


```

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: ;      Makemsgb.com
0000 9: ;
0000 10: ; This is a program to peak a byte and display it to
0000 11: ; to Make a database structor 05/27/2010 sjc
0000 12: ; added a recursive dir maker 05/29/2010 sjc
0000 13: ;
0000 14: ; ~~~~~
0000 15: ;-----
0000 16: ; Notes: o This source code MAY NOT be placed for download
0000 17: ; o "mea" is a macro that loads the address of the
0000 18: ; into the pointer specified by the second field.
0000 19: ;-----
0000 20: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 21: ; XASM st and IBM VERSIONS
0000 22: ;-----
0000 23:
0000 24: com_file_name: .macro
0000 25: ;      dc.b 13,"filename.ext",$9b
0000 26: ;      dc.b 12,"Makemsgb.com",$9b
0000 27: ;      .endm
0000 28:
0000 29: ;
0000 30: ;~~~~~
0000 31: ; File revision history. Version Number in hex
0000 32: ;
0000 33: file_ver:      equ $16
0000 34: ;~~~~~
0000 35: ; the Month in dec for the first time this revision
0000 36: ;was compiled
0000 37: ;
0000 38: c_month:      equ 8
0000 39: ;~~~~~
0000 40: ; the day in dec for the first time this revision
0000 41: ;was compiled
0000 42: c_day:      equ 28
0000 43: ;~~~~~
0000 44: ; the year in dec for the first time this revision
0000 45: ;was compiled
0000 46: c_year:      equ 2012
0000 47: ;~~~~~
0000 48: ; Type of compiler used in program
0000 49: ;
0000 50: ; 0 = unknown Compiler
0000 51: ; 1 = xasm
0000 52: ; 2 = mac_65
0000 53: ; 3 = basic

```

```

0000      54: ; 4 = compiled basic xl
0000      55: ; 5 = C65
0000      56: ; 6 = Action
0000      57: ;
0000      58: ; We can add more as time goes on!
0000      59: ;
0000      60: ;
0000      61: ;
0000      62: xasm:          equ    1
0000      63: mac_65:         equ    2
0000      64: basic:        equ    3
0000      65: basicxl:      equ    4
0000      66: c65:          equ    5
0000      67: action:        equ    6
0000      68:
0000      69: file_compiler:      equ    xasm
0000      70: ;~~~~~
0000      71: ; is this relocatable code ?
0000      72: ; anyother value other than 1 or 2 would be unknown
0000      73: r..yes:             equ    1
0000      74: r..no:              equ    2
0000      75: ;
0000      76: ;~~~~~
0000      77: ; Gotta define if it can be relocated
0000      78: l_relocatable:      equ    r..no
0000      79: ;
0000      80: ;
0000      81: ;~~~~~
0000      82: ; Gotta define our Header control Byte
0000      83: ;$ff=bypass intro
0000      84: ;$15 = Bypass Dos check
0000      85: r..crl:             equ    $7d
0000      86: r..crlf:            equ    $9b
0000      87: r..space:           equ    $20
0000      88: r..bypassDOS:       equ    $15
0000      89: r..bypass:          equ    $ff
0000      90:
0000      91: l_frstscreenbyte:   equ    $ff
0000      92: ;~~~~~
0000      93: ; The language the output file is in.
0000      94: ;
0000      95: ;
0000      96: ; 0 =    Undefined
0000      97: ; 1 =    English
0000      98: ; 2 =    German
0000      99: ;
0000     100: r..language:        equ    1
0000     101: ;~~~~~
0000     102:
0000     103:
0000     104:             .include    equates
0000     105:             .include    globals
0000     106:             .include    macros

```

```

0000      107:
4000      108:          .org $4000
4000      109:
4000      110: win_start:
4000      111: header_info:
4000      112:          .include    header
42a7      113:
42a7      114: tammy:
42a7 a210  115:          ldx    #disk1
42a9 20904f 116:          jsr    close
42ac a220  117:          ldx    #disk2
42ae 20904f 118:          jsr    close
42b1      119:
42b1 a50a  120:          lda    comtab
42b3 18    121:          CLC
42b4 6903  122:          adc    # low ZCRNAME
42b6 8d5a43 123:          sta    cli.crunch+1
42b9 a50b  124:          lda    comtab+1
42bb 6900  125:          adc    # high ZCRNAME
42bd 8d5b43 126:          sta    cli.crunch+2
42c0 205943 127:          jsr    cli.crunch
42c3 d003  128:          bne    .get_parm1
42c5 4c5c43 129:          jmp    parm_errors
42c8      130:
42c8      131: .get_parm1:
42c8 a50a  132:          lda    comtab
42ca 18    133:          CLC
42cb 6921  134:          adc    # low COMFNAM
42cd 8582  135:          sta    s_ptr
42cf a50b  136:          lda    comtab+1
42d1 6900  137:          adc    # high COMFNAM
42d3 8583  138:          sta    s_ptr+1
42d5      139:          mea    parm_1,d_ptr
42d5 a9c3  ----:          lda    #low parm_1
42d7 8584  ----:          sta    d_ptr
42d9 a944  ----:          lda    #high parm_1
42db 8585  ----:          sta    d_ptr + 1
42dd 208c4c 140:          jsr    tostr
42e0      141:
42e0      142:          mea    parm_1,s_ptr
42e0 a9c3  ----:          lda    #low parm_1
42e2 8582  ----:          sta    s_ptr
42e4 a944  ----:          lda    #high parm_1
42e6 8583  ----:          sta    s_ptr + 1
42e8      143:          mea    filename,d_ptr
42e8 a957  ----:          lda    #low filename
42ea 8584  ----:          sta    d_ptr
42ec a948  ----:          lda    #high filename
42ee 8585  ----:          sta    d_ptr + 1
42f0 20654c 144:          jsr    toascii
42f3 205249 145:          jsr    make_path
42f6      146:
42f6      147:

```

```

42f6 ad1fd0 148:      lda  consol
42f9 c907  149:      cmp  #consol_any
42fb f059  150:      beq  .no_tell      ;be
42fd      151:
42fd 20f94c 152:      jsr  printsi
4300 9b2074 153:      dc.b  $9b," this is my cli to par
431d 9b5061 154:      dc.b  $9b,"Parm_1 =",-1
4327      155:      mea  parm_1,pr_ptr
4327 a9c3  ----:      lda  #low parm_1
4329 8588  ----:      sta  pr_ptr
432b a944  ----:      lda  #high parm_1
432d 8589  ----:      sta  pr_ptr + 1
432f 20c04e 156:      jsr  echos
4332      157:
4332 203f4f 158:      jsr  cr_lf
4335      159:      mea  path1,pr_ptr
4335 a9a8  ----:      lda  #low path1
4337 8588  ----:      sta  pr_ptr
4339 a948  ----:      lda  #high path1
433b 8589  ----:      sta  pr_ptr + 1
433d 20c04e 160:      jsr  echos
4340      161:
4340 20f94c 162:      jsr  printsi
4343 2023ff 163:      dc.b  " #",-1
4346 ada748 164:      lda  totalpath
4349 20d649 165:      jsr  pr_byte
434c      166:
434c 20f94c 167:      jsr  printsi
434f 9b9bff 168:      dc.b  $9b,$9b,-1
4352 202a4f 169:      jsr  get_key
4355 60     170:      rts
4356      171:
4356      172: .no_tell:
4356 4c5e51 173:      jmp  begin
4359      174:
4359 4c0000 175: cli.crunch:      jmp  $0000
435c      176:
435c      177: parm_errors:
435c      178:
435c 20f94c 179:      jsr  printsi
435f 9b9b  180:      dc.b  $9B,$9B
4361 205379 181:      dc.b  " Syntax for use:",$9b
4372 204d61 182:      dc.b  " Makemsgb Dn:>dir>fname.ex
438e 205468 183:      dc.b  " This Program will make", $
43a6 204469 184:      dc.b  " Directorys for you!",$9b
43bb 9b    185:      dc.b  $9b
43bc      186:
43bc 9b    187:      dc.b  $9b
43bd ff    188:      dc.b  -1
43be 60    189:      rts
43bf      190: ;
43bf      191: ;
43bf      192:

```

```

43bf      193: word:          ds.b  30      ;damn backw
43dd 1a00  194: s_node:          dc.w  26      ; n
43df      195:
43df      196: s_today:          ds.b  14      ;today's da
43ed      197: cur_date:          ds.b  15      ; Mon 13-Ma
43fc      198: cur_time:          ds.b  15      ; 11:00:00a
440b      199: cur_month:          ds.b   1      ;1-12
440c      200: cur_day:           ds.b   1      ;1-31
440d      201: cur_year:          ds.b   1      ;1-99
440e      202: cur_hour:          ds.b   1      ;0-23
440f      203: cur_minute:        ds.b   1      ;0-59
4410      204: cur_second:        ds.b   1      ;0-59
4411      205: cur_weekday:       ds.b   1      ;0=sunday,
4412      206: s_msgbdr:          ds.b  32      ;drives eac
4432      207: hi_msglo:          ds.b  32      ;PRIVATE 3
4452      208: hi_msghi:          ds.b  32      ;PRIVATE
4472      209: u_handle:          ds.b  21      ;handle
4487 fffff 210: u_flags:          dc.w  $ffff,$ffff ; 1
448b fffff 211: u_seclvl:         dc.w  $ffff,$ffff ;security 1
448f      212: u_msgread:         ds.b   4      ;read messa
4493      213: u_msgpost:         ds.b   4      ;post messa
4497      214: u_msghedit:        ds.b   4      ;edit any m
449b fffff 215: u_msgdel:         dc.w  $ffff,$ffff ;
449f      216: ;
449f 00    217: in_clock:         dc.b   0      ;set by the
44a0      218: tdbuff:           ds.b  30      ;ti
44be 00    219: queue:           dc.b   0      ;stuff a ke
44bf 00    220: qout:            dc.b   0      ;
44c0 00    221: qin:            dc.b   0      ;
44c1 00    222: rc:             dc.b   0      ; Return Co
44c2 00    223: trans:          dc.b   0      ;0=atascii,
44c3      224:
44c3      225: parm_1:           ds.b  80
4513      226: parm_2:           ds.b  80
4563      227: parm_3:           ds.b  80
45b3      228: in_str:          ds.b  80      ;
4603      229: instr:          ds.b  80      ;
4653      230: read_buff:        ds.b 512      ;PRIVATE ge
4853      231:
4853      232: ;
4853      233: my_count:         ds.b   1
4854      234:
4854 0000  235: tammy_c:          dc.w   0
4856 1a    236: stevel:          dc.b  26
4857      237:
4857      238:
4857      239: filename:         ds.b  80
48a7      240:
48a7      241:
48a7 00    242: totalpath:        dc.b   0
48a8      243: path1:           ds.b 128
4928      244: ;
4928      245: ;~~~~~

```

```

4928 000000 246: f.temp:          dc.b  0,0,0
492b      247: ascii:           ds.b   6
4931 0000   248: num:           dc.w   0
4933      249: char:           ds.b   6
4939 0000   250: counter:       dc.w   0
493b 00     251: count:         dc.b   0
493c 000a   252: counter2:      dc.w  2560
493e 000300 253: mbsize:        dc.w  768,1024,1280,1536,
494e 000000 254: temp2:         dc.b   0,0,0
4951 00     255: stat:          dc.b   0
4952      256:
4952      257: ;~~~~~
4952      258: ;Make Path
4952      259: ;~~~~~
4952      260: make_path:
4952 a200   261:          ldx  #0
4954 a001   262:          ldy  #1
4956 b9c344 263: .a:      lda  parm_1,y
4959 c93e   264:          cmp  #'>'
495b d006   265:          bne  .b
495d a900   266:          lda  #$00          ;'%
495f eea748 267:          inc  totalpath
4962 e8     268:          inx
4963      269:
4963 c92e   270: .b:      cmp  #'.'
4965 f015   271:          beq  .strip
4967 ccc344 272:          cpy  parm_1
496a f010   273:          beq  .strip
496c 99a848 274:          sta  path1,y
496f 8ca848 275:          sty  path1
4972 c8     276:          iny
4973 a99b   277:          lda  #$9b
4975 99a848 278:          sta  path1,y
4978 f0dc   279:          beq  .a
497a d0da   280:          bne  .a
497c      281: .strip:
497c ada748 282:          lda  totalpath
497f c900   283:          cmp  #$00
4981 f00b   284:          beq  .done
4983 aca848 285:          ldy  path1
4986 88     286: .loop:   dey
4987 b9a848 287:          lda  path1,y
498a c900   288:          cmp  #$00          ;'%
498c d0f8   289:          bne  .loop
498e 8ca848 290: .done:   sty  path1
4991 a99b   291:          lda  #$9b
4993 99a848 292:          sta  path1,y
4996 a004   293:          ldy  #4
4998 b9a848 294:          lda  path1,y
499b c900   295:          cmp  #$00          ;'%
499d d00c   296:          bne  .exit
499f a93e   297:          lda  #'>'
49a1 99a848 298:          sta  path1,y

```

```

49a4 aca748 299:          ldy  totalpath
49a7 88      300:          dey
49a8 8ca748 301:          sty  totalpath
49ab        302: .exit:
49ab 60      303:          rts
49ac        304: ;~~~~~
49ac        305: ;next Path
49ac        306: ;~~~~~
49ac        307:
49ac        308: next_path:
49ac ada748 309:          lda  totalpath
49af c900    310:          cmp  #$00
49b1 d001    311:          bne  .next
49b3 60      312:          rts
49b4        313: .next:
49b4 a000    314:          ldy  #$00
49b6 c8      315: .a:          iny
49b7 b9a848 316:          lda  path1,y
49ba cca848 317:          cpy  path1
49bd f011    318:          beq  .done
49bf c900    319:          cmp  #$00          ;'%
49c1 d0f3    320:          bne  .a
49c3 a93e    321:          lda  #'>'
49c5 99a848 322:          sta  path1,y
49c8 aca748 323:          ldy  totalpath
49cb 88      324:          dey
49cc 8ca748 325:          sty  totalpath
49cf 60      326:          rts
49d0        327: .done:
49d0 a900    328:          lda  #$00
49d2 8da748 329:          sta  totalpath
49d5 60      330:          rts
49d6        331:
49d6        332: ;
49d6        333: ;~~~~~
49d6        334: ;Prints a byte or a card to the screen
49d6        335: ;
49d6        336: ;
49d6        337: pr_byte:
49d6 a200    338:          ldx  #$00
49d8        339: pr_card:
49d8 85d4    340:          sta  fr0
49da 86d5    341:          stx  fr0+1
49dc a900    342:          lda  #$00
49de 85f2    343:          sta  cix
49e0 20aad9 344:          jsr  ifp
49e3 20e6d8 345:          jsr  fasc
49e6 a000    346:          ldy  #$00
49e8        347: .aprbloop:
49e8 b1f3    348:          lda  (inbuff),y
49ea 08      349:          php
49eb 297f    350:          and  #$7f
49ed c8      351:          iny

```



```

49ee 20924e 352:          jsr  echo
49f1 28      353:          plp
49f2 10f4    354:          bpl  .aprbloop
49f4 60      355:          rts
49f5        356:
49f5        357:
49f5        358: ;   gets a BYTE value - returned in Accum
49f5        359: ;   ~~~~~
49f5        360:
49f5 202d4a 361: get_byte:      jsr  get_str
49f8 274a    362:          dc.w  .work
49fa 030100 363:          dc.b  3,1,0
49fd a8      364:          tay
49fe c8      365:          iny
49ff a99b    366:          lda  #$9b          ;in
4a01 99274a 367:          sta  .work,y
4a04        368:          mea  .work,inbuff
4a04 a927    ----:      lda  #low .work
4a06 85f3    ----:      sta  inbuff
4a08 a94a    ----:      lda  #high .work
4a0a 85f4    ----:      sta  inbuff + 1
4a0c a901    369:          lda  #1
4a0e 85f2    370:          sta  cix
4a10 2000d8 371:          jsr  afp          ;as
4a13 b00f    372:          bcs  .default
4a15 20d2d9 373:          jsr  fpi          ;fl
4a18 b00a    374:          bcs  .default
4a1a a5d5    375:          lda  fr0+1        ;en
4a1c d003    376:          bne  .ovfl        ;ye
4a1e a5d4    377:          lda  fr0
4a20 60      378:          rts
4a21 a9ff    379: .ovfl:      lda  #255
4a23 60      380:          rts
4a24 a900    381: .default:   lda  #0
4a26 60      382:          rts
4a27        383:
4a27        384: .work:      ds.b  6
4a2d        385:
4a2d        386:
4a2d        387:
4a2d        388: ;   get_str  gets an input line into the specified s
4a2d        389: ;
4a2d        390: ;   input:  address of string in in_ptr
4a2d        391: ;   A - max length
4a2d        392: ;   X - mask (0=alpha/numeric, 1=numeric)
4a2d        393: ;   Y - allow return key only (1=yes)
4a2d        394: ;
4a2d        395: ;   length of inputted string is returned in the 0th pos
4a2d        396: ;   ~~~~~
4a2d        397:
4a2d 68      398: get_str:      pla          ;en
4a2e 8588    399:          sta  pr_ptr
4a30 68      400:          pla

```

```

4a31 8589 401:      sta  pr_ptr+1
4a33 a001 402:      ldy  #1
4a35 b188 403:      lda  (pr_ptr),y      ;ge
4a37 8580 404:      sta  in_ptr
4a39 c8 405:      iny
4a3a b188 406:      lda  (pr_ptr),y      ;
4a3c 8581 407:      sta  in_ptr+1
4a3e c8 408:      iny
4a3f b188 409:      lda  (pr_ptr),y      ;
4a41 8d6b4a 410:     sta  .gsa
4a44 c8 411:      iny
4a45 b188 412:      lda  (pr_ptr),y      ;
4a47 8d6c4a 413:     sta  .gsx
4a4a c8 414:      iny
4a4b b188 415:      lda  (pr_ptr),y      ;
4a4d 8d6d4a 416:     sta  .gsy
4a50 a588 417:      lda  pr_ptr      ;no
4a52 18 418:      clc
4a53 6905 419:      adc  #5
4a55 8588 420:      sta  pr_ptr
4a57 a589 421:      lda  pr_ptr+1
4a59 6900 422:      adc  #0
4a5b 48 423:      pha
4a5c a588 424:      lda  pr_ptr
4a5e 48 425:      pha
4a5f ad6b4a 426:     lda  .gsa
4a62 ae6c4a 427:     ldx  .gsx
4a65 ac6d4a 428:     ldy  .gsy
4a68 4c6e4a 429:     jmp  get_string
4a6b 430:
4a6b 00 431: .gsa:      dc.b  0
4a6c 00 432: .gsx:      dc.b  0
4a6d 00 433: .gsy:      dc.b  0
4a6e 434:
4a6e 435:
4a6e c953 436: get_string:      cmp  #83      ;ma
4a70 9002 437:      bcc  .gsentry      ;ye
4a72 a950 438:      lda  #80
4a74 8d954b 439: .gsentry:      sta  .maxlth      ;sa
4a77 8e964b 440:      stx  .mask
4a7a 8c974b 441:      sty  .return
4a7d 442:
4a7d a900 443:      lda  #0
4a7f 854d 444:      sta  attract
4a81 445:
4a81 446:
4a81 a580 447:      lda  in_ptr
4a83 8d8f4b 448:      sta  .address      ;ho
4a86 a581 449:      lda  in_ptr+1      ;t
4a88 8d904b 450:      sta  .address+1
4a8b a900 451:      lda  #0
4a8d 8d994b 452:      sta  .instr      ;ze
4a90 453:

```

```

4a90 202a4f 454: .gsloop:      jsr  get_key      ;ge
4a93 c95c 455:          cmp  #"          ;ca
4a95 f0f9 456:          beq  .gsloop      ;ye
4a97 c99b 457:          cmp  #$9b        ;ca
4a99 d003 458:          bne  .nogs        ;no
4a9b 4c414b 459:          jmp  .gscr        ;ye
4a9e c90e 460: .nogs:      cmp  #14        ;tr
4aa0 d003 461:          bne  .not14
4aa2 4c414b 462:          jmp  .gscr
4aa5 c97e 463: .not14:     cmp  #126
4aa7 f070 464:          beq  .bspace
4aa9 c91e 465:          cmp  #30
4aab f06c 466:          beq  .bspace
4aad c918 467:          cmp  #24          ;^X
4aaf d003 468:          bne  .nowipe
4ab1 4c2e4b 469:          jmp  .wipeout
4ab4 297f 470: .nowipe:    and  #$7f        ;st
4ab6 8d984b 471:          sta  .key          ;ho
4ab9      472:
4ab9      473: ;    now, see if it falls within requested mask
4ab9      474: ;    ~~~~~
4ab9      475:
4ab9      476:
4ab9 ae964b 477:          ldx  .mask          ;ge
4abc f004 478:          beq  .alphanum
4abe e001 479:          cpx  #1
4ac0 f00b 480:          beq  .numeric
4ac2      481:
4ac2 c920 482: .alphanum:  cmp  #32          ;<
4ac4 90ca 483:          bcc  .gsloop      ;ye
4ac6 c97b 484:          cmp  #'z'+1        ;>
4ac8 b0c6 485:          bcs  .gsloop      ;ye
4aca 4cd84a 486:          jmp  .gsstore
4acd      487:
4acd c930 488: .numeric:   cmp  #'0'        ;<
4acf 90bf 489:          bcc  .gsloop      ;ye
4ad1 c93a 490:          cmp  #'9'+1        ;>
4ad3 b0bb 491:          bcs  .gsloop      ;ye
4ad5 4cd84a 492:          jmp  .gsstore
4ad8      493:
4ad8 ad994b 494: .gsstore:   lda  .instr      ;ge
4adb cd954b 495:          cmp  .maxlth
4ade f0b0 496:          beq  .gsloop      ;ig
4ae0 b0ae 497:          bcs  .gsloop      ;m
4ae2 a8 498:          tay
4ae3 c8 499:          iny
4ae4 ad984b 500:          lda  .key
4ae7 c920 501:          cmp  #32          ;is
4ae9 d007 502:          bne  .notspc      ;no
4aeb c001 503:          cpy  #1          ;ls
4aed d003 504:          bne  .notspc
4aef 4c904a 505:          jmp  .gsloop      ;ye
4af2 99994b 506: .notspc:    sta  .instr,y    ;sa

```

```

4af5 ee994b 507:      inc  .instr      ;in
4af8 ad984b 508: .nohide:    lda  .key
4afb 20924e 509: .showit:    jsr  echo
4afe ad954b 510:      lda  .maxlth      ;on
4b01 c901  511:      cmp  #1
4b03 d011  512:      bne  .nohot      ;no
4b05 ad994b 513:      lda  .instr
4b08 c901  514:      cmp  #1          ;1
4b0a d00a  515:      bne  .nohot
4b0c ad8744 516:      lda  u_flags
4b0f 2908  517:      and  #f_hotkey    ;us
4b11 f003  518:      beq  .nohot      ;no
4b13 4c414b 519:      jmp  .gscr        ;ye
4b16 4c904a 520: .nohot:      jmp  .gsloop
4b19      521:
4b19 ad994b 522: .bspace:    lda  .instr      ;ca
4b1c c900  523:      cmp  #0          ;i
4b1e d003  524:      bne  .nt0
4b20 4c904a 525:      jmp  .gsloop
4b23 ce994b 526: .nt0:      dec  .instr      ;-1
4b26 a97e  527:      lda  #126        ;ou
4b28 20924e 528:      jsr  echo
4b2b 4c904a 529:      jmp  .gsloop
4b2e      530:
4b2e ad994b 531: .wipeout:   lda  .instr      ;ca
4b31 d003  532:      bne  .wt0
4b33 4c904a 533:      jmp  .gsloop
4b36 ce994b 534: .wt0:      dec  .instr      ;-1
4b39 a97e  535:      lda  #126        ;ou
4b3b 20924e 536:      jsr  echo
4b3e 4c2e4b 537:      jmp  .wipeout      ;go
4b41      538:
4b41 ad994b 539: .gscr:      lda  .instr      ;ge
4b44 d008  540:      bne  .nonnull      ;it
4b46 ae974b 541:      ldx  .return      ;al
4b49 d003  542:      bne  .nonnull      ;ye
4b4b 4c904a 543:      jmp  .gsloop      ;no
4b4e      544:
4b4e      545: ;   done, so move our string into where they want it.
4b4e      546: ;   ~~~~~
4b4e      547:
4b4e a582  548: .nonnull:   lda  s_ptr
4b50 8d914b 549:      sta  .holds
4b53 a583  550:      lda  s_ptr+1
4b55 8d924b 551:      sta  .holds+1
4b58 a584  552:      lda  d_ptr
4b5a 8d934b 553:      sta  .holdd
4b5d a585  554:      lda  d_ptr+1
4b5f 8d944b 555:      sta  .holdd+1
4b62      556:      mea  .instr,s_ptr
4b62 a999  ----:      lda  #low.instr
4b64 8582  ----:      sta  s_ptr
4b66 a94b  ----:      lda  #high.instr

```

```

4b68 8583 ----:      sta  s_ptr + 1
4b6a ad8f4b 557:      lda  .address
4b6d 8584 558:      sta  d_ptr
4b6f ad904b 559:      lda  .address+1
4b72 8585 560:      sta  d_ptr+1
4b74 203e50 561:      jsr  move_str      ;mo
4b77 ad914b 562:      lda  .holds
4b7a 8582 563:      sta  s_ptr
4b7c ad924b 564:      lda  .holds+1
4b7f 8583 565:      sta  s_ptr+1
4b81 ad934b 566:      lda  .holdd
4b84 8584 567:      sta  d_ptr
4b86 ad944b 568:      lda  .holdd+1
4b89 8585 569:      sta  d_ptr+1
4b8b ad994b 570:      lda  .instr      ;re
4b8e 60 571:      rts
4b8f 572:
4b8f 573:
4b8f 574: .address:      ds.b 2
4b91 575: .holds:        ds.b 2
4b93 576: .holdd:        ds.b 2
4b95 577: .maxlth:       ds.b 1
4b96 578: .mask:        ds.b 1
4b97 579: .return:       ds.b 1
4b98 580: .key:         ds.b 1
4b99 581: .instr:        ds.b 86
4bef 582:
4bef 583:
4bef 584: ; Search the string pointed to by S_PTR for the subs
4bef 585: ; pointed to by D_PTR. Returns a 0 if substring not
4bef 586: ; ~~~~~
4bef 587:
4bef a000 588: substr:      ldy  #0
4bf1 b182 589:      lda  (s_ptr),y
4bf3 8d634c 590:      sta  .s_lth
4bf6 b184 591:      lda  (d_ptr),y
4bf8 8d644c 592:      sta  .d_lth
4bfb a582 593:      lda  s_ptr
4bfd 8d1f4c 594:      sta  .comp+1
4c00 a583 595:      lda  s_ptr+1
4c02 8d204c 596:      sta  .comp+2
4c05 a584 597:      lda  d_ptr
4c07 8d284c 598:      sta  .get+1
4c0a a585 599:      lda  d_ptr+1
4c0c 8d294c 600:      sta  .get+2
4c0f a901 601:      lda  #1
4c11 8d624c 602:      sta  .loc
4c14 a000 603:      ldy  #0
4c16 a200 604:      ldx  #0
4c18 20464c 605:      jsr  .up_str
4c1b 20534c 606:      jsr  .up_sub
4c1e 607:
4c1e b9ffff 608: .comp:      lda  $ffff,y

```

```

4c21 20c44f 609:      jsr  upcase
4c24 8d614c 610:      sta  .temp
4c27 bdffff 611: .get:      lda  $ffff,x
4c2a 20c44f 612:      jsr  upcase
4c2d cd614c 613:      cmp  .temp
4c30 f00b 614:      beq  .match
4c32 a201 615: .nomatch:      ldx  #1
4c34 ee624c 616:      inc  .loc
4c37 20464c 617:      jsr  .up_str
4c3a 4c1e4c 618:      jmp  .comp
4c3d 20534c 619: .match:      jsr  .up_sub
4c40 20464c 620:      jsr  .up_str
4c43 4c1e4c 621:      jmp  .comp
4c46      622:
4c46 c8 623: .up_str:      iny
4c47 cc634c 624:      cpy  .s_lth
4c4a 9006 625:      bcc  .xup
4c4c f004 626:      beq  .xup
4c4e 68 627:      pla
4c4f 68 628:      pla
4c50 a900 629:      lda  #0
4c52 60 630: .xup:      rts
4c53      631:
4c53 e8 632: .up_sub:      inx
4c54 ec644c 633:      cpx  .d_lth
4c57 9007 634:      bcc  .xsub
4c59 f005 635:      beq  .xsub
4c5b 68 636:      pla
4c5c 68 637:      pla
4c5d ad624c 638:      lda  .loc
4c60 60 639: .xsub:      rts
4c61      640:
4c61 00 641: .temp:      dc.b 0
4c62 00 642: .loc:      dc.b 0
4c63 00 643: .s_lth:     dc.b 0
4c64 00 644: .d_lth:     dc.b 0
4c65      645:
4c65      646: ;   Make an ASCII ($9b terminated) string out of a lth-b
4c65      647: ;   ~~~~~
4c65      648:
4c65 a000 649: toascii:      ldy  #0
4c67 b182 650:      lda  (s_ptr),y      ;ge
4c69 8d8b4c 651:      sta  .lth
4c6c f019 652:      beq  .null
4c6e c8 653:      iny
4c6f ee8b4c 654:      inc  .lth
4c72 b182 655: .loop:      lda  (s_ptr),y
4c74 88 656:      dey
4c75 9184 657:      sta  (d_ptr),y
4c77 c8 658:      iny
4c78 c8 659:      iny
4c79 cc8b4c 660:      cpy  .lth
4c7c d0f4 661:      bne  .loop

```

```

4c7e ac8b4c 662: .done:      ldy  .lth
4c81 88      663:          dey
4c82 a99b    664: .rtn:      lda  #$9b
4c84 9184    665:          sta  (d_ptr),y
4c86 60      666:          rts
4c87 a000    667: .null:     ldy  #0
4c89 f0f7    668:          beq  .rtn
4c8b        669:
4c8b 01      670: .lth:      dc.b  1
4c8c        671:
4c8c        672: ;   Make a length-byte string from an ASCII ($9b termina
4c8c        673: ;   ~~~~~
4c8c        674:
4c8c a000    675: tostr:     ldy  #0
4c8e b182    676: .tfe:      lda  (s_ptr),y
4c90 f007    677:          beq  .gotend
4c92 c99b    678:          cmp  #$9b
4c94 f003    679:          beq  .gotend
4c96 c8      680:          iny
4c97 d0f5    681:          bne  .tfe
4c99 8cb44c  682: .gotend:   sty  .lth      ;save l
4c9c c000    683:          cpy  #0
4c9e f00c    684:          beq  .rtn
4ca0 a000    685:          ldy  #0
4ca2 b182    686: .loop:     lda  (s_ptr),y
4ca4 c8      687:          iny
4ca5 9184    688:          sta  (d_ptr),y
4ca7 ccb44c  689:          cpy  .lth
4caa 90f6    690:          bcc  .loop
4cac a000    691: .rtn:      ldy  #0
4cae adb44c  692:          lda  .lth
4cb1 9184    693:          sta  (d_ptr),y
4cb3 60      694:          rts
4cb4        695:
4cb4 01      696: .lth:      dc.b  1
4cb5        697:
4cb5        698:
4cb5        699: ;=====
4cb5        700: ;   These routines work very much like the filename crunch
4cb5        701: ;   SpartaDOS.  On consecutive calls to CRUNCH, the next
4cb5        702: ;   command line (pointed to by IN_PTR) will be placed int
4cb5        703: ;   If there are no more words on the command line, then C
4cb5        704: ;   with carry set.
4cb5        705: ;   When calling the parser for the first time on a new
4cb5        706: ;   MUST zero out the BUFOFF field.  Also, extraneous spac
4cb5        707: ;   compressed out of the input line, and the command word
4cb5        708: ;   to upper case.
4cb5        709: ;=====
4cb5        710:
4cb5        711:
4cb5 a40a    712: crunch:    ldy  bufoff      ;ge
4cb7 a200    713:          ldx  #0          ;zero stori
4cb9        714:

```

```

4cb9 b90346 715: .nonspc:      lda  instr,y      ;move buffo
4cbc c99b 716:          cmp  #$9b
4cbe f011 717:          beq  .nomore      ;end of lin
4cc0 c920 718:          cmp  #32         ;= space?
4cc2 f00a 719:          beq  .go_on       ;yep
4cc4 c92c 720:          cmp  #','
4cc6 f006 721:          beq  .go_on
4cc8 c92d 722:          cmp  #'-'
4cca f002 723:          beq  .go_on
4ccc d005 724:          bne  .gotone      ;found one
4cce c8 725: .go_on:      iny
4ccf d0e8 726:          bne  .nonspc      ;keep looki
4cd1      727:
4cd1 38 728: .nomore:      sec          ;signal err
4cd2 60 729:          rts          ; and retur
4cd3      730:
4cd3 b90346 731: .gotone:      lda  instr,y      ;get byte o
4cd6 c99b 732:          cmp  #$9b
4cd8 f016 733:          beq  .end         ;end of lin
4cda      734:
4cda c920 735:          cmp  #32         ;space?
4cdc f012 736:          beq  .end         ;yes
4cde c92c 737:          cmp  #','
4ce0 f00e 738:          beq  .end
4ce2 c92d 739:          cmp  #'-'
4ce4 f00a 740:          beq  .end
4ce6      741:
4ce6 20c44f 742:          jsr  upcase
4ce9 9dbf43 743:          sta  word,x
4cec e8 744:          inx
4ced c8 745:          iny
4cee d0e3 746:          bne  .gotone      ;keep going
4cf0      747:
4cf0 840a 748: .end:          sty  bufoff      ;save new b
4cf2 a99b 749:          lda  #$9b          ;put null b
4cf4 9dbf43 750:          sta  word,x
4cf7 18 751:          clc
4cf8 60 752: end_crunch:      rts
4cf9      753:
4cf9      754:
4cf9      755:
4cf9      756: ;-----
4cf9      757: ;prints Routeen macro
4cf9      758: ;this is a slow print to the screen
4cf9      759: ;usage jsr  prints
4cf9      760: ; .byte $9b," string to be printed",$ff
4cf9      761: ;-----
4cf9      762: ;
4cf9      763: prints:
4cf9 68 764:          pla
4cfa 8d0a4d 765:          sta  .pstr+1
4cfd 68 766:          pla
4cfe 8d0b4d 767:          sta  .pstr+2

```



```

4d01 ee0a4d 768: .prsl:      inc    .pstr+1
4d04 d003 769:             bne    .pstr
4d06 ee0b4d 770:             inc    .pstr+2
4d09 adffff 771: .pstr:      lda    $ffff
4d0c c9ff 772:             cmp    #$ff
4d0e f006 773:             beq    .estri
4d10 204f4d 774:             jsr    fast_output
4d13 4c014d 775:             jmp    .prsl
4d16 ad0b4d 776: .estri:     lda    .pstr+2
4d19 48 777:             pha
4d1a ad0a4d 778:             lda    .pstr+1
4d1d 48 779:             pha
4d1e 60 780:             rts
4d1f      781: ;
4d1f      782: ;-----
4d1f      783: ;actual screen handler
4d1f      784: ;screen offset definitions
4d1f      785: ;-----
4d1f 0000 786: scr_offset:  dc.w    0
4d21 2800 787:             dc.w    40
4d23 5000 788:             dc.w    80
4d25 7800 789:             dc.w   120
4d27 a000 790:             dc.w   160
4d29 c800 791:             dc.w   200
4d2b f000 792:             dc.w   240
4d2d 1801 793:             dc.w   280
4d2f 4001 794:             dc.w   320
4d31 6801 795:             dc.w   360
4d33 9001 796:             dc.w   400
4d35 b801 797:             dc.w   440
4d37 e001 798:             dc.w   480
4d39 0802 799:             dc.w   520
4d3b 3002 800:             dc.w   560
4d3d 5802 801:             dc.w   600
4d3f 8002 802:             dc.w   640
4d41 a802 803:             dc.w   680
4d43 d002 804:             dc.w   720
4d45 f802 805:             dc.w   760
4d47 2003 806:             dc.w   800
4d49 4803 807:             dc.w   840
4d4b 7003 808:             dc.w   880
4d4d 9803 809:             dc.w   920
4d4f      810: ;
4d4f      811: fast_output:
4d4f 855a 812:             sta    $5a
4d51 ad0407 813:             lda    sc.redirect
4d54 c902 814:             cmp    #print_p
4d56 f01f 815:             beq    .go_cio
4d58      816: .leo:
4d58 adff02 817:             lda    $02ff      ; CONTROL 1
4d5b c900 818:             cmp    #$00
4d5d d0f9 819:             bne    .leo
4d5f      820:

```

```

4d5f a55a 821:      lda  $5a
4d61 c920 822:      cmp  #32
4d63 9004 823:      bcc  .do_look
4d65 c97d 824:      cmp  #125
4d67 9013 825:      bcc  .do_here
4d69 a000 826: .do_look:      ldy  #0
4d6b b9724e 827: .lookup:      lda  .char_table,y
4d6e f00c 828:      beq  .do_here
4d70 c55a 829:      cmp  $5a
4d72 f003 830:      beq  .go_cio
4d74 c8 831:      iny
4d75 d0f4 832:      bne  .lookup
4d77      833:
4d77      834:
4d77 a55a 835: .go_cio:      lda  $5a
4d79 4c824e 836:      jmp  putlocal
4d7c      837:
4d7c      838:
4d7c a000 839: .do_here:      ldy  #0
4d7e a55d 840:      lda  $5d
4d80 915e 841:      sta  ($5e),y
4d82 a55a 842:      lda  $5a
4d84 c99b 843:      cmp  #$9b
4d86 f044 844:      beq  .docr
4d88 20514e 845: .notcr:      jsr  .get_adr
4d8b a55a 846:      lda  $5a
4d8d 297f 847:      and  #$7f
4d8f c920 848:      cmp  #32
4d91 9007 849:      bcc  .add64
4d93 c960 850:      cmp  #96
4d95 9009 851:      bcc  .sub32
4d97 4ca34d 852:      jmp  .asis
4d9a 18 853: .add64:      clc
4d9b 6940 854:      adc  #64
4d9d 4ca34d 855:      jmp  .asis
4da0 38 856: .sub32:      sec
4da1 e920 857:      sbc  #32
4da3 245a 858: .asis:      bit  $5a
4da5 1002 859:      bpl  .xxlate
4da7 0980 860:      ora  #$80
4da9 a000 861: .xxlate:      ldy  #0
4dab 915e 862:      sta  ($5e),y
4dad e655 863:      inc  $55
4daf e663 864:      inc  $63
4db1 a555 865:      lda  $55
4db3 c928 866:      cmp  #40
4db5 b019 867:      bcs  .next_row
4db7 c8 868:      iny
4db8 b15e 869:      lda  ($5e),y
4dba 855d 870:      sta  $5d
4dbc 0980 871:      ora  #$80
4dbe aef002 872:      ldx  752
4dc1 d002 873:      bne  .nocurs1

```

```

4dc3 915e 874:      sta  ($5e),y
4dc5 e65e 875: .nocurs1:  inc  $5e
4dc7 d002 876:      bne  .nooav
4dc9 e65f 877:      inc  $5e+1
4dcb 60    878: .nooav:      rts
4dcc a900 879: .docr:      lda  #0
4dce 8563 880:      sta  $63
4dd0 a552 881: .next_row:  lda  $52
4dd2 8555 882:      sta  $55
4dd4 e654 883:      inc  $54
4dd6 a454 884:      ldy  $54
4dd8 c018 885:      cpy  #24
4dda b013 886:      bcs  .scroll
4ddc 20514e 887:      jsr  .get_adr
4ddf a000 888:      ldy  #0
4de1 b15e 889:      lda  ($5e),y
4de3 855d 890:      sta  $5d
4de5 aef002 891:      ldx  752
4de8 d004 892:      bne  .nocurs2
4dea 0980 893:      ora  #$80
4dec 915e 894:      sta  ($5e),y
4dee 60    895: .nocurs2:  rts
4def      896:
4def c654 897: .scroll:  dec  $54
4df1 a558 898:      lda  $58
4df3 8568 899:      sta  $68
4df5 18    900:      clc
4df6 6928 901:      adc  #40
4df8 855e 902:      sta  $5e
4dfa a559 903:      lda  $58+1
4dfc 8569 904:      sta  $68+1
4dfe 6900 905:      adc  #0
4e00 855f 906:      sta  $5e+1
4e02 a000 907:      ldy  #0
4e04 a203 908:      ldx  #3
4e06 b15e 909: .scloop:  lda  ($5e),y
4e08 9168 910:      sta  ($68),y
4e0a c8    911:      iny
4e0b d0f9 912:      bne  .scloop
4e0d e65f 913:      inc  $5e+1
4e0f e669 914:      inc  $68+1
4e11 ca    915:      dex
4e12 d0f2 916:      bne  .scloop
4e14 b15e 917: .scloop2:  lda  ($5e),y
4e16 9168 918:      sta  ($68),y
4e18 c8    919:      iny
4e19 c098 920:      cpy  #152
4e1b 90f7 921:      bcc  .scloop2
4e1d a558 922:      lda  $58
4e1f 18    923:      clc
4e20 6998 924:      adc  #920&$ff
4e22 855e 925:      sta  $5e
4e24 a559 926:      lda  $58+1

```

```

4e26 6903 927:      adc  #920/256
4e28 855f 928:      sta  $5e+1
4e2a a000 929:      ldy  #0
4e2c 98   930:      tya
4e2d 915e 931: .clear:      sta  ($5e),y
4e2f c8   932:      iny
4e30 c028 933:      cpy  #40
4e32 90f9 934:      bcc  .clear
4e34 aef002 935:      ldx  752
4e37 d006 936:      bne  .nocurs3
4e39 a980 937:      lda  #$80
4e3b a455 938:      ldy  $55
4e3d 915e 939:      sta  ($5e),y
4e3f a900 940: .nocurs3:      lda  #0
4e41 855d 941:      sta  $5d
4e43 a55e 942:      lda  $5e
4e45 18   943:      clc
4e46 6555 944:      adc  $55
4e48 855e 945:      sta  $5e
4e4a a55f 946:      lda  $5e+1
4e4c 6900 947:      adc  #0
4e4e 855f 948:      sta  $5e+1
4e50 60   949:      rts
4e51 a554 950: .get_adr:      lda  $54
4e53 0a   951:      asl
4e54 a8   952:      tay
4e55 a558 953:      lda  $58
4e57 18   954:      clc
4e58 791f4d 955:      adc  scr_offset,y
4e5b 855e 956:      sta  $5e
4e5d a559 957:      lda  $58+1
4e5f 79204d 958:      adc  scr_offset+1,y
4e62 855f 959:      sta  $5e+1
4e64 a55e 960:      lda  $5e
4e66 18   961:      clc
4e67 6555 962:      adc  $55
4e69 855e 963:      sta  $5e
4e6b a55f 964:      lda  $5e+1
4e6d 6900 965:      adc  #0
4e6f 855f 966:      sta  $5e+1
4e71 60   967:      rts
4e72      968: .char_table:
4e72 1b1c1d 969:      dc.b  27,28,29,30,31,125,126,127
4e7a 9c9d9e 970:      dc.b  156,157,158,159,253,254,255
4e82      971:
4e82      972:
4e82      973: putlocal:
4e82 a20b 974:      ldx  #11
4e84 8e4203 975:      stx  $0342
4e87 a200 976:      ldx  #0
4e89 8e4803 977:      stx  $0348
4e8c 8e4903 978:      stx  $0349
4e8f 4c56e4 979:      jmp  $e456

```

```

4e92      980: ;
4e92      981: ;
4e92      982: ;-----
4e92      983: ; PRINT BYTE
4e92      984: ;-----
4e92      985: echo:
4e92 85b8  986: sprint:          sta  $b8
4e94 a904  987:          lda  #$04
4e96 8d4a03 988:          sta  $034a
4e99 86b6  989:          stx  $b6
4e9b 84b7  990:          sty  $b7
4e9d a206  991:          ldx  #$06
4e9f bd01e4 992:          lda  $e400+1,x
4ea2 85b3  993:          sta  $b3
4ea4 bd00e4 994:          lda  $e400,x
4ea7 85b2  995:          sta  $b2
4ea9 a5b8  996:          lda  $b8
4eab 20ba4e 997:          jsr  .a
4eae a6b6  998:          ldx  $b6
4eb0 a4b7  999:          ldy  $b7
4eb2 a90c 1000:          lda  #$0c
4eb4 8d4a03 1001:          sta  $034a
4eb7 a5b8 1002:          lda  $b8
4eb9 60    1003:          rts
4eba      1004: ;
4eba e6b2 1005: .a:          inc  $b2
4ebc 6cb200 1006:          jmp  ($b2)
4ebf 60    1007:          rts
4ec0      1008:
4ec0      1009: ; prints: prints string with address in PR_PTR, length by
4ec0      1010: ; ~~~~~
4ec0      1011:
4ec0      1012: echos:
4ec0 a000 1013:          ldy  #0
4ec2 b188 1014:          lda  (pr_ptr),y          ;ge
4ec4 f017 1015:          beq  .xpri              ;ex
4ec6 8ddf4e 1016:          sta  .maxlth
4ec9 c8    1017:          iny
4eca 8cde4e 1018:          sty  .tempy
4ecd      1019:
4ecd acde4e 1020: .pril:      ldy  .tempy
4ed0 b188 1021:          lda  (pr_ptr),y
4ed2 20924e 1022:          jsr  echo
4ed5 eede4e 1023: .prnext:    inc  .tempy
4ed8 cedf4e 1024:          dec  .maxlth
4edb d0f0 1025:          bne  .pril
4edd 60    1026: .xpri:      rts
4ede      1027:
4ede 01    1028: .tempy:     dc.b  1
4edf 01    1029: .maxlth:    dc.b  1
4ee0      1030:
4ee0      1031: ; prints: prints string with address in PR_PTR, length by
4ee0      1032: ; ~~~~~

```

```

4ee0      1033:
4ee0      1034: echosE:
4ee0 a000  1035:          ldy  #0
4ee2 b188  1036:          lda  (pr_ptr),y          ;ge
4ee4 f01b  1037:          beq  .xpri              ;ex
4ee6 8d034f 1038:          sta  .maxlth
4ee9 c8    1039:          iny
4eea 8c024f 1040:          sty  .tempy
4eed      1041:
4eed ac024f 1042: .pril:          ldy  .tempy
4ef0 b188  1043:          lda  (pr_ptr),y
4ef2 c900  1044:          cmp  #$00
4ef4 f00b  1045:          beq  .xpri              ; s
4ef6 20924e 1046:          jsr  echo
4ef9 ee024f 1047: .prnext:        inc  .tempy
4efc ce034f 1048:          dec  .maxlth
4eff d0ec  1049:          bne  .pril
4f01 60    1050: .xpri:          rts
4f02      1051:
4f02 01    1052: .tempy:         dc.b  1
4f03 01    1053: .maxlth:        dc.b  1
4f04      1054:
4f04      1055:
4f04      1056:
4f04      1057: ;   lookup command in table (in s_ptr) and execute it
4f04      1058: ;   ~~~~~
4f04      1059:
4f04 20c44f 1060: execute:        jsr  upcase
4f07 8d294f 1061:          sta  .key
4f0a a000  1062:          ldy  #0
4f0c b182  1063: .loop:          lda  (s_ptr),y
4f0e f00a  1064:          beq  .found
4f10 cd294f 1065:          cmp  .key
4f13 f005  1066:          beq  .found
4f15 c8    1067:          iny
4f16 c8    1068:          iny
4f17 c8    1069:          iny
4f18 d0f2  1070:          bne  .loop
4f1a c8    1071: .found:         iny
4f1b b182  1072:          lda  (s_ptr),y
4f1d 8d274f 1073:          sta  .goaddr+1
4f20 c8    1074:          iny
4f21 b182  1075:          lda  (s_ptr),y
4f23 8d284f 1076:          sta  .goaddr+2
4f26 4cffff 1077: .goaddr:        jmp  $ffff
4f29 00    1078: .key:          dc.b  0
4f2a      1079:
4f2a      1080: ;-----
4f2a      1081: ; GET KEY ROUTEEN
4f2a      1082: ;this routeen get's one key and
4f2a      1083: ;return it to the accumulator
4f2a      1084: ;-----
4f2a      1085: ;

```

```

4f2a 86b6 1086: get_key:      stx  $b6
4f2c 84b7 1087:                sty  $b7
4f2e 20364f 1088:                jsr  .a
4f31 a6b6 1089:                ldx  $b6
4f33 a4b7 1090:                ldy  $b7
4f35 60 1091:                rts
4f36 ad25e4 1092: .a:          lda  $e420+5
4f39 48 1093:                pha
4f3a ad24e4 1094:                lda  $e420+4
4f3d 48 1095:                pha
4f3e 60 1096:                rts
4f3f      1097:
4f3f      1098:
4f3f      1099:
4f3f a99b 1100: cr_lf:        lda  #$9b
4f41 20924e 1101:                jsr  echo
4f44 60 1102:                rts
4f45      1103:
4f45      1104:
4f45 a904 1105: open_in:      lda  #4
4f47 4c634f 1106:                jmp  do_open
4f4a      1107:
4f4a a908 1108: open_out:     lda  #8
4f4c 4c634f 1109:                jmp  do_open
4f4f      1110:
4f4f a909 1111: open_app:     lda  #9
4f51 4c634f 1112:                jmp  do_open
4f54      1113:
4f54 a90c 1114: open_updt:    lda  #12
4f56 4c634f 1115:                jmp  do_open
4f59      1116:
4f59 a906 1117: open_dir:     lda  #6
4f5b 4c634f 1118:                jmp  do_open
4f5e      1119:
4f5e a914 1120: open_raw:     lda  #20      ;open raw d
4f60 4c634f 1121:                jmp  do_open
4f63      1122:
4f63      1123: do_open:
4f63 48 1124:                pha      ;open files
4f64 8a 1125:                txa      ;
4f65 48 1126:                pha
4f66 20904f 1127:                jsr  close      ;make sure
4f69 68 1128:                pla
4f6a aa 1129:                tax
4f6b 68 1130:                pla
4f6c 9d4a03 1131:                sta  icaux1,x
4f6f a903 1132:                lda  #ccopen
4f71 9d4203 1133:                sta  iccom,x
4f74 a586 1134:                lda  fn_ptr
4f76 9d4403 1135:                sta  icbal,x
4f79 a587 1136:                lda  fn_ptr+1
4f7b 9d4503 1137:                sta  icbah,x
4f7e a950 1138:                lda  #80

```

```

4f80 9d4803 1139:      sta  icbll,x
4f83 a900  1140:      lda  #0
4f85 9d4903 1141:      sta  icblh,x
4f88 a980  1142:      lda  #128      ;for long d
4f8a 9d4b03 1143:      sta  icaux2,x
4f8d 4c56e4 1144:      jmp  ciov
4f90      1145:
4f90 a90c  1146: close:      lda  #cccclose
4f92 9d4203 1147:      sta  iccom,x
4f95 4c56e4 1148:      jmp  ciov
4f98      1149:
4f98      1150: ;   Generic system error routine
4f98      1151: ;   ~~~~~
4f98      1152:
4f98      1153:
4f98      1154: sys_error:      save                ;sa
4f98 48  ----:      pha
4f99 98  ----:      tya
4f9a 48  ----:      pha
4f9b 8a  ----:      txa
4f9c 48  ----:      pha
4f9d 8cc34f 1155:      sty  .temp                ;sa
4fa0 20f94c 1156:      jsr  printsi
4fa3 9b9b53 1157:      dc.b  $9b,$9b,"System Error: ",-1
4fb4 adc34f 1158:      lda  .temp
4fb7 20d649 1159:      jsr  pr_byte
4fba 203f4f 1160:      jsr  cr_lf
4fbd      1161:      restore                ;re
4fbd 68  ----:      pla
4fbe aa  ----:      tax
4fbf 68  ----:      pla
4fc0 a8  ----:      tay
4fc1 68  ----:      pla
4fc2 60  1162:      rts
4fc3      1163:
4fc3 01  1164: .temp:      dc.b  1
4fc4      1165:
4fc4      1166:
4fc4      1167:
4fc4      1168:
4fc4      1169:
4fc4      1170:
4fc4 c961  1171: upcase:      cmp  #'a'
4fc6 9009  1172:      bcc  xupcase
4fc8 c97b  1173:      cmp  #'z'+1
4fca f005  1174:      beq  xupcase
4fcc b003  1175:      bcs  xupcase
4fce 38  1176:      sec
4fcf e920  1177:      sbc  #32
4fd1 60  1178: xupcase:      rts
4fd2      1179:
4fd2      1180:
4fd2 c941  1181: locase:      cmp  #'A'

```



```

4fd4 9009 1182:      bcc  xlocase
4fd6 c95b 1183:      cmp  #'Z'+1
4fd8 f005 1184:      beq  xlocase
4fda b003 1185:      bcs  xlocase
4fdc 18   1186:      clc
4fdd 6920 1187:      adc  #32
4fdf 60   1188: xlocase:      rts
4fe0      1189:
4fe0      1190:
4fe0      1191: ;   Compares the strings pointed to by s_ptr and d_ptr
4fe0      1192: ;   Returns with equal set if true, not equal if false
4fe0      1193: ;   The comparison IS NOT case-sensitive.
4fe0      1194: ;   ~~~~~
4fe0      1195:
4fe0 a000 1196: cmp_str:      ldy  #0          ;zero index
4fe2 b182 1197:      lda  (s_ptr),y      ;get length
4fe4 8d1050 1198:      sta  .lth
4fe7 d184 1199:      cmp  (d_ptr),y      ;they match
4fe9 d01f 1200:      bne  .noteq         ;nope
4feb c900 1201:      cmp  #0            ;null strin
4fed f01e 1202:      beq  .end          ;they match
4fef c8   1203:      iny
4ff0 b182 1204: .loop:      lda  (s_ptr),y      ;get next b
4ff2 20c44f 1205:      jsr  upcase
4ff5 8d1150 1206:      sta  .temp
4ff8 b184 1207:      lda  (d_ptr),y      ;they match
4ffa 20c44f 1208:      jsr  upcase
4ffd cd1150 1209:      cmp  .temp
5000 d008 1210:      bne  .noteq         ;nope..
5002 cc1050 1211:      cpy  .lth          ;end of str
5005 f006 1212:      beq  .end          ;yes -- the
5007 c8   1213:      iny
5008 d0e6 1214:      bne  .loop
500a a901 1215: .noteq:      lda  #1          ;se
500c 60   1216:      rts
500d a900 1217: .end:      lda  #0          ;set BEQ
500f 60   1218:      rts
5010 01   1219: .lth:      dc.b  1
5011 01   1220: .temp:     dc.b  1
5012      1221:
5012      1222:
5012 202d4a 1223: y_or_n:      jsr  get_str
5015 3c50 1224:      dc.w  .temp
5017 010000 1225:      dc.b  1,0,0
501a ad3d50 1226:      lda  .temp+1
501d 20c44f 1227:      jsr  upcase
5020 c959 1228:      cmp  #'Y'
5022 f012 1229:      beq  .yes
5024 c94e 1230:      cmp  #'N'
5026 f008 1231:      beq  .no
5028 a97e 1232:      lda  #126          ;ba
502a 20924e 1233:      jsr  echo
502d 4c1250 1234:      jmp  y_or_n        ; a

```

```

5030 203f4f 1235: .no:          jsr   cr_lf
5033 a94e 1236:          lda   #'N'
5035 60 1237:          rts
5036 203f4f 1238: .yes:          jsr   cr_lf
5039 a959 1239:          lda   #'Y'
503b 60 1240:          rts
503c 1241: .temp:        ds.b 2
503e 1242:
503e 1243:
503e 1244: ;   Move string pointed to by s_ptr to string pointed
503e 1245: ;   ~~~~~
503e 1246:
503e 1247:
503e a000 1248: move_str:      ldy   #0
5040 b182 1249:          lda   (s_ptr),y      ;ge
5042 a8 1250:          tay                ;go
5043 b182 1251: .loop:         lda   (s_ptr),y
5045 9184 1252:          sta   (d_ptr),y
5047 c000 1253:          cpy   #0
5049 f004 1254:          beq   .done
504b 88 1255:          dey
504c 4c4350 1256:          jmp   .loop
504f 60 1257: .done:         rts
5050 1258:
5050 ad9f44 1259: timestamp:     lda   in_clock      ;ho
5053 8d8a50 1260:          sta   .hold
5056 a901 1261:          lda   #1                ;si
5058 8d9f44 1262:          sta   in_clock          ;
505b 208b50 1263:          jsr   waitvbi          ;ma
505e a90d 1264:          lda   #13
5060 8ded43 1265:          sta   cur_date
5063 a90a 1266:          lda   #10
5065 8dfc43 1267:          sta   cur_time
5068 209850 1268:          jsr   get_TD            ;ge
506b ad8a50 1269:          lda   .hold            ;re
506e 8d9f44 1270:          sta   in_clock
5071 1271:          mea   cur_date,s_ptr    ;if
5071 a9ed ----:          lda   #low cur_date
5073 8582 ----:          sta   s_ptr
5075 a943 ----:          lda   #high cur_date
5077 8583 ----:          sta   s_ptr + 1
5079 1272:          mea   s_today,d_ptr    ; r
5079 a9df ----:          lda   #low s_today
507b 8584 ----:          sta   d_ptr
507d a943 ----:          lda   #high s_today
507f 8585 ----:          sta   d_ptr + 1
5081 20e04f 1273:          jsr   cmp_str
5084 f003 1274:          beq   .noupd
5086 203e50 1275:          jsr   move_str
5089 60 1276: .noupd:        rts
508a 1277:
508a 00 1278: .hold:         dc.b 0
508b 1279:

```

```

508b 1280: ;      Wait for a VBI to occur
508b 1281: ;      ~~~~~
508b 1282:
508b a514 1283: waitvbi:      lda  $14
508d c514 1284: .lp1:      cmp  $14
508f f0fc 1285:      beq  .lp1
5091 a514 1286:      lda  $14
5093 c514 1287: .lp2:      cmp  $14
5095 f0fc 1288:      beq  .lp2
5097 60 1289:      rts
5098 1290:
5098 1291: ;      get both the "formatted" and "unformatted" date
5098 1292: ;      ~~~~~
5098 1293:
5098 ad01d3 1294: get_TD:      lda  portb      ;sa
509b 48 1295:      pha
509c 29fe 1296:      and  #$FE
509e 8d01d3 1297:      sta  portb      ;enable RAM
50a1 a040 1298:      ldy  #.tdbuff & $ff
50a3 a251 1299:      ldx  #.tdbuff / 256
50a5 20c9ff 1300:      jsr  $FFC9      ;ge
50a8 20c0ff 1301:      jsr  $FFC0      ;ge
50ab 68 1302:      pla
50ac 8d01d3 1303:      sta  portb      ;restore PO
50af 9009 1304:      bcc  .noerr
50b1 1305:
50b1 a900 1306: .error:      lda  #0
50b3 8ded43 1307:      sta  cur_date
50b6 8dfc43 1308:      sta  cur_time
50b9 60 1309:      rts
50ba 1310:
50ba 1311: ;      move the formatted date into cur_date, cur_time
50ba 1312: ;      ~~~~~
50ba 1313:
50ba a000 1314: .noerr:      ldy  #0
50bc a201 1315:      ldx  #1
50be b9a044 1316: .lp:      lda  tdbuff,y
50c1 9ded43 1317:      sta  cur_date,x
50c4 e8 1318:      inx
50c5 c8 1319:      iny
50c6 c00d 1320:      cpy  #13
50c8 d0f4 1321:      bne  .lp
50ca a201 1322:      ldx  #1
50cc c8 1323:      iny
50cd b9a044 1324: .lp2:      lda  tdbuff,y
50d0 9dfc43 1325:      sta  cur_time,x
50d3 e8 1326:      inx
50d4 c8 1327:      iny
50d5 c018 1328:      cpy  #24
50d7 d0f4 1329:      bne  .lp2
50d9 1330:
50d9 1331: ;      move the unformatted date into the byte fields
50d9 1332: ;      ~~~~~

```

```

50d9      1333:
50d9 a00d 1334:          ldy  #13      ;point to D
50db b10a 1335:          lda  (comtab),y
50dd 8d0c44 1336:          sta  cur_day
50e0 c8    1337:          iny
50e1 b10a 1338:          lda  (comtab),y
50e3 8d0b44 1339:          sta  cur_month
50e6 c8    1340:          iny
50e7 b10a 1341:          lda  (comtab),y
50e9 8d0d44 1342:          sta  cur_year
50ec c8    1343:          iny
50ed b10a 1344:          lda  (comtab),y
50ef 8d0e44 1345:          sta  cur_hour
50f2 c8    1346:          iny
50f3 b10a 1347:          lda  (comtab),y
50f5 8d0f44 1348:          sta  cur_minute
50f8 c8    1349:          iny
50f9 b10a 1350:          lda  (comtab),y
50fb 8d1044 1351:          sta  cur_second
50fe      1352:
50fe      1353: ;      now, figure out what weekday it is
50fe      1354: ;      ~~~~~
50fe      1355:
50fe adef43 1356:          lda  cur_date+2  ;2nd char o
5101 c96f 1357:          cmp  #'o'
5103 f021 1358:          beq  .monday
5105 c965 1359:          cmp  #'e'
5107 f022 1360:          beq  .wednesday
5109 c968 1361:          cmp  #'h'
510b f023 1362:          beq  .thursday
510d c972 1363:          cmp  #'r'
510f f024 1364:          beq  .friday
5111 c961 1365:          cmp  #'a'
5113 f025 1366:          beq  .saturday
5115 adee43 1367: .tues_sun:      lda  cur_date+1
5118 c954 1368:          cmp  #'T'
511a f005 1369:          beq  .tues
511c a900 1370: .sun:          lda  #0
511e 4c3c51 1371:          jmp  .exit
5121 a902 1372: .tues:         lda  #2
5123 4c3c51 1373:          jmp  .exit
5126 a901 1374: .monday:       lda  #1
5128 4c3c51 1375:          jmp  .exit
512b a903 1376: .wednesday:    lda  #3
512d 4c3c51 1377:          jmp  .exit
5130 a904 1378: .thursday:     lda  #4
5132 4c3c51 1379:          jmp  .exit
5135 a905 1380: .friday:       lda  #5
5137 4c3c51 1381:          jmp  .exit
513a a906 1382: .saturday:     lda  #6
513c 8d1144 1383: .exit:         sta  cur_weekday
513f 60    1384:          rts
5140      1385:

```

```
5140      1386: .tdbuff:      ds.b  30
515e      1387:
515e      1388:
515e      1389:
515e      1390:
515e      1391:
515e      1392:
515e      1393: begin:
515e      1394: top1:
515e a920 1395:      lda  #$20
5160 8d0952 1396:      sta  header
5163      1397: top:
5163 a97d 1398:      lda  #$7d
5165 20924e 1399:      jsr  echo
5168 20a442 1400:      jsr  tammy-3
516b      1401:
516b 20f94c 1402:      jsr  printsi
516e 9b 1403:      dc.b  $9b
516f 919292 1404:      dc.b  <+128>,"
5193 9b 1405:      dc.b  $9B
5194 fca0a0 1406:      dc.b  <+128>,"|  RealDos Datab
51b8 9b 1407:      dc.b  $9B
51b9 fca0a0 1408:      dc.b  <+128>,"| (c)2010 by Step
51dd 9b 1409:      dc.b  $9B
51de 9a9292 1410:      dc.b  <+128>,"
```