



```

0000 1: ;-----
0000 2: ;      Copyright 2010 Integrated Logic Systems
0000 3: ;      Source Code is Copyright Stephen J. Car
0000 4: ;
0000 5: ;
0000 6: ;      Please do not share this source!
0000 7: ;-----
0000 8: ;      Peek.com
0000 9: ;
0000 10: ;      This is a program to peak a byte and display it to
0000 11: ;      init 2/24/2010   sjc new design
0000 12: ;      added hex delimiter      2/25/2010
0000 13: ;      added new code for registered user 2/26/201
0000 14: ;
0000 15: ;
0000 16: ;Also added code for Control+1 pause of screen to the high
0000 17: ;handling.      8/24/2012   sjc
0000 18: ;
0000 19: ;
0000 20: ;
0000 21: ;Added code so this file cannot run on the AtariWin800 Emu
0000 22: ;      8/24/2012   sjc
0000 23: ;
0000 24: ;
0000 25: ;      ~~~~~
0000 26: ;-----
0000 27: ; Notes: o This source code MAY NOT be placed for download
0000 28: ;      o "mea" is a macro that loads the address of the
0000 29: ;      into the pointer specified by the second field.
0000 30: ;-----
0000 31: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 32: ;      XASM st and IBM VERSIONS
0000 33: ;-----
0000 34: ;=====
0000 35:
0000 36: com_file_name:      .macro
0000 37: ;      dc.b 13,"filename.ext",$9b
0000 38: ;      dc.b 12,"Shutdown.com",$9b
0000 39: ;      .endm
0000 40:
0000 41: ;
0000 42: ;~~~~~
0000 43: ; File revision history. Version Number in hex
0000 44: ;
0000 45: file_ver:      equ $12
0000 46: ;~~~~~
0000 47: ; the Month in dec for the first time this revision
0000 48: ;was compiled
0000 49: ;
0000 50: c_month:      equ 10
0000 51: ;~~~~~
0000 52: ; the day in dec for the first time this revision
0000 53: ;was compiled

```

```

0000 54: c_day:          equ 13
0000 55: ;~~~~~
0000 56: ; the year in dec for the first time this revision
0000 57: ;was compiled
0000 58: c_year:          equ 2012
0000 59: ;~~~~~
0000 60: ; Type of compiler used in program
0000 61: ;
0000 62: ; 0 = unknown Compiler
0000 63: ; 1 = xasm
0000 64: ; 2 = mac_65
0000 65: ; 3 = basic
0000 66: ; 4 = compiled basic xl
0000 67: ; 5 = C65
0000 68: ; 6 = Action
0000 69: ;
0000 70: ; We can add more as time goes on!
0000 71: ;
0000 72: ;
0000 73: ;
0000 74: xasm:            equ 1
0000 75: mac_65:          equ 2
0000 76: basic:           equ 3
0000 77: basicxl:         equ 4
0000 78: c65:            equ 5
0000 79: action:          equ 6
0000 80:
0000 81: file_compiler:    equ xasm
0000 82: ;~~~~~
0000 83: ; is this relocatable code ?
0000 84: ; anyother value other than 1 or 2 would be unknown
0000 85: r..yes:          equ 1
0000 86: r..no:           equ 2
0000 87: ;
0000 88: ;~~~~~
0000 89: ; Gotta define if it can be relocated
0000 90: l_relocatable:    equ r..yes
0000 91: ;
0000 92: ;
0000 93: ;~~~~~
0000 94: ; Gotta define if it can be relocated
0000 95: r..crl:          equ $7d
0000 96: r..crlf:         equ $9b
0000 97: r..space:        equ $20
0000 98:
0000 99: l_frstscreenbyte: equ r..space
0000 100: ;~~~~~
0000 101: ; The language the output file is in.
0000 102: ;
0000 103: ;
0000 104: ; 0 = Undefined
0000 105: ; 1 = English
0000 106: ; 2 = German

```

```

0000    107: ;
0000    108: r.language:      equ    1
0000    109: ;~~~~~
0000    110:
0000    111:
0000    112:          .include   equates
0000    113:          .include   globals
0000    114:          .include   macros
0000    115:
4000    116:          .org $4000
4000    117:
4000    118: win_start:
4000    119: header_info:
4000    120:          .include   header
42a7 204d47 121:          jsr    os_off
42aa    122:          mea    tsr,s_ptr
42aa a9f5 ----:          lda    #low tsr
42ac 8582 ----:          sta    s_ptr
42ae a9ff ----:          lda    #high tsr
42b0 8583 ----:          sta    s_ptr + 1
42b2 a91d 123:          lda    #shutdown
42b4 206347 124:          jsr    seclvl          ; c
42b7 304e 125:          bmi    .aa
42b9 20b445 126:          jsr    prints_i
42bc 9b 127:          dc.b    $9b
42bd 20 128:          dc.b    " "
42be bdbdbe 129:          dc.b    <+128>,"==>"
42c1 d3e8f5 130:          dc.b    <+128>,"Shutdown is already
42dc bcbdbd 131:          dc.b    <+128>,"<=="
42df 9b 132:          dc.b    $9b
42e0 20 133:          dc.b    " "
42e1 bdbdbe 134:          dc.b    <+128>,"==>"
42e4 c1ee4 135:          dc.b    <+128>,"And Registered with
42fe bcbdbd 136:          dc.b    <+128>,"<=="
4301 9bff 137:          dc.b    $9b,-1
4303 205847 138:          jsr    os_on
4306 60 139:          rts          ; w
4307    140:
4307    141:
4307    142: .aa:
4307 205847 143:          jsr    os_on
430a 200648 144:          jsr    set_handler
430d    145:
430d    146:
430d ade702 147:          lda    memlo          ; s
4310 8da745 148:          sta    segtab+dorgadr  ; P
4313 8dad45 149:          sta    segtab+blkdes   ; P
4316 8db145 150:          sta    segtab+14
4319 ade802 151:          lda    memlo+1        ; s
431c 8da845 152:          sta    segtab+dorgadr+1 ; P
431f 8dae45 153:          sta    segtab+blkdes+1 ; P
4322 8db245 154:          sta    segtab+15
4325    155:

```

```

4325 a900 156:          lda  #seg_on
4327 8db345 157:          sta  segtab+16          ; y
432a      158:
432a 206844 159:          jsr  rlocate
432d      160:
432d ad2202 161:          lda  $0222
4330 8d7a43 162: w25:          sta  brap+1
4333 ad2302 163:          lda  $0222+1
4336 8d7b43 164: w26:          sta  brap+2
4339      165: ;
4339      166: top1:
4339 a03f 167:          ldy  #63
433b b10a 168: mux_1a:          lda  ($0a),y
433d c920 169:          cmp  #$20
433f f00b 170:          beq  mux_2a
4341 c99b 171:          cmp  #$9b
4343 f004 172:          beq  mux_3a
4345 c8 173:          iny
4346 4c3b43 174:          jmp  mux_1a
4349      175: mux_3a:
4349 4c1a48 176:          jmp  mux_3b
434c      177: mux_2a:
434c a200 178:          ldx  #$00
434e c8 179:          iny
434f b10a 180: mux_2b:          lda  ($0a),y
4351 9d4044 181: w27:          sta  filename,x
4354 c99b 182:          cmp  #$9b
4356 f005 183:          beq  mux_2c
4358 e8 184:          inx
4359 c8 185:          iny
435a 4c4f43 186:          jmp  mux_2b
435d      187: mux_2c:
435d a99b 188:          lda  #$9b
435f 9d4044 189: w28:          sta  filename,x          ; I
4362 a50c 190:          lda  $0c
4364 8d1144 191: w29:          sta  trap+1
4367 a50d 192:          lda  $0c+1
4369 8d1244 193: w30:          sta  trap+2
436c 20b743 194: w31:          jsr  int_top
436f 60 195:          rts
4370      196:
4370      197: ;
4370      198: ;~~~~~
4370      199: ; Ok the relocation marker is to tell the DOS we are now u
4370      200: ;more data at memlow..
4370      201: ;
4370      202: ;
4370      203: initz:
4370      204:
4370      205: ;
4370      206: vbi_run:
4370 48 207:          pha
4371 ad10d0 208:          lda  $d010

```

```

4374 c900 209:      cmp    #0
4376 f004 210:      beq    fast_down
4378 68 211:        pla
4379      212: brap:
4379 4cffff 213:      jmp    $ffff
437c      214: ;
437c      215: fast_down:
437c ad10d0 216:      lda    $d010
437f c900 217:      cmp    #$00
4381 f0f9 218: w01:      beq    fast_down
4383 a901 219:      lda    #$01
4385 8d3f44 220: w02:      sta    recurse
4388 a210 221:      ldx    #$10
438a 20ae43 222: w03:      jsr    close
438d a220 223:      ldx    #$20
438f 20ae43 224: w04:      jsr    close
4392 a230 225:      ldx    #$30
4394 20ae43 226: w05:      jsr    close
4397 a240 227:      ldx    #$40
4399 20ae43 228: w06:      jsr    close
439c a250 229:      ldx    #$50
439e 20ae43 230: w07:      jsr    close
43a1 a260 231:      ldx    #$60
43a3 20ae43 232: w08:      jsr    close
43a6 a270 233:      ldx    #$70
43a8 20ae43 234: w09:      jsr    close
43ab 4c74e4 235:      jmp    $e474
43ae      236: ;
43ae      237: ;
43ae      238: ;
43ae      239: close:
43ae a90c 240:      lda    #$0c
43b0 9d4203 241:      sta    $0342,x
43b3 2056e4 242:      jsr    $e456
43b6 60 243:      rts
43b7      244: ;
43b7      245: ;
43b7      246: ;
43b7      247: ;
43b7      248: int_top:
43b7 a970 249: lb01:      lda    # low vbi_run
43b9 8d2202 250:      sta    $0222
43bc a943 251: hb01:      lda    # high vbi_run
43be 8d2302 252:      sta    $0222+1
43c1      253:
43c1 a968 254: lb02:      lda    # low zend
43c3 8de702 255:      sta    $02e7
43c6 a944 256: hb02:      lda    # high zend
43c8 8de802 257:      sta    $02e7+1
43cb      258:
43cb a910 259: lb03:      lda    # low trap
43cd 850c 260:      sta    $0c
43cf a944 261: hb03:      lda    # high trap

```

```

43d1 850d 262:          sta  $0c+1
43d3 60   263:          rts
43d4      264: ;
43d4      265: ;
43d4      266: ;-----
43d4      267: ; STRING PROCESSOR
43d4      268: ;-----
43d4      269: ;
43d4      270: bat_load:
43d4 a940 271: lb04:      lda   # low filename
43d6 8ded43 272: w10:      sta   bl+1
43d9 a944 273: hb04:      lda   # high filename
43db 8dee43 274: w11:      sta   bl+2
43de a03f 275:          ldy   #63
43e0 4cec43 276: w12:      jmp   bl
43e3      277: ;
43e3      278: bprsl:
43e3 c8   279:          iny
43e4 eeed43 280: w13:      inc   bl+1
43e7 d003 281:          bne   bl
43e9 eeee43 282: w14:      inc   bl+2
43ec adffff 283: bl:      lda   $ffff
43ef 910a 284:          sta   ($0a),y
43f1 c900 285:          cmp   #0
43f3 f003 286:          beq   bestri
43f5 4ce343 287: w15:      jmp   bprsl
43f8      288: bestri:
43f8 60   289:          rts
43f9      290: ;
43f9      291: ;-----
43f9      292: ; spartados run vector
43f9      293: ;
43f9      294: ;After a srting has been copied into dos's buffer
43f9      295: ;This will execute it just like you typed it in!
43f9      296: ;
43f9      297: ;-----
43f9      298: ;
43f9      299: line_process:
43f9 a900 300:          lda   #$00
43fb a00a 301:          ldy   #10
43fd 910a 302:          sta   ($0a),y
43ff ad01d3 303:          lda   $d301
4402 48   304:          pha
4403 29fe 305:          and   #$fe
4405 8d01d3 306:          sta   $d301
4408 20d2ff 307:          jsr   $ffd2
440b 68   308:          pla
440c 8d01d3 309:          sta   $d301
440f 60   310:          rts
4410      311: ;
4410      312: ;-----
4410      313: ;
4410      314: ;-----

```

```

4410      315: trap:
4410 201044 316: w16:      jsr  trap
4413 ad3f44 317: w17:      lda  recurse
4416 c901  318:      cmp  #$01
4418 f00a  319:      beq  tp_1
441a ad1fd0 320:      lda  $d01f
441d c903  321:      cmp  #3
441f d003  322:      bne  tp_1
4421 4c7c43 323: w18:      jmp  fast_down
4424      324: tp_1:
4424 20b743 325: w19:      jsr  int_top
4427 ad3f44 326: w20:      lda  recurse
442a c901  327:      cmp  #$01
442c f006  328:      beq  trap1
442e a900  329:      lda  #$00
4430 8d3f44 330: w21:      sta  recurse
4433 60     331:      rts
4434      332: trap1:
4434 a900  333:      lda  #$00
4436 8d3f44 334: w22:      sta  recurse
4439 20d443 335: w23:      jsr  bat_load
443c 4cf943 336: w24:      jmp  line_process
443f      337: ;
443f      338: ;
443f      339: ;
443f      340: ;
443f 00     341: recurse:  dc.b  0
4440      342: filename: ds.b  40
4468      343:
4468      344:
4468      345: zend:    ds.b  0
4468      346: ;
4468      347: ;~~~~~
4468      348: ;
4468      349: ;      *** END OF HANDLER ***
4468      350: ;~~~~~
4468      351: ;
4468      352: ; The rest of this relocater Crap Took me a week to f
4468      353: ;
4468      354: ; Relocater: main entry
4468      355: ; -----
4468      356: ; in:
4468      357: ; segtab = table of segment descriptors
4468      358: ; +00 = relocater table address
4468      359: ; +02 = originate address of block
4468      360: ; +04 = destination originate of block
4468      361: ; +06 = address of block
4468      362: ; +08 = number of bytes in block
4468      363: ; +10 = destination address of block
4468      364: ; +12 = Length of segment descriptor
4468      365: ; +14 = this is the word location adjust mem
4468      366: ; +16 = -1 means do not mem move
4468      367: ;

```



```

4468      368: ;
4468      369: ;
4468      370: ;
4468      371: ;      reltab = relocater table
4468      372: ;          list of address of words to adjust      2
4468      373: ;          list of address low bytes to adjust      2
4468      374: ;          list of address high bytes to adjust      3
4468      375: ;          followed by their low bytes
4468      376: ;~~~~~
4468      377: rlocate:
4468      378:
4468 adb145 379:          lda segtab+segaddress      ; s
446b 8da745 380:          sta segtab+dorgadr      ; P
446e 8dad45 381:          sta segtab+blkdes      ; P
4471 adb245 382:          lda segtab+segaddress+1      ; s
4474 8da845 383:          sta segtab+dorgadr+1      ; P
4477 8dae45 384:          sta segtab+blkdes+1      ; P
447a      385:
447a      386:
447a a900 387:          lda #0                      ; s
447c 8d4745 388:          sta segment
447f      389:
447f ae4745 390: segloop:      ldx segment
4482 bda345 391:          lda segtab+rettad,x      ; g
4485 8d2d45 392:          sta relget+1
4488 bda445 393:          lda segtab+rettad+1,x
448b 8d2e45 394:          sta relget+2
448e 0d2d45 395:          ora relget+1              ; i
4491 d001 396:          bne havseg
4493 60 397:          rts
4494      398:
4494 38 399: havseg:          sec
4495 bda745 400:          lda segtab+dorgadr,x
4498 fda545 401:          sbc segtab+orgadr,x
449b 8d4845 402:          sta zoffset
449e bda845 403:          lda segtab+dorgadr+1,x
44a1 fda645 404:          sbc segtab+orgadr+1,x
44a4 8d4945 405:          sta zoffset+1
44a7      406:
44a7 203845 407: zwordlp:      jsr getzwp
44aa f013 408:          beq zlbytlp              ; i
44ac      409:
44ac b1d7 410:          lda (zwptr),y            ; a
44ae 18 411:          clc
44af 6d4845 412:          adc zoffset
44b2 91d7 413:          sta (zwptr),y
44b4 c8 414:          iny
44b5 b1d7 415:          lda (zwptr),y
44b7 6d4945 416:          adc zoffset+1
44ba 91d7 417:          sta (zwptr),y
44bc 4ca744 418:          jmp zwordlp
44bf      419:
44bf 203845 420: zlbytlp:      jsr getzwp              ; g

```

```

44c2 f00b 421:      beq  zhbytlp      ; i
44c4      422:
44c4 b1d7 423:      lda  (zwptr),y
44c6 18 424:      clc
44c7 6d4845 425:      adc  zoffset
44ca 91d7 426:      sta  (zwptr),y
44cc 4cbf44 427:      jmp  zlbytlp
44cf      428:
44cf 203845 429: zhbytlp:      jsr  getzwp      ; g
44d2 f012 430:      beq  zmovlp      ; j
44d4      431:
44d4 202c45 432:      jsr  relget      ; g
44d7 18 433:      clc
44d8 6d4845 434:      adc  zoffset
44db b1d7 435:      lda  (zwptr),y
44dd 6d4945 436:      adc  zoffset+1
44e0 91d7 437:      sta  (zwptr),y
44e2 4ccf44 438:      jmp  zhbytlp
44e5 60 439:      rts
44e6      440:
44e6      441: zmovlp:
44e6 ad5645 442:      lda  segment+segmove      ; g
44e9 c9ff 443:      cmp  #seg_off      ; $
44eb d001 444:      bne  .zmovlp      ; m
44ed 60 445:      rts      ; o
44ee ae4745 446: .zmovlp:      ldx  segment      ; g
44f1 bda945 447:      lda  segtab+blkadr,x
44f4 8d0f45 448:      sta  zmovfr+1
44f7 bdaa45 449:      lda  segtab+blkadr+1,x
44fa 8d1045 450:      sta  zmovfr+2
44fd      451:
44fd bdad45 452:      lda  segtab+blkdes,x
4500 8d1245 453:      sta  zmovto+1
4503 bdae45 454:      lda  segtab+blkdes+1,x
4506 8d1345 455:      sta  zmovto+2
4509      456:
4509      457:      ; g
4509 bcac45 458:      ldy  segtab+blkbyt+1,x
450c      459:
450c a200 460:      ldx  #0
450e bdffff 461: zmovfr:      lda  $ffff,x
4511 9dffff 462: zmovto:      sta  $ffff,x
4514 e8 463:      inx
4515 d0f7 464:      bne  zmovfr
4517 ee1045 465:      inc  zmovfr+2
451a ee1345 466:      inc  zmovto+2
451d 88 467:      dey
451e 10ee 468:      bpl  zmovfr      ; m
4520      469:      ;
4520 ad4745 470:      lda  segment
4523 18 471:      clc
4524 690c 472:      adc  #seglen      ; g
4526 8d4745 473:      sta  segment

```

```

4529 4c7f44 474:          jmp  segloop
452c      475:
452c      476:
452c adffff 477: relget:          lda  $ffff
452f ee2d45 478:          inc  relget+1
4532 d003  479:          bne  nc1
4534 ee2e45 480:          inc  relget+2
4537 60    481: nc1:      rts
4538      482:
4538 202c45 483: getzwp:          jsr  relget
453b 85d7  484:          sta  zwptr          ; s
453d 202c45 485:          jsr  relget
4540 a000  486:          ldy  #0
4542 85d8  487:          sta  zwptr+1
4544 05d7  488:          ora  zwptr          ; c
4546 60    489:          rts
4547      490:
4547 00    491: segment:          dc.b  0          ; c
4548 0000  492: zoffset:          dc.w  0          ; o
454a      493: ;
454a      494:
454a      495:
454a      496: ;~~~~~
454a      497: ;
454a      498: ;          Relocation data table
454a      499: ;          -----
454a      500: ;          We resolve Word locations First!
454a      501: ;
454a      502: ;  Things like  Lda $5000
454a      503: ;              Sta $5000
454a      504: ;              lda $5000,y
454a      505: ;              jsr $5000
454a      506: ;              jmp $5000
454a      507: ;
454a      508: ;  Well you get the Idea..
454a      509: ;
454a      510: ;~~~~~
454a      511: ;
454a      512: ;
454a      513: rtable:
454a 8243  514:          dc.w  w01+1
454c 8643  515:          dc.w  w02+1
454e 8b43  516:          dc.w  w03+1
4550 9043  517:          dc.w  w04+1
4552 9543  518:          dc.w  w05+1
4554 9a43  519:          dc.w  w06+1
4556 9f43  520:          dc.w  w07+1
4558 a443  521:          dc.w  w08+1
455a a943  522:          dc.w  w09+1
455c      523:
455c d643  524:          dc.w  w10
455e dc43  525:          dc.w  w11+1
4560 e143  526:          dc.w  w12+1

```

```

4562 e543 527:      dc.w  w13+1
4564 ea43 528:      dc.w  w14+1
4566 f643 529:      dc.w  w15+1
4568 1144 530:      dc.w  w16+1
456a 1444 531:      dc.w  w17+1
456c 2244 532:      dc.w  w18+1
456e 2544 533:      dc.w  W19+1
4570      534:
4570 2844 535:      dc.w  w20+1
4572 3144 536:      dc.w  w21+1
4574 3744 537:      dc.w  w22+1
4576 3a44 538:      dc.w  w23+1
4578 3d44 539:      dc.w  w24+1
457a 3143 540:      dc.w  w25+1
457c 3743 541:      dc.w  w26+1
457e 5243 542:      dc.w  w27+1
4580 6043 543:      dc.w  w28+1
4582 6543 544:      dc.w  w29+1
4584      545:
4584 6a43 546:      dc.w  w30+1
4586 6d43 547:      dc.w  w31+1
4588 0000 548:      dc.w  0          ; o
458a      549:
458a      550: ;~~~~~
458a      551: ;
458a      552: ;Ok these are low Byte Locations.. only one byte changed!
458a      553: ; OK Resolving Low Byte's First
458a      554: ;
458a b843 555:      dc.w  lb01+1
458c c243 556:      dc.w  lb02+1
458e cc43 557:      dc.w  lb03+1
4590 d543 558:      dc.w  lb04+1
4592 0000 559:      dc.w  0          ; o
4594      560:
4594      561:
4594 bd43 562:      dc.w  hb01+1
4596 70   563:      dc.b  vbi_run
4597      564:
4597 c743 565:      dc.w  Hb02+1
4599 68   566:      dc.b  zend
459a      567:
459a d043 568:      dc.w  hb03+1
459c 10   569:      dc.b  trap
459d      570:
459d da43 571:      dc.w  hb04+1
459f 40   572:      dc.b  filename
45a0      573:
45a0 0000 574:      dc.w  0          ; o
45a2 00   575:      dc.b  0          ; o
45a3      576:
45a3      577:
45a3      578: ;~~~~~
45a3      579:

```

```

45a3      580: ;End of relocation Table!!!
45a3      581: ;master table:
45a3      582: ;
45a3      583: segtab:
45a3 4a45  584:          dc.w  rtable          ; a
45a5 7043  585:          dc.w  initz           ; a
45a7 0000  586:          dc.w  0               ; r
45a9 7043  587:          dc.w  initz           ; a
45ab f800  588:          dc.w  zend-initz      ; s
45ad 0000  589:          dc.w  0               ; d
45af 0000  590:          dc.w  0               ; E
45b1 0030  591:          dc.w  $3000          ; s
45b3      592:                      ; i
45b3      593:                      ; m
45b3 ff    594:          dc.b  seg_off        ; -
45b4      595:
45b4      596:
45b4      597:
45b4      598:
45b4      599:
45b4      600: ;-----
45b4      601: ; jsr printsi
45b4      602: ;this prints an inline string
45b4      603: ; terminated by $ff
45b4      604: ;-----
45b4      605: ; jsr prints
45b4      606: ; .BYTE "print this",$9B,$FF
45b4      607: echosi:
45b4 68    608: printsi:      pla
45b5 8dc545 609:              sta  pstr+1
45b8 68    610:              pla
45b9 8dc645 611:              sta  pstr+2
45bc eec545 612: prsl:        inc  pstr+1
45bf d003  613:              bne  pstr
45c1 eec645 614:              inc  pstr+2
45c4 adffff 615: pstr:        lda  $ffff
45c7 c9ff   616:              cmp  #$ff
45c9 f006   617:              beq  estri
45cb 200a46 618:              jsr  fast_output
45ce 4cbc45 619:              jmp  prsl
45d1 adc645 620: estri:      lda  pstr+2
45d4 48     621:              pha
45d5 adc545 622:              lda  pstr+1
45d8 48     623:              pha
45d9 60     624:              rts
45da      625: ;
45da      626: ;-----
45da      627: ;actual screen handler
45da      628: ;screen offset definitions
45da      629: ;-----
45da 0000  630: scr_offset:   dc.w  0
45dc 2800  631:              dc.w  40
45de 5000  632:              dc.w  80

```

```

45e0 7800 633:          dc.w 120
45e2 a000 634:          dc.w 160
45e4 c800 635:          dc.w 200
45e6 f000 636:          dc.w 240
45e8 1801 637:          dc.w 280
45ea 4001 638:          dc.w 320
45ec 6801 639:          dc.w 360
45ee 9001 640:          dc.w 400
45f0 b801 641:          dc.w 440
45f2 e001 642:          dc.w 480
45f4 0802 643:          dc.w 520
45f6 3002 644:          dc.w 560
45f8 5802 645:          dc.w 600
45fa 8002 646:          dc.w 640
45fc a802 647:          dc.w 680
45fe d002 648:          dc.w 720
4600 f802 649:          dc.w 760
4602 2003 650:          dc.w 800
4604 4803 651:          dc.w 840
4606 7003 652:          dc.w 880
4608 9803 653:          dc.w 920
460a      654: ;
460a      655: fast_output:
460a 855a 656:          sta $5a
460c ad0407 657:         lda  sc.redirect
460f c902 658:          cmp  #print_p
4611 f01f 659:         beq  .go_cio
4613      660: .leo:
4613 adff02 661:         lda  $02ff      ; CONTROL 1
4616 c900 662:          cmp  #$00
4618 d0f9 663:          bne  .leo
461a      664:
461a a55a 665:         lda  $5a
461c c920 666:          cmp  #32
461e 9004 667:          bcc  .do_look
4620 c97d 668:          cmp  #125
4622 9013 669:          bcc  .do_here
4624 a000 670: .do_look:      ldy  #0
4626 b92d47 671: .lookup:      lda  .char_table,y
4629 f00c 672:          beq  .do_here
462b c55a 673:          cmp  $5a
462d f003 674:          beq  .go_cio
462f c8 675:          iny
4630 d0f4 676:          bne  .lookup
4632      677:
4632      678:
4632 a55a 679: .go_cio:      lda  $5a
4634 4c3d47 680:          jmp  putlocal
4637      681:
4637      682:
4637 a000 683: .do_here:      ldy  #0
4639 a55d 684:          lda  $5d
463b 915e 685:          sta  ($5e),y

```

```

463d a55a 686:      lda  $5a
463f c99b 687:      cmp  #$9b
4641 f044 688:      beq  .docr
4643 200c47 689: .notcr:      jsr  .get_adr
4646 a55a 690:      lda  $5a
4648 297f 691:      and  #$7f
464a c920 692:      cmp  #32
464c 9007 693:      bcc  .add64
464e c960 694:      cmp  #96
4650 9009 695:      bcc  .sub32
4652 4c5e46 696:      jmp  .asis
4655 18 697: .add64:      clc
4656 6940 698:      adc  #64
4658 4c5e46 699:      jmp  .asis
465b 38 700: .sub32:      sec
465c e920 701:      sbc  #32
465e 245a 702: .asis:      bit  $5a
4660 1002 703:      bpl  .xxlate
4662 0980 704:      ora  #$80
4664 a000 705: .xxlate:      ldy  #0
4666 915e 706:      sta  ($5e),y
4668 e655 707:      inc  $55
466a e663 708:      inc  $63
466c a555 709:      lda  $55
466e c928 710:      cmp  #40
4670 b019 711:      bcs  .next_row
4672 c8 712:      iny
4673 b15e 713:      lda  ($5e),y
4675 855d 714:      sta  $5d
4677 0980 715:      ora  #$80
4679 aef002 716:      ldx  752
467c d002 717:      bne  .nocurs1
467e 915e 718:      sta  ($5e),y
4680 e65e 719: .nocurs1:      inc  $5e
4682 d002 720:      bne  .nooav
4684 e65f 721:      inc  $5e+1
4686 60 722: .nooav:      rts
4687 a900 723: .docr:      lda  #0
4689 8563 724:      sta  $63
468b a552 725: .next_row:      lda  $52
468d 8555 726:      sta  $55
468f e654 727:      inc  $54
4691 a454 728:      ldy  $54
4693 c018 729:      cpy  #24
4695 b013 730:      bcs  .scroll
4697 200c47 731:      jsr  .get_adr
469a a000 732:      ldy  #0
469c b15e 733:      lda  ($5e),y
469e 855d 734:      sta  $5d
46a0 aef002 735:      ldx  752
46a3 d004 736:      bne  .nocurs2
46a5 0980 737:      ora  #$80
46a7 915e 738:      sta  ($5e),y

```

```

46a9 60 739: .nocurs2:      rts
46aa      740:
46aa c654 741: .scroll:      dec  $54
46ac a558 742:              lda  $58
46ae 8568 743:              sta  $68
46b0 18 744:              clc
46b1 6928 745:              adc  #40
46b3 855e 746:              sta  $5e
46b5 a559 747:              lda  $58+1
46b7 8569 748:              sta  $68+1
46b9 6900 749:              adc  #0
46bb 855f 750:              sta  $5e+1
46bd a000 751:              ldy  #0
46bf a203 752:              ldx  #3
46c1 b15e 753: .sloop:          lda  ($5e),y
46c3 9168 754:              sta  ($68),y
46c5 c8 755:              iny
46c6 d0f9 756:              bne  .sloop
46c8 e65f 757:              inc  $5e+1
46ca e669 758:              inc  $68+1
46cc ca 759:              dex
46cd d0f2 760:              bne  .sloop
46cf b15e 761: .sloop2:         lda  ($5e),y
46d1 9168 762:              sta  ($68),y
46d3 c8 763:              iny
46d4 c098 764:              cpy  #152
46d6 90f7 765:              bcc  .sloop2
46d8 a558 766:              lda  $58
46da 18 767:              clc
46db 6998 768:              adc  #920&$ff
46dd 855e 769:              sta  $5e
46df a559 770:              lda  $58+1
46e1 6903 771:              adc  #920/256
46e3 855f 772:              sta  $5e+1
46e5 a000 773:              ldy  #0
46e7 98 774:              tya
46e8 915e 775: .clear:          sta  ($5e),y
46ea c8 776:              iny
46eb c028 777:              cpy  #40
46ed 90f9 778:              bcc  .clear
46ef aef002 779:              ldx  752
46f2 d006 780:              bne  .nocurs3
46f4 a980 781:              lda  #$80
46f6 a455 782:              ldy  $55
46f8 915e 783:              sta  ($5e),y
46fa a900 784: .nocurs3:         lda  #0
46fc 855d 785:              sta  $5d
46fe a55e 786:              lda  $5e
4700 18 787:              clc
4701 6555 788:              adc  $55
4703 855e 789:              sta  $5e
4705 a55f 790:              lda  $5e+1
4707 6900 791:              adc  #0

```



```

4709 855f 792:          sta  $5e+1
470b 60 793:          rts
470c a554 794: .get_adr:      lda  $54
470e 0a 795:          asl
470f a8 796:          tay
4710 a558 797:          lda  $58
4712 18 798:          clc
4713 79da45 799:          adc  scr_offset,y
4716 855e 800:          sta  $5e
4718 a559 801:          lda  $58+1
471a 79db45 802:          adc  scr_offset+1,y
471d 855f 803:          sta  $5e+1
471f a55e 804:          lda  $5e
4721 18 805:          clc
4722 6555 806:          adc  $55
4724 855e 807:          sta  $5e
4726 a55f 808:          lda  $5e+1
4728 6900 809:          adc  #0
472a 855f 810:          sta  $5e+1
472c 60 811:          rts
472d      812: .char_table:
472d 1b1c1d 813:          dc.b  27,28,29,30,31,125,126,127
4735 9c9d9e 814:          dc.b  156,157,158,159,253,254,255
473d      815:
473d      816:
473d a20b 817: putlocal:      ldx  #11
473f 8e4203 818:          stx  $0342
4742 a200 819:          ldx  #0
4744 8e4803 820:          stx  $0348
4747 8e4903 821:          stx  $0349
474a 4c56e4 822:          jmp  $e456
474d      823: ;
474d      824:
474d      825:
474d      826:
474d      827: ;~~~~~
474d      828: ; OS on and off ROUTEEN
474d      829: ;~~~~~
474d      830: ;
474d      831:
474d      832: os_off:
474d 78 833:          SEI
474e ad01d3 834:          LDA  PORTB
4751 29fe 835:          AND  #$FE
4753 8d01d3 836:          STA  PORTB
4756 58 837:          CLI
4757 60 838:          RTS
4758      839: ;
4758      840: os_on:
4758 78 841:          SEI
4759 ad01d3 842:          LDA  PORTB
475c 0901 843:          ORA  #1
475e 8d01d3 844:          STA  PORTB

```

```

4761 58 845:          CLI
4762 60 846:          RTS
4763      847: ;~~~~~
4763      848: ;checks user security level pointed to by S_PTR to see if
4763      849: ;bit specified in the A register is ON. Returns MINus if
4763      850: ;user doesn't have the specified security level.
4763      851: ;~~~~~
4763      852:
4763 8cbe47 853: seclvl:          sty .tempy
4766 8ebf47 854:          stx .tempx
4769 c900 855:          cmp #0          ;se
476b f042 856:          beq .passed      ;ye
476d a8 857:          tay              ;-1
476e a980 858:          lda #128
4770 8dc047 859:          sta .temp        ;in
4773 a900 860:          lda #0
4775 8dc147 861:          sta .temp+1
4778 8dc247 862:          sta .temp+2
477b 8dc347 863:          sta .temp+3
477e 88 864:          dey
477f f011 865:          beq .mask        ;if
4781 4ec047 866: .shloop:      lsr .temp    ;sh
4784 6ec147 867:          ror .temp+1
4787 6ec247 868:          ror .temp+2
478a 6ec347 869:          ror .temp+3
478d 88 870:          dey
478e d0f1 871:          bne .shloop
4790 a000 872:          ldy #0
4792 b9c047 873: .mask:        lda .temp,y   ;no
4795 3182 874:          and (s_ptr),y
4797 99c047 875:          sta .temp,y
479a c8 876:          iny
479b c004 877:          cpy #4
479d d0f3 878:          bne .mask
479f adc047 879: .check:      lda .temp
47a2 0dc147 880:          ora .temp+1      ;i
47a5 0dc247 881:          ora .temp+2
47a8 0dc347 882:          ora .temp+3
47ab c900 883:          cmp #0
47ad f004 884:          beq .failed
47af a900 885: .passed:     lda #0
47b1 f002 886:          beq .return
47b3 a980 887: .failed:     lda #$80
47b5 08 888: .return:     php
47b6 acbe47 889:          ldy .tempy
47b9 aebf47 890:          ldx .tempx
47bc 28 891:          plp
47bd 60 892:          rts
47be      893:
47be 00 894: .tempy:      dc.b 0
47bf 00 895: .tempx:      dc.b 0
47c0 000000 896: .temp:       dc.b 0,0,0
47c4      897:

```

```

47c4      898:
47c4      899: ;   sets the bit specified (in the A register) of the se
47c4      900: ;   level pointed to by S_PTR to a TRUE value.
47c4      901: ;   ~~~~~
47c4      902:
47c4 8c0048 903: setlvl:          sty  .tempy
47c7 8e0148 904:          stx  .tempx
47ca c900   905:          cmp  #0          ;se
47cc f031   906:          beq  .return      ;ye
47ce a8     907:          tay          ;:-
47cf a980   908:          lda  #128
47d1 8d0248 909:          sta  .temp        ;in
47d4 a900   910:          lda  #0
47d6 8d0348 911:          sta  .temp+1
47d9 8d0448 912:          sta  .temp+2
47dc 8d0548 913:          sta  .temp+3
47df 88     914:          dey
47e0 f011   915:          beq  .mask        ;if
47e2 4e0248 916: .shloop:      lsr  .temp        ;sh
47e5 6e0348 917:          ror  .temp+1
47e8 6e0448 918:          ror  .temp+2
47eb 6e0548 919:          ror  .temp+3
47ee 88     920:          dey
47ef d0f1   921:          bne  .shloop
47f1 a000   922:          ldy  #0
47f3 b90248 923: .mask:        lda  .temp,y        ;no
47f6 1182   924:          ora  (s_ptr),y
47f8 9182   925:          sta  (s_ptr),y
47fa c8     926:          iny
47fb c004   927:          cpy  #4
47fd d0f4   928:          bne  .mask
47ff 60     929: .return:      rts
4800      930:
4800 00     931: .tempy:       dc.b  0
4801 00     932: .tempx:       dc.b  0
4802 000000 933: .temp:        dc.b  0,0,0,0
4806      934:
4806      935: ;   ~~~~~
4806      936: set_handler:
4806 204d47 937:          jsr  os_off
4809      938:          mea  tsr,s_ptr
4809 a9f5   ----:          lda  #low tsr
480b 8582   ----:          sta  s_ptr
480d a9ff   ----:          lda  #high tsr
480f 8583   ----:          sta  s_ptr + 1
4811 a91d   939:          lda  #shutdown
4813 20c447 940:          jsr  setlvl
4816 205847 941:          jsr  os_on
4819 60     942:          rts
481a      943:
481a      944:
481a      945: ;
481a      946: mux_3b:

```

---

```
481a 20b445 947:      jsr  printsi
481d 7d9b9b 948:      dc.b  $7d,$9b,$9b,$9b
4821 204669 949:      dc.b  " File Function Aborted",$9
4839 205379 950:      dc.b  " Syntax error!",$9B
4848 207573 951:      dc.b  " usage = dx:>path>shutdown
4868 207061 952:      dc.b  " parm = dn:>path>filename.
4888 60     953:      rts
4889      954: ;
4889      955: ;
4889      956: win_end:      ds.b  0
4889      957: ;
```

End assembly: no errors