152: ;
52ae 153: GO_ON
52ae a900 154: LDA #$00
52b0 8d0b60 155: STA RC
52b3 20e55b 156: JSR printsi
52b6 9b2053 157: dc.b $9b," SIO2PC File transfer
52dd 158:
52dd 159: top:

```

```

52dd a210 160:      ldx  #disk1
52df 20195e 161:      jsr  close
52e2 a220 162:      ldx  #disk2
52e4      163: ;
52e4 20935d 164:      JSR   cr_lf
52e7 20935d 165:      JSR   cr_lf
52ea a901 166:      lda  #$01
52ec cd3b56 167:      cmp  i.in_local
52ef d065 168:      bne  jump1
52f1 20315b 169:      JSR   KEYCK
52f4 9003 170:      BCC   KOK1
52f6 4c2b56 171:      JMP   QUIT
52f9      172: KOK1:
52f9 20e55b 173:      jsr  printsi
52fc 9b9b50 174:      dc.b  $9b,$9b,"Please enter the A
531e 6f7220 175:      dc.b  "or press ESC to quit.", $9b
5334 203eff 176:      dc.b  ">",-1
5337      177:
5337      178: ; WAIT FOR KEY OTHER THAN ESCAPE..
5337      179: ;
5337 adfc02 180: ESKY:      LDA   CH
533a c91c 181:      CMP   #$1C          ; K
533c d003 182:      BNE   NT91
533e 4c2b56 183:      JMP   QUIT
5341 c9ff 184: NT91:      CMP   #$FF
5343 f0f2 185:      BEQ   ESKY
5345      186: ;
5345      187:      mea  i.parm_1,in_ptr
5345 a93c ----:      lda  #low i.parm_1
5347 8580 ----:      sta  in_ptr
5349 a956 ----:      lda  #high i.parm_1
534b 8581 ----:      sta  in_ptr + 1
534d a940 188:      lda  #64
534f a200 189:      ldx  #0
5351 a001 190:      ldy  #1
5353 20225e 191:      jsr  get_string
5356      192:
5356      193: jump1:
5356      194:      mea  i.parm_1,s_ptr
5356 a93c ----:      lda  #low i.parm_1
5358 8582 ----:      sta  s_ptr
535a a956 ----:      lda  #high i.parm_1
535c 8583 ----:      sta  s_ptr + 1
535e      195:      mea  fspc,d_ptr
535e a90b ----:      lda  #low fspc
5360 8584 ----:      sta  d_ptr
5362 a95a ----:      lda  #high fspc
5364 8585 ----:      sta  d_ptr + 1
5366 20cf5f 196:      jsr  toascii
5369      197: ;
5369 20935d 198:      JSR   cr_lf
536c      199:
536c      200: ; NOW, OPEN THE ATARI FILE TO READ:

```

```

536c      201:
536c a220  202:      ldx  #disk2
536e 20195e 203:      jsr  close
5371 ad0c5a 204:      LDA  FSPC+1
5374 c93a   205:      CMP  #'.'
5376 f017   206:      BEQ  YCOLON
5378 ad0d5a 207:      LDA  FSPC+2
537b c93a   208:      CMP  #'.'
537d f010   209:      BEQ  YCOLON
537f      210:
537f      211:      mea  prefs,fn_ptr
537f a909   ----:      lda  #low prefs
5381 8586   ----:      sta  fn_ptr
5383 a95a   ----:      lda  #high prefs
5385 8587   ----:      sta  fn_ptr + 1
5387 a220  212:      ldx  #disk2
5389 20ce5d 213:      jsr  open_in
538c 4c9c53 214:      JMP  TESTOP
538f      215: YCOLON:
538f      216:      mea  fspc,fn_ptr
538f a90b   ----:      lda  #low fspc
5391 8586   ----:      sta  fn_ptr
5393 a95a   ----:      lda  #high fspc
5395 8587   ----:      sta  fn_ptr + 1
5397 a220  217:      ldx  #disk2
5399 20ce5d 218:      jsr  open_in
539c      219:
539c      220:
539c      221: TESTOP:
539c 103d   222:      BPL  OKOP
539e a210   223:      ldx  #disk1
53a0 20195e 224:      jsr  close
53a3 a220   225:      ldx  #disk2
53a5 20195e 226:      jsr  close
53a8 20e55b 227:      jsr  printsi
53ab 9b2045 228:      dc.b  $9b," ERROR in opening the
53ce a901   229:      lda  #$01
53d0 cd3b56 230:      cmp  i.in_local
53d3 f003   231:      beq  .a
53d5 4c2b56 232:      JMP  quit
53d8 4cdd52 233: .a:      jmp  top
53db      234:
53db      235: OKOP:
53db adfc02 236:      LDA  CH
53de c91c   237:      CMP  #$1C          ; K
53e0 d003   238:      BNE  KKOK2
53e2 4cdd52 239:      JMP  top
53e5      240:
53e5      241: ; GET PC PATHNAME FROM USER:
53e5      242:
53e5      243: KKOK2:
53e5 20e55b 244:      jsr  printsi
53e8 9b9b50 245:      dc.b  $9b,$9b,"Please enter the P

```

```
5410 203eff 246:      dc.b  ">",-1
5413      247:      mea  i.parm_2,in_ptr
5413 a93b ----:      lda  #low i.parm_2
5415 8580 ----:      sta  in_ptr
5417 a957 ----:      lda  #high i.parm_2
5419 8581 ----:      sta  in_ptr + 1
541b a950 248:      lda  #80
541d a200 249:      ldx  #0
541f a001 250:      ldy  #1
5421 20225e 251:      jsr  get_string
5424      252: kkok2a:
5424 20a35b 253:      jsr  convert_it
5427      254:
5427 20935d 255:      jsr  cr_lf
542a      256:
542a      257:      mea  i.parm_2,pr_ptr
542a a93b ----:      lda  #low i.parm_2
542c 8588 ----:      sta  pr_ptr
542e a957 ----:      lda  #high i.parm_2
5430 8589 ----:      sta  pr_ptr + 1
5432 209d5d 258:      jsr  echos
5435      259:
5435 20935d 260:      jsr  cr_lf
5438      261:
5438      262:      mea  i.parm_2,s_ptr
5438 a93b ----:      lda  #low i.parm_2
543a 8582 ----:      sta  s_ptr
543c a957 ----:      lda  #high i.parm_2
543e 8583 ----:      sta  s_ptr + 1
5440      263:      mea  buffer,d_ptr
5440 a989 ----:      lda  #low buffer
5442 8584 ----:      sta  d_ptr
5444 a959 ----:      lda  #high buffer
5446 8585 ----:      sta  d_ptr + 1
5448 20cf5f 264:      jsr  toascii
544b      265: ;
544b      266:
544b      267: ; FIRST, GET PC'S STATUS
544b      268:
544b      269: GPCS:
544b 20675b 270:      JSR  GETSTAT
544e f02b 271:      BEQ  PCSOK
5450 20e55b 272:      jsr  printsi
5453 9b2042 273:      dc.b  $9b," BAD STATUS from PC. R
5478 4cdd52 274: .a:      jmp  top
547b      275:
547b      276: ; SEND IT TO THE PC ...
547b      277:
547b a957 278: PCSOK:      LDA  #'W'
547d 8d0203 279:      STA  DCOMND
5480 a901 280:      LDA  #1
5482 8d0a03 281:      STA  DAUX1
5485 a900 282:      LDA  #0
```

```

5487 8d0b03 283:          STA  DAUX2
548a 204f5b 284:          JSR  GOSIO
548d      285:
548d      286: ; NOW, WAIT 6 SECONDS FOR PC TO CREATE
548d      287: ; OR OPEN THE FILE...
548d      288: ; REV. 1.0 CORRECTED LDA BELOW TO LDX
548d      289:
548d 20e55b 290:          jsr  printsi
5490 9b2053 291:          dc.b  $9b," STANDBY 6 sec. - PC C
54b5 a2ff  292:          LDX  #255          ; 2
54b7 20435b 293:          JSR  JIFFY
54ba a27f  294:          LDX  #127
54bc 20435b 295:          JSR  JIFFY
54bf      296:
54bf      297:
54bf 20675b 298:          JSR  GETSTAT
54c2 f02a  299:          BEQ  .b
54c4 20e55b 300:          jsr  printsi
54c7 204241 301:          dc.b  " BAD STATUS from PC. Resta
54eb 4cdd52 302: .a:      jmp  top
54ee      303:
54ee      304: ; NOW START PUTTING THE ATARI FILE TO
54ee      305: ; THE PC:
54ee      306: .b:
54ee 20e55b 307:          jsr  printsi
54f1 9b2043 308:          dc.b  $9b," Copying file to PC ..
550b      309:
550b      310:
550b      311: PCOK:
550b a220  312:          LDX  #disk2
550d a907  313:          LDA  #7          ; C
550f 9d4203 314:          STA  ICCOM,X
5512 a900  315:          LDA  #0
5514 9d4903 316:          STA  ICBLL,X
5517 a980  317:          LDA  #$80          ; 1
5519 9d4803 318:          STA  ICBLL,X
551c a989  319:          LDA  #LOW BUFFER
551e 9d4403 320:          STA  ICBAL,X
5521 a959  321:          LDA  #HIGH BUFFER
5523 9d4503 322:          STA  ICBAH,X
5526 2056e4 323:          JSR  CIOV
5529      324:
5529 3007  325:          BMI  SHOP
552b c003  326:          CPY  #03          ; I
552d f06f  327:          BEQ  LAST1          ; R
552f 4ced55 328:          JMP  GMORE
5532 c088  329: SHOP:    CPY  #$88          ; M
5534 f068  330:          BEQ  LAST1
5536 ce3956 331:          DEC  RETRY          ; R
5539 302a  332:          BMI  NO_TRY          ; I
553b 20e55b 333:          jsr  printsi
553e 9b4552 334:          dc.b  $9b,"ERROR: 3 SEC DELAY AND
555d      335:

```



```

555d a2b4 336:      LDX  #180          ; 3
555f 20435b 337:      JSR  JIFFY
5562 4c0b55 338:      JMP  PCOK
5565 ad3a56 339: NO_TRY:      LDA  TRIES
5568 8d3956 340:      STA  RETRY
556b 20e55b 341:      JSR  printsi
556e 9b2020 342:      dc.b  $9b," ERROR in reading Ata
558e 20905b 343:      JSR  SAYBAD
5591 a901 344:      lda  #$01
5593 cd3b56 345:      cmp  i.in_local
5596 f003 346:      beq  .a
5598 4c2b56 347:      JMP  quit
559b 4cdd52 348: .a:      jmp  top
559e      349:
559e 20675b 350: LAST1:      JSR  GETSTAT
55a1 f006 351:      BEQ  LUST1
55a3 200a5b 352:      JSR  bsa
55a6 4cdd52 353:      jmp  top
55a9      354:
55a9 a957 355: LUST1:      LDA  #'W'
55ab 8d0203 356:      STA  DCOMND
55ae a901 357:      LDA  #1          ; M
55b0 8d0a03 358:      STA  DAUX1      ; A
55b3 a220 359:      LDX  #$20      ; R
55b5 bd4803 360:      LDA  ICBLL,X    ; B
55b8 8d0b03 361:      STA  DAUX2
55bb 204f5b 362:      JSR  GOSIO
55be      363:
55be      364:
55be      365: ; ***** HANDLE ERRORS *****
55be 20e55b 366:      jsr  printsi
55c1 9c2020 367:      dc.b  156,"      File Transfe
55ea 4cdd52 368:      jmp  top
55ed      369: ;
55ed 20675b 370: GMORE:      JSR  GETSTAT
55f0 f029 371:      BEQ  GMIRE
55f2 200a5b 372:      jsr  bsa
55f5      373: .a:
55f5 20e55b 374:      jsr  printsi
55f8 204573 375:      dc.b  " Escape key exits this pro
5618 4cdd52 376:      JMP  top
561b      377:
561b a957 378: GMIRE:      LDA  #'W'
561d 8d0203 379:      STA  DCOMND
5620 a900 380:      LDA  #0          ; M
5622 8d0a03 381:      STA  DAUX1
5625 204f5b 382:      JSR  GOSIO
5628      383:
5628      384: ; ***** HANDLE ERRORS *****
5628      385:
5628 4c0b55 386:      JMP  PCOK          ; G
562b      387:
562b      388:

```

```

562b      389: QUIT:
562b a210 390:      ldx  #disk1
562d 20195e 391:      jsr  close
5630 a220 392:      ldx  #disk2
5632 20195e 393:      jsr  close
5635 60   394:      RTS              ; B
5636      395:
5636      396:
5636      397:
5636      398: ;
5636      399:
5636 4b3a9b 400: KSPEC:      dc.b  "K:",$9B
5639 01   401: RETRY:      dc.b  1
563a 01   402: TRIES:      dc.b  1          ; U
563b 01   403: i.in_local:  dc.b  1
563c      404:
563c      405: i.parm_1:      ds.b  255
573b      406: i.parm_2:      ds.b  255
583a      407: i.in_str:      ds.b  255
5939      408: i.instr:      ds.b  80
5989      409:
5989      410: ; PLACE FOR RECORD:
5989      411:
5989      412: BUFFER:      ds.b  128
5a09      413:
5a09      414: ;   ASSERT LOW BUFFER <> $FF
5a09      415: ;   ASSERT *-1 <> $FF
5a09      416:
5a09 443a 417: PREFS:      dc.b  "D:"          ; D
5a0b      418:
5a0b      419:
5a0b      420: FSPC:      ds.b  255          ; F
5b0a      421:
5b0a      422: ;   SUBROUTINES
5b0a      423: ;
5b0a      424: bsa:
5b0a 20e55b 425:      jsr  printsi
5b0d 424144 426:      dc.b  "BAD STATUS from PC. Restar
5b30 60   427:      rts
5b31      428:
5b31      429:
5b31      430: ; SUBROUTINE TO SEE IF KEY HAS BEEN
5b31      431: ; PRESSED AND RETURN WITH CARRY SET
5b31      432: ; IF THE KEY IS 'ESCAPE' KEYBOARD MUST
5b31      433: ; BE OPEN VIA IOCB #1
5b31      434:
5b31 adfc02 435: KEYCK:      LDA  CH
5b34 c9ff 436:      CMP  #$FF
5b36 f009 437:      BEQ  KOKY
5b38      438: ;      ldx  #disk1
5b38 207e5d 439:      JSR  get_key
5b3b c91b 440:      CMP  #27          ; E
5b3d d002 441:      BNE  KOKY

```

```

5b3f 38 442:          SEC
5b40 60 443:          RTS
5b41 18 444: KOKY:      CLC
5b42 60 445:          RTS
5b43    446:
5b43    447:
5b43    448: ; THIS SUB DELAYS # OF JIFFIES IN X REG
5b43    449:
5b43 48 450: JIFFY:      PHA
5b44 a514 451: .JIF1:      LDA  RTCLOCK+2      ; I
5b46 c514 452: .CHECK:      CMP  RTCLOCK+2
5b48 f0fc 453:          BEQ  .CHECK
5b4a ca 454:          DEX
5b4b d0f7 455:          BNE  .JIF1
5b4d 68 456:          PLA
5b4e 60 457:          RTS
5b4f    458:
5b4f    459:
5b4f    460:
5b4f    461: ; SUBROUTINE TO SEND COMMAND TO PC
5b4f    462: ; USER SET COMMAND 'W' OR 'R' TO DCOMND
5b4f    463: ; AND SECTOR # TO DAUX1 & DAUX2.
5b4f    464:
5b4f    465: ; SEND RECORD TO SIO2PC:
5b4f    466:
5b4f    467: GOSIO:
5b4f a90a 468:          LDA  #$A          ; D
5b51 8d0103 469:          STA  $301          ; F
5b54 a980 470:          LDA  #$80
5b56 8d0303 471:          STA  $303
5b59 a989 472:          LDA  #LOW BUFFER
5b5b 8d0403 473:          STA  $304
5b5e a959 474:          LDA  #HIGH BUFFER
5b60 8d0503 475:          STA  $305
5b63 2053e4 476:          JSR  $E453          ; D
5b66 60 477:          RTS
5b67    478:
5b67    479: ; GET STATUS FROM SIO2PC:
5b67    480: ; RETURNS "NOT EQUAL" IF ERROR
5b67    481: ; NOTE: IT IS ASSUMED THAT ALL VERSIONS
5b67    482: ; OF OS PUT STATUS AT 02EA
5b67    483:
5b67    484: GETSTAT:
5b67 a900 485:          LDA  #0
5b69 8dea02 486:          STA  $02EA          ; 1
5b6c a953 487:          LDA  #'S'
5b6e 8d0203 488:          STA  DCOMND
5b71 a90a 489:          LDA  #$A          ; D
5b73 8d0103 490:          STA  $301
5b76 a940 491:          LDA  #$40          ; M
5b78 8d0303 492:          STA  $303
5b7b a904 493:          LDA  #4          ; L
5b7d 8d0803 494:          STA  $308

```

```

5b80 a900 495:          LDA  #0
5b82 8d0903 496:          STA  $309
5b85 2053e4 497:          JSR  $E453
5b88 adea02 498:          LDA  $2EA          ; G
5b8b c910 499:          CMP  #$10          ; $
5b8d d001 500:          bne  saybad        ; a
5b8f 60 501:          RTS              ; N
5b90 502:
5b90 503:
5b90 504: ; A SUBROUTINE TO TELL THE PC THAT A
5b90 505: ; FATAL ERROR HAS OCCURRED AND IT
5b90 506: ; SHOULD ABORT THE PROCESS. DAUX1 = 2
5b90 507: ; OF A SECTOR WRITE TELLS THE PC IT'S
5b90 508: ; ALL OVER. SECTOR DATA IS IRRELEVANT
5b90 509:
5b90 510: SAYBAD:
5b90 511:
5b90 a902 512:          LDA  #2
5b92 8d0a03 513:          STA  DAUX1
5b95 a900 514:          LDA  #0
5b97 8d0b03 515:          STA  DAUX2
5b9a a957 516:          LDA  #'W'
5b9c 8d0203 517:          STA  DCOMND
5b9f 204f5b 518:          JSR  GOSIO
5ba2 60 519:          RTS
5ba3 520:
5ba3 521:
5ba3 522: convert_it:
5ba3 ad3b56 523:          lda  i.in_local
5ba6 c901 524:          cmp  #$01
5ba8 f003 525:          beq  .c
5baa 20c25b 526:          jsr  pull_it
5bad 527: .c:
5bad ac3b57 528:          ldy  i.parm_2
5bb0 529: .b:
5bb0 b93b57 530:          lda  i.parm_2,y
5bb3 c93e 531:          cmp  #'>'
5bb5 d005 532:          bne  .a
5bb7 a95c 533:          lda  #"
5bb9 993b57 534:          sta  i.parm_2,y
5bbc 535: .a:
5bbc 88 536:          dey
5bbd c002 537:          cpy  #2
5bbf b0ef 538:          bcs  .b
5bc1 60 539:          rts
5bc2 540: ;~~~~~
5bc2 541: ;
5bc2 542: ; dx:
5bc2 543: pull_it:
5bc2 a201 544:          ldx  #$01
5bc4 a001 545:          ldy  #$01
5bc6 ad3e57 546:          lda  i.parm_2+3
5bc9 c93a 547:          cmp  #':'

```

```

5bcb f004 548:      beq  .a
5bcd a004 549:      ldy  #$04
5bcf a201 550:      idx  #$01
5bd1 b93b57 551: .a:      lda  i.parm_2,y
5bd4 9d3b57 552:      sta  i.parm_2,x
5bd7 cc3b57 553:      cpy  i.parm_2
5bda f005 554:      beq  .b
5bdc c8 555:      iny
5bdd e8 556:      inx
5bde 4cd15b 557:      jmp  .a
5be1 558: .b:
5be1 8e3b57 559:      stx  i.parm_2
5be4 60 560:      rts
5be5 561:
5be5 562:
5be5 563:
5be5 564: ;
5be5 565: ;-----
5be5 566: ;prints Routine macro
5be5 567: ;this is a slow print to the screen
5be5 568: ;usage jsr prints
5be5 569: ; .byte $9b," string to be printed",$ff
5be5 570: ;-----
5be5 571: ;
5be5 572: prints:
5be5 68 573:      pla
5be6 8df65b 574:      sta  .pstr+1
5be9 68 575:      pla
5bea 8df75b 576:      sta  .pstr+2
5bed eef65b 577: .prsl:  inc  .pstr+1
5bf0 d003 578:      bne  .pstr
5bf2 eef75b 579:      inc  .pstr+2
5bf5 adffff 580: .pstr:  lda  $ffff
5bf8 c9ff 581:      cmp  #$ff
5bfa f006 582:      beq  .estri
5bfc 203b5c 583:      jsr  fast_output
5bff 4ced5b 584:      jmp  .prsl
5c02 adf75b 585: .estri:  lda  .pstr+2
5c05 48 586:      pha
5c06 adf65b 587:      lda  .pstr+1
5c09 48 588:      pha
5c0a 60 589:      rts
5c0b 590: ;
5c0b 591: ;
5c0b 592: ;
5c0b 593: ;-----
5c0b 594: ;actual screen handler
5c0b 595: ;screen offset definitions
5c0b 596: ;-----
5c0b 0000 597: scr_offset:  dc.w 0
5c0d 2800 598:      dc.w 40
5c0f 5000 599:      dc.w 80
5c11 7800 600:      dc.w 120

```

```

5c13 a000 601:          dc.w 160
5c15 c800 602:          dc.w 200
5c17 f000 603:          dc.w 240
5c19 1801 604:          dc.w 280
5c1b 4001 605:          dc.w 320
5c1d 6801 606:          dc.w 360
5c1f 9001 607:          dc.w 400
5c21 b801 608:          dc.w 440
5c23 e001 609:          dc.w 480
5c25 0802 610:          dc.w 520
5c27 3002 611:          dc.w 560
5c29 5802 612:          dc.w 600
5c2b 8002 613:          dc.w 640
5c2d a802 614:          dc.w 680
5c2f d002 615:          dc.w 720
5c31 f802 616:          dc.w 760
5c33 2003 617:          dc.w 800
5c35 4803 618:          dc.w 840
5c37 7003 619:          dc.w 880
5c39 9803 620:          dc.w 920
5c3b      621: ;
5c3b      622: ;echo:
5c3b      623: fast_output:
5c3b 855a 624:          sta $5a
5c3d ad0407 625:         lda  sc.redirect
5c40 c902 626:          cmp  #print_p
5c42 f01f 627:         beq  .go_cio
5c44      628: .leo:
5c44 adff02 629:         lda  $02ff      ; CONTROL 1
5c47 c900 630:          cmp  #$00
5c49 d0f9 631:          bne  .leo
5c4b      632:
5c4b a55a 633:          lda  $5a
5c4d c920 634:          cmp  #32
5c4f 9004 635:          bcc  .do_look
5c51 c97d 636:          cmp  #125
5c53 9013 637:          bcc  .do_here
5c55 a000 638: .do_look:      ldy  #0
5c57 b95e5d 639: .lookup:      lda  .char_table,y
5c5a f00c 640:          beq  .do_here
5c5c c55a 641:          cmp  $5a
5c5e f003 642:          beq  .go_cio
5c60 c8 643:          iny
5c61 d0f4 644:          bne  .lookup
5c63      645:
5c63      646:
5c63 a55a 647: .go_cio:      lda  $5a
5c65 4c6e5d 648:          jmp  putlocal
5c68      649:
5c68      650:
5c68 a000 651: .do_here:      ldy  #0
5c6a a55d 652:          lda  $5d
5c6c 915e 653:          sta  ($5e),y

```

```

5c6e a55a 654:      lda  $5a
5c70 c99b 655:      cmp  #$9b
5c72 f044 656:      beq  .docr
5c74 203d5d 657: .notcr:      jsr  .get_adr
5c77 a55a 658:      lda  $5a
5c79 297f 659:      and  #$7f
5c7b c920 660:      cmp  #32
5c7d 9007 661:      bcc  .add64
5c7f c960 662:      cmp  #96
5c81 9009 663:      bcc  .sub32
5c83 4c8f5c 664:      jmp  .asis
5c86 18 665: .add64:      clc
5c87 6940 666:      adc  #64
5c89 4c8f5c 667:      jmp  .asis
5c8c 38 668: .sub32:      sec
5c8d e920 669:      sbc  #32
5c8f 245a 670: .asis:      bit  $5a
5c91 1002 671:      bpl  .xxlate
5c93 0980 672:      ora  #$80
5c95 a000 673: .xxlate:      ldy  #0
5c97 915e 674:      sta  ($5e),y
5c99 e655 675:      inc  $55
5c9b e663 676:      inc  $63
5c9d a555 677:      lda  $55
5c9f c928 678:      cmp  #40
5ca1 b019 679:      bcs  .next_row
5ca3 c8 680:      iny
5ca4 b15e 681:      lda  ($5e),y
5ca6 855d 682:      sta  $5d
5ca8 0980 683:      ora  #$80
5caa aef002 684:      ldx  752
5cad d002 685:      bne  .nocurs1
5caf 915e 686:      sta  ($5e),y
5cb1 e65e 687: .nocurs1:      inc  $5e
5cb3 d002 688:      bne  .nooav
5cb5 e65f 689:      inc  $5e+1
5cb7 60 690: .nooav:      rts
5cb8 a900 691: .docr:      lda  #0
5cba 8563 692:      sta  $63
5cbc a552 693: .next_row:      lda  $52
5cbe 8555 694:      sta  $55
5cc0 e654 695:      inc  $54
5cc2 a454 696:      ldy  $54
5cc4 c018 697:      cpy  #24
5cc6 b013 698:      bcs  .scroll
5cc8 203d5d 699:      jsr  .get_adr
5ccb a000 700:      ldy  #0
5ccd b15e 701:      lda  ($5e),y
5ccf 855d 702:      sta  $5d
5cd1 aef002 703:      ldx  752
5cd4 d004 704:      bne  .nocurs2
5cd6 0980 705:      ora  #$80
5cd8 915e 706:      sta  ($5e),y

```

```

5cda 60 707: .nocurs2:      rts
5cdb 708:
5cdb c654 709: .scroll:      dec  $54
5cdd a558 710:      lda  $58
5cdf 8568 711:      sta  $68
5ce1 18 712:      clc
5ce2 6928 713:      adc  #40
5ce4 855e 714:      sta  $5e
5ce6 a559 715:      lda  $58+1
5ce8 8569 716:      sta  $68+1
5cea 6900 717:      adc  #0
5cec 855f 718:      sta  $5e+1
5cee a000 719:      ldy  #0
5cf0 a203 720:      ldx  #3
5cf2 b15e 721: .sloop:      lda  ($5e),y
5cf4 9168 722:      sta  ($68),y
5cf6 c8 723:      iny
5cf7 d0f9 724:      bne  .sloop
5cf9 e65f 725:      inc  $5e+1
5cfb e669 726:      inc  $68+1
5cfd ca 727:      dex
5cfe d0f2 728:      bne  .sloop
5d00 b15e 729: .sloop2:      lda  ($5e),y
5d02 9168 730:      sta  ($68),y
5d04 c8 731:      iny
5d05 c098 732:      cpy  #152
5d07 90f7 733:      bcc  .sloop2
5d09 a558 734:      lda  $58
5d0b 18 735:      clc
5d0c 6998 736:      adc  #920&$ff
5d0e 855e 737:      sta  $5e
5d10 a559 738:      lda  $58+1
5d12 6903 739:      adc  #920/256
5d14 855f 740:      sta  $5e+1
5d16 a000 741:      ldy  #0
5d18 98 742:      tya
5d19 915e 743: .clear:      sta  ($5e),y
5d1b c8 744:      iny
5d1c c028 745:      cpy  #40
5d1e 90f9 746:      bcc  .clear
5d20 aef002 747:      ldx  752
5d23 d006 748:      bne  .nocurs3
5d25 a980 749:      lda  #$80
5d27 a455 750:      ldy  $55
5d29 915e 751:      sta  ($5e),y
5d2b a900 752: .nocurs3:      lda  #0
5d2d 855d 753:      sta  $5d
5d2f a55e 754:      lda  $5e
5d31 18 755:      clc
5d32 6555 756:      adc  $55
5d34 855e 757:      sta  $5e
5d36 a55f 758:      lda  $5e+1
5d38 6900 759:      adc  #0

```



```

5d3a 855f 760:      sta  $5e+1
5d3c 60 761:      rts
5d3d a554 762: .get_adr:      lda  $54
5d3f 0a 763:      asl
5d40 a8 764:      tay
5d41 a558 765:      lda  $58
5d43 18 766:      clc
5d44 790b5c 767:      adc  scr_offset,y
5d47 855e 768:      sta  $5e
5d49 a559 769:      lda  $58+1
5d4b 790c5c 770:      adc  scr_offset+1,y
5d4e 855f 771:      sta  $5e+1
5d50 a55e 772:      lda  $5e
5d52 18 773:      clc
5d53 6555 774:      adc  $55
5d55 855e 775:      sta  $5e
5d57 a55f 776:      lda  $5e+1
5d59 6900 777:      adc  #0
5d5b 855f 778:      sta  $5e+1
5d5d 60 779:      rts
5d5e 780: .char_table:
5d5e 1b1c1d 781:      dc.b  27,28,29,30,31,125,126,127
5d66 9c9d9e 782:      dc.b  156,157,158,159,253,254,255
5d6e 783:
5d6e 784:
5d6e a20b 785: putlocal:      ldx  #11
5d70 8e4203 786:      stx  $0342
5d73 a200 787:      ldx  #0
5d75 8e4803 788:      stx  $0348
5d78 8e4903 789:      stx  $0349
5d7b 4c56e4 790:      jmp  $e456
5d7e 791: ;
5d7e 792:
5d7e 793:
5d7e 794: ;-----
5d7e 795: ; GET KEY ROUTEEN
5d7e 796: ;this routeen get's one key and
5d7e 797: ;return it to the accumulator
5d7e 798: ;-----
5d7e 799: ;
5d7e 86b6 800: get_key:      stx  $b6
5d80 84b7 801:      sty  $b7
5d82 208a5d 802:      jsr  g_fff
5d85 a6b6 803:      ldx  $b6
5d87 a4b7 804:      ldy  $b7
5d89 60 805:      rts
5d8a ad25e4 806: g_fff:      lda  $e420+5
5d8d 48 807:      pha
5d8e ad24e4 808:      lda  $e420+4
5d91 48 809:      pha
5d92 60 810:      rts
5d93 811: ;
5d93 812:

```

```

5d93      813:
5d93      814:
5d93 20e55b 815: cr_lf:      jsr  printsi
5d96 9bff  816:      dc.b  $9b,-1
5d98 60    817:      rts
5d99      818:
5d99      819: ; prints: prints string with address in PR_PTR, length by
5d99      820: ; ~~~~~
5d99      821:
5d99 a080  822: prints:      ldy  #$80
5d9b d002  823:      bne  pri2
5d9d a000  824: echos:      ldy  #0
5d9f      825:
5d9f 8ccd5d 826: pri2:      sty  pr_mode
5da2 a000  827:      ldy  #0
5da4 b188  828:      lda  (pr_ptr),y      ;ge
5da6 f022  829:      beq  .xpri          ;ex
5da8 8dcc5d 830:      sta  .maxlth
5dab c8    831:      iny
5dac 8ccb5d 832:      sty  .tempy
5daf      833:
5daf accb5d 834: .pril:      ldy  .tempy
5db2 b188  835:      lda  (pr_ptr),y
5db4 2ccd5d 836:      bit  pr_mode
5db7 3006  837:      bmi  .prloc
5db9 203b5c 838:      jsr  fast_output
5dbc 4cc25d 839:      jmp  .prnext
5dbf 203b5c 840: .prloc:      jsr  fast_output
5dc2 eecb5d 841: .prnext:      inc  .tempy
5dc5 cecc5d 842:      dec  .maxlth
5dc8 d0e5  843:      bne  .pril
5dca 60    844: .xpri:      rts
5dcb      845:
5dcb 01    846: .tempy:      dc.b  1
5dcc 01    847: .maxlth:      dc.b  1
5dcd      848:
5dcd 00    849: pr_mode:      dc.b  0
5dce      850:
5dce      851:
5dce      852:
5dce a904  853: open_in:      lda  #4
5dd0 4cec5d 854:      jmp  do_open
5dd3      855:
5dd3 a908  856: open_out:      lda  #8
5dd5 4cec5d 857:      jmp  do_open
5dd8      858:
5dd8 a909  859: open_app:      lda  #9
5dda 4cec5d 860:      jmp  do_open
5ddd      861:
5ddd a90c  862: open_updt:      lda  #12
5ddf 4cec5d 863:      jmp  do_open
5de2      864:
5de2 a906  865: open_dir:      lda  #6

```

```

5de4 4cec5d 866:          jmp  do_open
5de7      867:
5de7 a914 868: open_raw:      lda  #20          ;op
5de9 4cec5d 869:          jmp  do_open
5dec      870:
5dec      871: do_open
5dec 48 872:          pha          ;op
5ded 8a 873:          txa          ;
5dee 48 874:          pha
5def 20195e 875:         jsr  close      ;ma
5df2 68 876:          pla
5df3 aa 877:          tax
5df4 68 878:          pla
5df5 9d4a03 879:         sta  icaux1,x
5df8 a903 880:         lda  #ccopen
5dfa 9d4203 881:         sta  iccom,x
5dfd a586 882:         lda  fn_ptr
5dff 9d4403 883:         sta  icbal,x
5e02 a587 884:         lda  fn_ptr+1
5e04 9d4503 885:         sta  icbah,x
5e07 a950 886:         lda  #80
5e09 9d4803 887:         sta  icbll,x
5e0c a900 888:         lda  #0
5e0e 9d4903 889:         sta  icblh,x
5e11 a980 890:         lda  #128       ;fo
5e13 9d4b03 891:         sta  icaux2,x
5e16 4c56e4 892:         jmp  ciov
5e19      893:
5e19      894: close:
5e19 a90c 895:         LDA  #$0C
5e1b 9d4203 896:         STA  ICCOM,X
5e1e 2056e4 897:         JSR  CIOV
5e21 60 898:         rts
5e22      899:
5e22      900:
5e22 c953 901: get_string:      cmp  #83          ;ma
5e24 9002 902:         bcc  .gsentry      ;ye
5e26 a950 903:         lda  #80
5e28 8d535f 904: .gsentry:      sta  .maxlth      ;sa
5e2b 8e545f 905:         stx  .mask
5e2e 8c555f 906:         sty  .return
5e31      907:
5e31 a900 908:         lda  #0
5e33 854d 909:         sta  attract
5e35      910:
5e35      911:
5e35 a580 912:         lda  in_ptr
5e37 8d4d5f 913:         sta  .address      ;ho
5e3a a581 914:         lda  in_ptr+1      ; t
5e3c 8d4e5f 915:         sta  .address+1
5e3f a900 916:         lda  #0
5e41 8d575f 917:         sta  .i.instr
5e44      918:

```

```

5e44 207e5d 919: .gsloop:      jsr  get_key      ;ge
5e47      920: ;          cmp  #"          ;ca
5e47      921: ;          beq  .gsloop      ;ye
5e47 c99b  922:          cmp  #$9b      ;ca
5e49 d003  923:          bne  .nogs      ;no
5e4b 4cfa5e 924:          jmp  .gscr      ;ye
5e4e c90e  925: .nogs:      cmp  #14      ;tr
5e50 d003  926:          bne  .not14
5e52 4cfa5e 927:          jmp  .gscr
5e55 c97e  928: .not14:      cmp  #126
5e57 f079  929:          beq  .bspace
5e59 c91e  930:          cmp  #30
5e5b f075  931:          beq  .bspace
5e5d c918  932:          cmp  #24      ;^X
5e5f d003  933:          bne  .nowipe
5e61 4ce75e 934:          jmp  .wipeout
5e64 297f  935: .nowipe:      and  #$7f      ;st
5e66 8d565f 936:          sta  .key      ;ho
5e69      937:
5e69      938: ;      now, see if it falls within requested mask
5e69      939: ;      ~~~~~
5e69      940:
5e69      941:
5e69 ae545f 942:          ldx  .mask      ;ge
5e6c f004  943:          beq  .alphanum
5e6e e001  944:          cpx  #1
5e70 f00b  945:          beq  .numeric
5e72      946:
5e72 c920  947: .alphanum:      cmp  #32      ; <
5e74 90ce  948:          bcc  .gsloop      ;ye
5e76 c97b  949:          cmp  #'z'+1      ; >
5e78 b0ca  950:          bcs  .gsloop      ;ye
5e7a 4c885e 951:          jmp  .gsstore
5e7d      952:
5e7d c930  953: .numeric:      cmp  #'0'      ; <
5e7f 90c3  954:          bcc  .gsloop      ;ye
5e81 c93a  955:          cmp  #'9'+1      ; >
5e83 b0bf  956:          bcs  .gsloop      ;ye
5e85 4c885e 957:          jmp  .gsstore
5e88      958:
5e88 ad575f 959: .gsstore:      lda  .i.instr      ;
5e8b cd535f 960:          cmp  .maxlth
5e8e f0b4  961:          beq  .gsloop      ;ig
5e90 b0b2  962:          bcs  .gsloop      ; m
5e92 a8  963:          tay
5e93 c8  964:          iny
5e94 ad565f 965:          lda  .key
5e97 c920  966:          cmp  #32      ;is
5e99 d007  967:          bne  .notspc      ;no
5e9b c001  968:          cpy  #1      ;ls
5e9d d003  969:          bne  .notspc
5e9f 4c445e 970:          jmp  .gsloop      ;ye
5ea2 99575f 971: .notspc:      sta  .i.instr,y      ;sa

```

```

5ea5 ee575f 972:      inc    .i.instr      ;in
5ea8 2c0860 973:      bit    hide           ;hi
5eab 1004 974:      bpl    .nohide
5ead a92a 975:      lda    #'*'
5eaf d003 976:      bne    .showit
5eb1 ad565f 977: .nohide:      lda    .key
5eb4 203b5c 978: .showit:     jsr    fast_output
5eb7 ad535f 979:      lda    .maxlth        ;on
5eba c901 980:      cmp    #1
5ebc d011 981:      bne    .nohot         ;no
5ebe ad575f 982:      lda    .i.instr
5ec1 c901 983:      cmp    #1             ;1
5ec3 d00a 984:      bne    .nohot
5ec5 ad0960 985:      lda    u_flags
5ec8 2908 986:      and    #f_hotkey      ;us
5eca f003 987:      beq    .nohot         ;no
5ecc 4cfa5e 988:      jmp    .gscr          ;ye
5ecf 4c445e 989: .nohot:      jmp    .gsloop
5ed2      990:
5ed2 ad575f 991: .bspace:     lda    .i.instr      ;ca
5ed5 c900 992:      cmp    #0             ;i
5ed7 d003 993:      bne    .nt0
5ed9 4c445e 994:      jmp    .gsloop
5edc ce575f 995: .nt0:      dec    .i.instr      ;-1
5edf a97e 996:      lda    #126           ;ou
5ee1 203b5c 997:      jsr    fast_output
5ee4 4c445e 998:      jmp    .gsloop
5ee7      999:
5ee7 ad575f 1000: .wipeout:    lda    .i.instr      ;ca
5eea d003 1001:      bne    .wt0
5eec 4c445e 1002:      jmp    .gsloop
5eef ce575f 1003: .wt0:      dec    .i.instr      ;-1
5ef2 a97e 1004:      lda    #126           ;ou
5ef4 203b5c 1005:      jsr    fast_output
5ef7 4ce75e 1006:      jmp    .wipeout        ;go
5efa      1007:
5efa ad575f 1008: .gscr:      lda    .i.instr      ;ge
5efd d008 1009:      bne    .nonnull        ;it
5eff ae555f 1010:      ldx    .return         ;al
5f02 d003 1011:      bne    .nonnull        ;ye
5f04 4c445e 1012:      jmp    .gsloop         ;no
5f07      1013:
5f07      1014: ;    done, so move our string into where they want it.
5f07      1015: ;    ~~~~~
5f07      1016:
5f07 a582 1017: .nonnull:    lda    s_ptr
5f09 8d4f5f 1018:      sta    .holds
5f0c a583 1019:      lda    s_ptr+1
5f0e 8d505f 1020:      sta    .holds+1
5f11 a584 1021:      lda    d_ptr
5f13 8d515f 1022:      sta    .holdd
5f16 a585 1023:      lda    d_ptr+1
5f18 8d525f 1024:      sta    .holdd+1

```

```

5f1b      1025:      mea    .i.instr,s_ptr
5f1b a957 ----:      lda    #low .i.instr
5f1d 8582 ----:      sta    s_ptr
5f1f a95f ----:      lda    #high .i.instr
5f21 8583 ----:      sta    s_ptr + 1
5f23 ad4d5f 1026:      lda    .address
5f26 8584 1027:      sta    d_ptr
5f28 ad4e5f 1028:      lda    .address+1
5f2b 8585 1029:      sta    d_ptr+1
5f2d 20f65f 1030:      jsr    move_str      ;mo
5f30 ad4f5f 1031:      lda    .holds
5f33 8582 1032:      sta    s_ptr
5f35 ad505f 1033:      lda    .holds+1
5f38 8583 1034:      sta    s_ptr+1
5f3a ad515f 1035:      lda    .holdd
5f3d 8584 1036:      sta    d_ptr
5f3f ad525f 1037:      lda    .holdd+1
5f42 8585 1038:      sta    d_ptr+1
5f44 a900 1039:      lda    #0
5f46 8d0860 1040:      sta    hide
5f49 ad575f 1041:      lda    .i.instr      ;re
5f4c 60 1042:      rts
5f4d      1043:
5f4d      1044:
5f4d      1045: .address:      ds.b    2
5f4f      1046: .holds:          ds.b    2
5f51      1047: .holdd:          ds.b    2
5f53      1048: .maxlth:        ds.b    1
5f54      1049: .mask:          ds.b    1
5f55      1050: .return:        ds.b    1
5f56      1051: .key:           ds.b    1
5f57      1052: .i.instr:       ds.b    120
5fcf      1053:
5fcf      1054:
5fcf      1055:
5fcf      1056: ;   Make an ASCII ($9b terminated) string out of a lth-b
5fcf      1057: ;   ~~~~~
5fcf      1058:
5fcf a000 1059: toascii:      ldy    #0
5fd1 b182 1060:      lda    (s_ptr),y      ;ge
5fd3 8df55f 1061:      sta    .lth
5fd6 f019 1062:      beq    .null
5fd8 c8 1063:      iny
5fd9 eef55f 1064:      inc    .lth
5fdc b182 1065: .loop:      lda    (s_ptr),y
5fde 88 1066:      dey
5fdf 9184 1067:      sta    (d_ptr),y
5fe1 c8 1068:      iny
5fe2 c8 1069:      iny
5fe3 ccf55f 1070:      cpy    .lth
5fe6 d0f4 1071:      bne    .loop
5fe8 acf55f 1072: .done:      ldy    .lth
5feb 88 1073:      dey

```

```

5fec a99b 1074: .rtn:      lda  #$9b
5fee 9184 1075:          sta  (d_ptr),y
5ff0 60   1076:          rts
5ff1 a000 1077: .null:      ldy  #0
5ff3 f0f7 1078:          beq  .rtn
5ff5      1079:
5ff5 01   1080: .lth:      dc.b  1
5ff6      1081:
5ff6      1082: ;   Move string pointed to by s_ptr to string pointed
5ff6      1083: ;   ~~~~~
5ff6      1084:
5ff6      1085:
5ff6 a000 1086: move_str:   ldy  #0
5ff8 b182 1087:          lda  (s_ptr),y      ;ge
5ffa a8   1088:          tay                ;go
5ffb b182 1089: .loop:      lda  (s_ptr),y
5ffd 9184 1090:          sta  (d_ptr),y
5fff c000 1091:          cpy  #0
6001 f004 1092:          beq  .done
6003 88   1093:          dey
6004 4cfb5f 1094:          jmp  .loop
6007 60   1095: .done:      rts
6008      1096:
6008      1097:
6008      1098:
6008      1099: hide:      ds.b  1          ;PR
6009      1100: u_flags:   ds.b  2          ; 1
600b      1101: rc:        ds.b  1          ;re
600c      1102:
600c      1103:
600c      1104: win_end:   ds.b  0
600c      1105: ;
600c      1106: ;
600c      1107:
600c      1108:
600c      1109:
600c      1110:
600c      1111:
600c      1112:
600c      1113:

```

End assembly: no errors