


```

0000 1: ;-----
0000 2: ;
0000 3: ; Copyright 2006 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: ; (General DOS formatter for real.dos 1.0 and Sparta
0000 10: ;-----
0000 11: ; Real.dos SnapShot.com
0000 12: ; This program is to replace
0000 13: ; Snap30hd.com
0000 14: ;
0000 15: ;snapshot v2.0 hd written By Tom Hunt Orginal date unknow
0000 16: ;
0000 17: ; SnapShot V3.0 hd
0000 18: ; Modified by Stephen J. Carden adding Mux Support 8/14/19
0000 19: ; Converted to xasm from Mac/65 3/6/2010 sjc
0000 20: ;
0000 21: ;
0000 22: ; ~~~~~
0000 23: ;-----
0000 24: ; Notes: o This source code MAY NOT be placed for download
0000 25: ; o "mea" is a macro that loads the address of the
0000 26: ; into the pointer specified by the second field.
0000 27: ;-----
0000 28: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 29: ; XASM st and IBM VERSIONS
0000 30: ;-----
0000 31:
0000 32:
0000 33:
0000 34:
0000 35:
0000 36:
0000 37:
1f00 38: .org $1F00
1f00 39: ;
1f00 40: BEGIN:
1f00 4c6522 41: CHECK_BEG: JMP START
1f03 42:
1f03 43:
1f03 44: ;
1f03 be 45: dc.b $be
1f04 00 46: STACK: dc.b 0
1f05 00 47: MY_PORTB: dc.b 0
1f06 48: ;
1f06 49: VECTORS:
1f06 50: ds.b 21
1f1b 9b 51: dc.b $9B
1f1c be 52: dc.b $be
1f1d 53:

```

```

1f1d      54: VIRTUAL:
1f1d 44313a 55:          dc.b  "D1:VIRTUAL1.0", $9B
1f2b      56:
1f2b      57:          ds.b  31
1f4a      58:
1f4a 00    59: R1F26:    dc.b  0
1f4b      60: ;
1f4b      61: HOT_KEY:
1f4b f2    62:          dc.b  242 ; Shift+Ctrl+0
1f4c df    63:          dc.b  223 ; Shift+Ctrl+1
1f4d de    64:          dc.b  222 ; Shift+Ctrl+2
1f4e da    65:          dc.b  218 ; Shift+Ctrl+3
1f4f d8    66:          dc.b  216 ; Shift+Ctrl+4
1f50 dd    67:          dc.b  221 ; Shift+Ctrl+5
1f51 db    68:          dc.b  219 ; Shift+Ctrl+6
1f52 f3    69:          dc.b  243 ; Shift+Ctrl+7
1f53 f5    70:          dc.b  245 ; Shift+Ctrl+8
1f54 f0    71:          dc.b  240 ; Shift+Ctrl+9
1f55      72:
1f55      73:
1f55 bc    74:          dc.b  $bc ; inverse '<'
1f56 9b    75:          dc.b  $9b ; ctrl+6
1f57 be    76:          dc.b  $be ; Ctrl+S
1f58      77: ;
1f58 01    78: IS:      dc.b  1
1f59 01    79: FROM:    dc.b  1
1f5a 01    80: TO:      dc.b  1
1f5b 00    81: VBI_GATE: dc.b  0
1f5c 00    82: KBD_GATE: dc.b  0
1f5d      83: ;-----
1f5d      84: ; BUFFER POINTER
1f5d 0000  85: S_LNPTR: dc.w  0
1f5f      86: ;-----
1f5f      87: CHECKER:
1f5f ad5b1f 88:          lda  VBI_GATE
1f62 d023  89:          bne  L1F63
1f64 d8    90:          CLD
1f65 ad09d2 91:          lda  $D209
1f68 c9ff  92:          cmp  #$FF
1f6a f015  93:          beq  L1F5D
1f6c a209  94:          ldx  #9 ; goona loo
1f6e ad09d2 95: L1F4A:    lda  $D209
1f71 dd4b1f 96:          cmp  HOT_KEY,X
1f74 f024  97:          beq  L1F66
1f76 adfc02 98:          lda  $02FC
1f79 dd4b1f 99:          cmp  HOT_KEY,X
1f7c f01c 100:          beq  L1F66
1f7e ca    101:          DEX
1f7f 10ed 102:          bpl  L1F4A
1f81 bd4b1f 103: L1F5D:    lda  HOT_KEY,X
1f84 8d5c1f 104:          sta  KBD_GATE
1f87 4cffff 105: L1F63:    jmp  $FFFF
1f8a      106: ;

```

```

1f8a      107: JOHN:
1f8a a9ff  108:      lda  #$FF
1f8c 8dfc02 109:      sta  764
1f8f 8d09d2 110:      sta  $D209
1f92 a200  111:      ldx  #$00
1f94 bd4b1f 112:      lda  HOT_KEY,X
1f97 4ca51f 113:      jmp  J_R_B
1f9a      114:
1f9a ad5c1f 115: L1F66:  lda  KBD_GATE
1f9d dd4b1f 116:      cmp  HOT_KEY,X
1fa0 f0e5  117:      beq  L1F63
1fa2 bd4b1f 118:      lda  HOT_KEY,X
1fa5 8d5c1f 119: J_R_B:  sta  KBD_GATE
1fa8 ad581f 120:      lda  IS
1fab 8d591f 121:      sta  FROM
1fae 8e5a1f 122:      stx  TO
1fb1 ba    123:      TSX
1fb2 8a    124:      TXA
1fb3 ae581f 125:      ldx  IS
1fb6 8d041f 126:      sta  STACK
1fb9 ad591f 127:      lda  FROM
1fbc 18    128:      CLC
1bfd 6930  129:      adc  #$30
1fbf ac4a1f 130:      ldy  R1F26
1fc2 991d1f 131:      sta  VIRTUAL,Y      ; L1F26,Y
1fc5 a901  132:      lda  #1
1fc7 8d5b1f 133:      sta  VBI_GATE
1fca d8    134:      CLD
1fcb 78    135:      SEI
1fcc a900  136:      lda  #0
1fce 8d0ed4 137:      sta  $D40E
1fd1 8df21f 138:      sta  L1FBA+1
1fd4 8df31f 139:      sta  L1FBA+2
1fd7 8df51f 140:      sta  L1FBD+1
1fda a940  141:      lda  #$40
1fdc 8df61f 142:      sta  L1FBD+2
1fdf ae581f 143:      ldx  IS
1fe2 ad01d3 144:      lda  $D301
1fe5 8d051f 145:      sta  MY_PORTB
1fe8 29e2  146:      and  #$E2
1fea 0902  147:      ora  #2
1fec 8d01d3 148:      sta  $D301
1fef a000  149:      ldy  #0
1ff1 b9ffff 150: L1FBA:  lda  $FFFF,Y
1ff4 990040 151: L1FBD:  sta  $4000,Y
1ff7 c8    152:      INY
1ff8 d0f7  153:      bne  L1FBA
1ffa eef31f 154:      inc  L1FBA+2
1ffd eef61f 155:      inc  L1FBD+2
2000 adf61f 156:      lda  L1FBD+2
2003 c980  157:      cmp  #$80
2005 d0ea  158:      bne  L1FBA
2007 ad01d3 159:      lda  $D301

```

```

200a 29e2 160:      and  #$E2
200c 0906 161:      ora  #6
200e 8d01d3 162:      sta  $D301
2011 a900 163:      lda  #0
2013 8d00d4 164:      sta  $D400
2016 8d2f02 165:      sta  $022F
2019 8d3620 166:      sta  L1FFE+1
201c 8d3920 167:      sta  L2001+1
201f a9c0 168:      lda  #high $C000      ; o
2021 8d3720 169:      sta  L1FFE+2
2024 a940 170:      lda  #high $4000      ; R
2026 8d3a20 171:      sta  L2001+2
2029 a000 172:      ldy  #0
202b ad01d3 173:      lda  $D301
202e 29fe 174:      and  #$FE
2030 0902 175:      ora  #2
2032 8d01d3 176:      sta  $D301
2035 b900c0 177: L1FFE:      lda  $C000,Y
2038 990040 178: L2001:      sta  $4000,Y
203b c8 179:      INY
203c d0f7 180:      bne  L1FFE
203e ee3720 181:      inc  L1FFE+2
2041 ee3a20 182:      inc  L2001+2
2044 ad3a20 183:      lda  L2001+2
2047 c980 184:      cmp  #$80
2049 d0ea 185:      bne  L1FFE
204b a012 186:      ldy  #$12
204d b9061f 187: L2016:      lda  VECTORS,Y
2050 991802 188:      sta  $0218,Y
2053 88 189:      dey
2054 10f7 190:      bpl  L2016
2056 a012 191:      ldy  #18
2058 b9071f 192:      lda  VECTORS+1,Y
205b 8d0002 193:      sta  $0200
205e b9081f 194:      lda  VECTORS+2,Y
2061 8d0102 195:      sta  $0200+1
2064 ad051f 196:      lda  MY_PORTB
2067 29e2 197:      and  #$E2
2069 0903 198:      ora  #3
206b 8d01d3 199:      sta  $D301
206e a9c0 200:      lda  #$C0
2070 8d0ed4 201:      sta  $D40E
2073 58 202:      CLI
2074 8d0fd4 203:      sta  $D40F
2077 202d22 204:      jsr  CLOSE
207a a210 205:      ldx  #$10
207c a903 206:      lda  #3
207e 9d4203 207:      sta  $0342,X
2081 a908 208:      lda  #8
2083 9d4a03 209:      sta  $034A,X
2086 a900 210:      lda  #0
2088 9d4b03 211:      sta  $034B,X
208b a91d 212:      LDA  # low VIRTUAL

```

```

208d 9d4403 213:      sta  $0344,X
2090 a91f  214:      LDA  # high VIRTUAL
2092 9d4503 215:      sta  $0345,X
2095 2056e4 216:      jsr  $E456
2098 a900  217:      LDA  # low $4000
209a 8d5d1f 218:      sta  S_LNPTR
209d a940  219:      LDA  # high $4000
209f 8d5e1f 220:      sta  S_LNPTR+1
20a2 203822 221:      jsr  B_PUT
20a5 ad01d3 222:      lda  $D301
20a8 29e3  223:      and  #$E3
20aa 0907  224:      ora  #7
20ac 8d01d3 225:      sta  $D301
20af 203822 226:      jsr  B_PUT
20b2 ad051f 227:      lda  MY_PORTB
20b5 8d01d3 228:      sta  $D301
20b8 a900  229:      LDA  # low $8000
20ba 8d5d1f 230:      sta  S_LNPTR
20bd a980  231:      LDA  # high $8000
20bf 8d5e1f 232:      sta  S_LNPTR+1
20c2 203822 233:      jsr  B_PUT
20c5 202d22 234:      jsr  CLOSE
20c8 ad5a1f 235:      lda  TO
20cb 18     236:      CLC
20cc 6930  237:      adc  #$30
20ce ac4a1f 238:      ldy  R1F26
20d1 991d1f 239:      sta  VIRTUAL,Y
20d4 a210  240:      ldx  #$10
20d6 a903  241:      lda  #3
20d8 9d4203 242:      sta  $0342,X
20db a904  243:      lda  #4
20dd 9d4a03 244:      sta  $034A,X
20e0 a900  245:      lda  #0
20e2 9d4b03 246:      sta  $034B,X
20e5 a91d  247:      LDA  # low VIRTUAL
20e7 9d4403 248:      sta  $0344,X
20ea a91f  249:      LDA  # high VIRTUAL
20ec 9d4503 250:      sta  $0345,X
20ef 2056e4 251:      jsr  $E456
20f2 1009  252:      bpl  L210E
20f4 ad5a1f 253:      lda  TO
20f7 8d581f 254:      sta  IS
20fa 4c3f21 255:      jmp  L2194
20fd      256: ;
20fd ad5a1f 257: L210E:  lda  TO
2100 8d581f 258:      sta  IS
2103 ad01d3 259:      lda  $D301
2106 29e3  260:      and  #$E3
2108 0903  261:      ora  #3
210a 8d01d3 262:      sta  $D301
210d a900  263:      LDA  # low $4000
210f 8d5d1f 264:      sta  S_LNPTR
2112 a940  265:      LDA  # high $4000

```

```

2114 8d5e1f 266:      sta  S_LNPTR+1
2117 204022 267:      jsr  B_GET
211a ad01d3 268:      lda  $D301
211d 29e3 269:      and  #$E3
211f 0907 270:      ora  #7
2121 8d01d3 271:      sta  $D301
2124 204022 272:      jsr  B_GET
2127 ad051f 273:      lda  MY_PORTB
212a 0912 274:      ora  #$12
212c 8d01d3 275:      sta  $D301
212f a900 276:      LDA  # low $8000
2131 8d5d1f 277:      sta  S_LNPTR
2134 a980 278:      LDA  # high $8000
2136 8d5e1f 279:      sta  S_LNPTR+1
2139 204022 280:      jsr  B_GET
213c 202d22 281:      jsr  CLOSE
213f 78 282: L2194:      SEI
2140 a900 283:      lda  #0
2142 8d0ed4 284:      sta  $D40E
2145 ad01d3 285:      lda  $D301
2148 29e2 286:      and  #$E2
214a 0902 287:      ora  #2
214c 8d01d3 288:      sta  $D301
214f a900 289:      lda  #0
2151 8d6221 290:      sta  L21B6+1
2154 8d6521 291:      sta  L21B9+1
2157 8d6621 292:      sta  L21B9+2
215a a940 293:      lda  #$40
215c 8d6321 294:      sta  L21B6+2
215f a000 295:      ldy  #0
2161 b90040 296: L21B6:      lda  $4000,Y
2164 99ffff 297: L21B9:      sta  $FFFF,Y
2167 c8 298:      INY
2168 c002 299:      cpy  #2
216a d016 300:      bne  L21D7
216c ad6621 301:      lda  L21B9+2
216f c91f 302:      cmp  #$1F
2171 d00f 303:      bne  L21D7
2173 a086 304:      ldy  #$86
2175 a922 305:      lda  #$22
2177 8d6621 306:      sta  L21B9+2
217a 18 307:      CLC
217b 6940 308:      adc  #$40
217d 8d6321 309:      sta  L21B6+2
2180 d0df 310:      bne  L21B6
2182 c000 311: L21D7:      cpy  #0
2184 d0db 312:      bne  L21B6
2186 ee6321 313:      inc  L21B6+2
2189 ee6621 314:      inc  L21B9+2
218c ad6621 315:      lda  L21B9+2
218f c940 316:      cmp  #$40
2191 317: ;
2191 318: ;

```

```

2191      319: ; This find's the point in the
2191      320: ; where to stick the counter in the line
2191      321: ;-----
2191 d0ce 322:      bne  L21B6
2193 ad01d3 323:      lda  $D301
2196 29e2 324:      and  #$E2
2198 0906 325:      ora  #6
219a 8d01d3 326:      sta  $D301
219d a900 327:      lda  #0
219f 8db221 328:      sta  L2206+1
21a2 8db521 329:      sta  L2209+1
21a5 a940 330:      lda  #$40
21a7 8db321 331:      sta  L2206+2
21aa a9c0 332:      lda  #$C0
21ac 8db621 333:      sta  L2209+2
21af a000 334:      ldy  #0
21b1 b90040 335: L2206:      lda  $4000,Y
21b4 9900c0 336: L2209:      sta  $C000,Y
21b7 c8 337:      INY
21b8 c000 338:      cpy  #0
21ba d0f5 339:      bne  L2206
21bc eeb321 340:      inc  L2206+2
21bf eeb621 341:      inc  L2209+2
21c2 adb621 342:      lda  L2209+2
21c5 c9d0 343:      cmp  #$D0
21c7 d0e8 344:      bne  L2206
21c9 a9d8 345:      lda  #$D8
21cb 8dd821 346:      sta  L222B+2
21ce a958 347:      lda  #$58
21d0 8dd521 348:      sta  L2228+2
21d3 b90058 349: L2228:      lda  $5800,Y
21d6 9900d8 350: L222B:      sta  $D800,Y
21d9 c8 351:      INY
21da d0f7 352:      bne  L2228
21dc eed521 353:      inc  L2228+2
21df eed821 354:      inc  L222B+2
21e2 add521 355:      lda  L2228+2
21e5 c980 356:      cmp  #$80
21e7 d0ea 357:      bne  L2228
21e9 a9ff 358:      lda  #$FF
21eb 8dfc02 359:      sta  $02FC
21ee ae5a1f 360:      ldx  TO
21f1 8e581f 361:      stx  IS
21f4 8e591f 362:      stx  FROM
21f7 ad051f 363:      lda  MY_PORTB
21fa 8d01d3 364:      sta  $D301
21fd ad041f 365:      lda  STACK
2200 aa 366:      TAX
2201 9a 367:      TXS
2202 a900 368:      lda  #0
2204 8d5b1f 369:      sta  VBI_GATE
2207 a9c0 370:      lda  #$C0
2209 8d0ed4 371:      sta  $D40E

```



```

220c 58 372:      CLI
220d 4c871f 373:      jmp  L1F63
2210      374: ;
2210      375: TRAP_RESET:
2210 20ffff 376:      jsr  $FFFF
2213 a95f 377: RUN_LOCAL:  LDA  # low CHECKER
2215 8d2402 378:      sta  $0224
2218 a91f 379:      LDA  # high CHECKER
221a 8d2502 380:      sta  $0224+1
221d a966 381:      LDA  # low CHECK_END+1  ;[CHECK_END
221f 8de702 382:      sta  $02E7
2222 a922 383:      LDA  # high CHECK_END+1  ;[CHECK_END
2224 8de802 384:      sta  $02E7+1
2227 a900 385:      lda  #0
2229 8d5b1f 386:      sta  VBI_GATE
222c 60 387:      rts
222d      388: ;
222d      389: CLOSE:
222d a210 390:      ldx  #$10
222f a90c 391:      lda  #$0C
2231 9d4203 392:      sta  $0342,X
2234 2056e4 393:      jsr  $E456
2237 60 394:      rts
2238      395: ;
2238      396: B_PUT:
2238 a210 397:      ldx  #$10
223a a90b 398:      lda  #$0B
223c 204422 399:      jsr  CIO_WORK
223f 60 400:      rts
2240      401: B_GET:
2240 a210 402:      ldx  #$10
2242 a907 403:      lda  #7
2244 9d4203 404: CIO_WORK:  sta  $0342,X
2247 a900 405:      LDA  # low $4000
2249 9d4403 406:      sta  $0344,X
224c a940 407:      LDA  # high $4000
224e 9d4503 408:      sta  $0345,X
2251 ad5d1f 409:      lda  S_LNPTR
2254 9d4803 410:      sta  $0348,X
2257 ad5e1f 411:      lda  S_LNPTR+1
225a 9d4903 412:      sta  $0349,X
225d 2056e4 413:      jsr  $E456
2260 60 414:      rts
2261      415: ;
2261      416: PAD:
2261 000000 417:      dc.b  0,0,0,0
2265      418: ;
2265      419: ;-----
2265      420: ; END OF MEMORY RESIDENT CODE
2265      421: ;-----
2265      422: CHECK_END:
2265      423: ;-----
2265      424: ;

```

```

2265      425: ;
2265      426: START:
2265 a980 427:      lda  #$80      ;changing the scree
2267 8dc602 428:      sta  $02C6      ; making it blue
226a a9ff 429:      lda  #$FF      ; well let'S make t
226c 8dc502 430:      sta  $02C5      ; so I can see them
226f      431:
226f 20072b 432:      jsr  mypage6
2272      433:
2272 206329 434:      jsr  PRINTSI
2275 7d9b 435:      dc.b  $7D,$9B
2277 209192 436:      dc.b  " .....
229c 20fc20 437:      dc.b  " ü      SnapShot V3.0c
22c1 20fc20 438:      dc.b  " ü      Written By
22e6 20fc20 439:      dc.b  " ü      Tom Hunt & Stephen J. Carden
230b 208192 440:      dc.b  " • .....
232f 9bff 441:      dc.b  $9B,$FF
2331      442:
2331      443:
2331 ad0007 444:      lda  $0700      ; let'S see if it i
2334 c953 445:      cmp  #'S'
2336 f02d 446:      beq  MUX_MAN
2338 c952 447:      cmp  #'R'
233a f029 448:      beq  mux_man      ; it is spartaDOS l
233c 206329 449:      jsr  PRINTSI
233f 9b204d 450:      dc.b  $9B," Must use RealDos or SpartaDos
2361 4ccd29 451:      jmp  START_1
2364 60 452:      rts
2365      453: ;
2365      454: MUX_MAN:
2365 ad01d3 455:      lda  $D301
2368 48 456:      pha
2369 29fe 457:      and  #$FE
236b 8d01d3 458:      sta  $D301
236e a000 459:      ldy  #0
2370 20c6ff 460:      jsr  $FFC6      ; turn the td line
2373 68 461:      pla
2374 8d01d3 462:      sta  $D301
2377 902c 463:      bcc  TDLINE
2379 206329 464:      jsr  PRINTSI
237c 20fc 465:      dc.b  " ü"
237e a0a0a0 466:      dc.b  <+128>," Do Not install tdline2.c
239f fc9bff 467:      dc.b  "ü",$9B,$FF
23a2 4cf323 468:      jmp  TDLINE_2
23a5      469: TDLINE:
23a5 206329 470:      jsr  PRINTSI
23a8 20fc 471:      dc.b  " ü"
23aa d4e4ec 472:      dc.b  <+128>,"Tdline2 Turned off Do not t
23cb fc9b 473:      dc.b  "ü",$9B
23cd 20fc 474:      dc.b  " ü"
23cf e2e1e3 475:      dc.b  <+128>,"back on!
23f0 fc 476:      dc.b  "ü"
23f1 9b 477:      dc.b  $9B

```

```

23f2 ff 478:      dc.b  -1
23f3      479: ;
23f3      480: TDLIN_2:
23f3 206329 481:      jsr  PRINTSI
23f6 20 482:      dc.b  " "
23f7 819292 483:      dc.b  "•,,,,,,,,,,,,,,,,,,,,,,,,,,,,,"
241a 9b 484:      dc.b  $9B
241b ff 485:      dc.b  -1
241c a03f 486:      ldy  #63      ; loading the offse
241e b10a 487: MUX_1A:      lda  ($0A),Y
2420 c920 488:      cmp  #$20      ; looking for the s
2422 f00b 489:      beq  MUX_1B      ; got the space let
2424 c99b 490:      cmp  #$9B      ; nope!! no parm
2426 f004 491:      beq  MUX_2C      ; let'S see if they
2428 c8 492:      INY
2429 4c1e24 493:      jmp  MUX_1A
242c 4c4424 494: MUX_2C:      jmp  MUX_1C
242f      495: ;
242f      496: ;
242f      497: ;
242f      498: ; let'S get one more byte of contab+63,y
242f      499: MUX_1B:
242f c8 500:      INY
2430 a200 501:      ldx  #$00
2432      502: MUX_1CA:
2432 b10a 503:      lda  ($0A),Y ; well got a byte
2434 208226 504:      jsr  UPCASE      ; let'S make sure i
2437 9d1d1f 505:      sta  VIRTUAL,X
243a c99b 506:      cmp  #$9B
243c f006 507:      beq  MUX_1C
243e c8 508:      INY
243f e8 509:      INX
2440 f0f0 510:      beq  MUX_1CA
2442 d0ee 511:      bne  MUX_1CA
2444      512: MUX_1C:
2444 a903 513:      lda  #$03      ; OPTION KEY
2446 cd1fd0 514:      cmp  $D01F      ; CONSOL KEY
2449 f003 515:      beq  MUX_MAN3      ; YEP!!! HELP MENU
244b 4c5124 516:      jmp  MUX_MAN2      ; let'S load the sl
244e 4c9826 517: MUX_MAN3:      jmp  HELP_MENU      ; help menu
2451      518: ;
2451      519: ; prep the multiplexer
2451      520: MUX_MAN2:
2451 a220 521:      ldx  #$20      ; preping the m dev
2453 a903 522:      lda  #3
2455 9d4203 523:      sta  $0342,X
2458 a95e 524:      LDA  # low MDEV
245a 9d4403 525:      sta  $0344,X
245d a926 526:      LDA  # high MDEV
245f 9d4503 527:      sta  $0345,X
2462 a90c 528:      lda  #12
2464 9d4a03 529:      sta  $034A,X
2467 a900 530:      lda  #0

```

```

2469 9d4b03 531:      sta  $034B,X
246c 2056e4 532:      jsr  $E456
246f 1003 533:      bpl  CHT030      ;hay it'S a slave
2471 4c2726 534:      jmp  CHTERR      ;either the master
2474 20e625 535: CHT030:      jsr  XSTAT      ; hay let'S
2477 98 536:      tya      ; let'S move let sl
2478 8d6226 537:      sta  NODE      ; just a holding re
247b 4930 538:      eor  #$30      ; let'S make it int
247d 8d6326 539:      sta  NODE+1      ; saving the slave
2480 4980 540:      eor  #$80
2482 8da124 541:      sta  NODE1      ; let'S put the sla
2485 206329 542:      jsr  PRINTSI
2488 20fc 543:      dc.b  " ü"
248a d9eff5 544:      dc.b  <+128>,"You are Running Slave #"
24a1      545:
24a1 78 546: NODE1:      dc.b  "x"
24a2 a0a0a0 547:      dc.b  <+128>,"      "
24ab fc 548:      dc.b  "ü"
24ac 9b 549:      dc.b  $9B
24ad ff 550:      dc.b  -1
24ae a220 551:      ldx  #$20      ; well let'S close
24b0 206426 552:      jsr  CCLOSE
24b3 ad6326 553:      lda  NODE+1      ; just seeing if it
24b6 c931 554:      cmp  #'1'      ; is 1 to 8
24b8 9007 555:      bcc  BAT_TROUBLE  ; hummm.. this is
24ba c939 556:      cmp  #'9'
24bc b003 557:      bcs  BAT_TROUBLE
24be 4c3c25 558:      jmp  BAT_11      ; alright!! got a
24c1 206329 559: BAT_TROUBLE: jsr  PRINTSI      ; trouble in the mu
24c4 9b9b9b 560:      dc.b  $9B,$9B,$9B
24c7 20fc 561:      dc.b  " ü"
24c9 a0d5ee 562:      dc.b  <+128>," Unexpected Multiplexer Err
24ea fc 563:      dc.b  "ü"
24eb 9b 564:      dc.b  $9B
24ec 20fc 565:      dc.b  " ü"
24ee a0c3e8 566:      dc.b  <+128>," Check Cable's and re-try P
250f fc 567:      dc.b  "ü"
2510 9b 568:      dc.b  $9B
2511 209a92 569:      dc.b  " §"
2535 9b9b 570:      dc.b  $9B,$9B
2537 ff 571:      dc.b  -1
2538 6c0a00 572:      jmp  ($0A)
253b 60 573:      rts
253c      574: BAT_11:
253c 206329 575:      jsr  PRINTSI
253f 208192 576:      dc.b  " •"
2564 20fc 577:      dc.b  " ü"
2566 a0a0cd 578:      dc.b  <+128>," Multiplexer Option Instal
2587 fc 579:      dc.b  "ü"
2588 9b 580:      dc.b  $9B
2589 208192 581:      dc.b  " •"
25af a56a 582:      lda  $6a      ; checking for a ca
25b1 c9a0 583:      cmp  #$a0

```

```

25b3 f004 584:      beq  TAT_12
25b5 4ccd29 585:      jmp  START_1
25b8 60 586:      rts
25b9 587: ;-----
25b9 588: ;
25b9 589: TAT_12:
25b9 206329 590:      jsr  PRINTSI
25bc 20fc 591:      dc.b  " ü"
25be a0cce5 592:      dc.b  <+128>," Left Cartridge or Basic Pr
25df fc 593:      dc.b  "ü"
25e0 9b 594:      dc.b  $9B
25e1 ff 595:      dc.b  -1
25e2 4ccd29 596:      jmp  START_1
25e5 60 597:      rts
25e6 598: ;
25e6 599: ;-----
25e6 600: ;
25e6 601: XSTAT:
25e6 a220 602:      ldx  #$20      ; this is a status
25e8 a90d 603:      lda  #13
25ea 9d4203 604:      sta  $0342,X
25ed a900 605:      lda  #$00
25ef 9d4a03 606:      sta  $034A,X
25f2 9d4b03 607:      sta  $034B,X
25f5 2056e4 608:      jsr  $E456
25f8 adeb02 609:      lda  747
25fb 8d5c26 610:      sta  L747
25fe 60 611:      rts
25ff 612: ;
25ff 613: CHTCLS:
25ff a220 614:      ldx  #$20      ; just some mux xio
2601 a90c 615:      lda  #12
2603 9d4203 616:      sta  $0342,X
2606 2056e4 617:      jsr  $E456
2609 60 618:      rts
260a 619: ;
260a a220 620: CHTXIO:      ldx  #$20
260c 9d4203 621:      sta  $0342,X
260f a95e 622:      LDA  # low MDEV
2611 9d4403 623:      sta  $0344,X
2614 a926 624:      LDA  # high MDEV
2616 9d4503 625:      sta  $0345,X
2619 a900 626:      lda  #0
261b 9d4a03 627:      sta  $034A,X
261e 9d4b03 628:      sta  $034B,X
2621 2056e4 629:      jsr  $E456
2624 3001 630:      bmi  CHTERR
2626 60 631:      rts
2627 98 632: CHTERR:      tya      ; outh!!!
2628 48 633:      pha
2629 a220 634:      ldx  #$20
262b 20ff25 635:      jsr  CHTCLS
262e 68 636:      pla

```

```

262f      637: ;
262f      638: ; well this assume's the file was booted on either
262f      639: ; a non multiplexer slave or a master
262f      640: EXIT:
262f 206329 641:      jsr  PRINTSI
2632 20      642:      dc.b  " "
2633 fc      643:      dc.b  "ü"
2634 ceefa0 644:      dc.b  <+128>,"No Multiplexer Present in S
2655 fc      645:      dc.b  "ü"
2656 9b      646:      dc.b  $9B
2657 ff      647:      dc.b  -1
2658 4ccd29 648:      jmp  START_1
265b 60      649:      rts
265c      650: ;
265c      651: ;data areas that are loaded
265c 0000     652: L747:      dc.b  $00,$00
265e 4d3a9b 653: MDEV:      dc.b  "M:",$9B,$00
2662 0030     654: NODE:      dc.b  0,"0"
2664      655: ;
2664      656: ; generic close
2664      657: CCLOSE:
2664 a90c     658:      lda  #$0C
2666 9d4203 659:      sta  $0342,X
2669 2056e4 660:      jsr  $E456
266c 60      661:      rts
266d      662: ;
266d      663: ;-----;
266d      664: ; GET KEY ROUTEEN      ;
266d      665: ;this routeen get's one key and;
266d      666: ;return it to the accumulator ;
266d      667: ;-----;
266d      668: ;
266d 86b6     669: GETKEY:      stx  $B6
266f 84b7     670:      sty  $B7
2671 207926 671:      jsr  G_FFF
2674 a6b6     672:      ldx  $B6
2676 a4b7     673:      ldy  $B7
2678 60      674:      rts
2679 ad25e4 675: G_FFF:      lda  $E420+5
267c 48      676:      pha
267d ad24e4 677:      lda  $E420+4
2680 48      678:      pha
2681 60      679:      rts
2682      680: ;
2682      681: ;-----;
2682      682: ;CHANGES THE CHARTER TO UPCASE;
2682      683: ;-----;
2682      684: UPCASE:
2682 c961     685:      cmp  #$61
2684 9006     686:      bcc  UPCASE_EXIT
2686 c97a     687:      cmp  #$7A
2688 b002     688:      bcs  UPCASE_EXIT
268a e91f     689:      sbc  #$1F

```

```

268c      690: UPCASE_EXIT:
268c 60    691:          rts
268d      692: ;
268d      693: ;-----;
268d      694: ;CHANGES THE CHARTER TO LOCASE;
268d      695: ;-----;
268d      696: LOCASE
268d c941   697:          cmp    #$41
268f 9006   698:          bcc    LOCASE_EXIT
2691 c95a   699:          cmp    #$5A
2693 b002   700:          bcs    LOCASE_EXIT
2695 691f   701:          adc    #$1F
2697      702: LOCASE_EXIT:
2697 60    703:          rts
2698      704: ;
2698      705: ;
2698      706: ;-----
2698      707: ; help menu section
2698      708: ;-----
2698      709: HELP_MENU
2698 206329 710:          jsr    PRINTSI
269b 7d9b   711:          dc.b   $7D,$9B
269d 919292 712:          dc.b   "....."
26c0 fc    713:          dc.b   "ü"
26c1 a0a0d7 714:          dc.b   <+128>," Welcome to SnapShot hard
26e1 fc    715:          dc.b   "ü"
26e2 9b    716:          dc.b   $9B
26e3 fc    717:          dc.b   "ü"
26e4 a0a0a0 718:          dc.b   <+128>,"      Written By:
2704 fc    719:          dc.b   "ü"
2705 9b    720:          dc.b   $9B
2706 fc    721:          dc.b   "ü"
2707 a0d4ef 722:          dc.b   <+128>," Tom Hunt and Stephen J. Ca
2727 fc    723:          dc.b   "ü"
2728 9b    724:          dc.b   $9B
2729 9a9292 725:          dc.b   "s....."f
274c 9b9b   726:          dc.b   $9B,$9B
274e 9bff   727:          dc.b   $9B,$FF
2750      728: ;
2750 20f828 729:          jsr    PAUSE
2753 206329 730:          jsr    PRINTSI
2756      731: ;
2756 9b    732:          dc.b   $9B
2757 20536e 733:          dc.b   " Snapshot is an (S)hareware progra
277d 205573 734:          dc.b   " Usage: Dn:>path>snap Dn:>path>vir
27a8 205768 735:          dc.b   " Where the parm contains the path
27cd 206f66 736:          dc.b   " of the output file's. If no parm
27f3 207468 737:          dc.b   " the program will default to:",$9B
2811 202044 738:          dc.b   " D1:virtual0.0",$9B,$9B
2822 20536e 739:          dc.b   " Snapshot will show the start asse
2849 207468 740:          dc.b   " the new memtop and a run address
2872 20796f 741:          dc.b   " you wish to do a run command fro
2896 205265 742:          dc.b   " RealDos.",$9B,$9B

```

```

28a1 20416c 743:      dc.b  " Also this version of Snapshot als
28c6 207375 744:      dc.b  " supports the CSS or ILS Multiplex
28eb 9bff 745:      dc.b  $9B,-1
28ed 206d26 746:      jsr  GETKEY
28f0 c99b 747:      cmp  #$9B
28f2 d001 748:      bne  HELP2
28f4 60 749:      rts
28f5      750:
28f5      751:
28f5 4ccd29 752: HELP2:  jmp  START_1
28f8      753: ;
28f8      754: ;
28f8      755: ; a more command
28f8      756: PAUSE:
28f8 206329 757:      jsr  PRINTSI
28fb 5b2d4d 758:      dc.b  "[-More-]",$FF
2904 206d26 759:      jsr  GETKEY
2907 206329 760:      jsr  PRINTSI
290a 7e7e7e 761:      dc.b  "~~~~~",$FF
2914 60 762:      rts
2915 0000 763: POPER:  dc.b  0,0
2917 0000 764: ILS_1:  dc.b  0,0
2919      765: SAVE_CLK:
2919 48 766:      pha
291a 4a 767:      LSR
291b 4a 768:      LSR
291c 4a 769:      LSR
291d 4a 770:      LSR
291e 202429 771:      jsr  R8913
2921 68 772:      pla
2922 290f 773:      and  #$0F
2924 c90a 774: R8913:  cmp  #$0A
2926 b004 775:      bcs  R891B
2928 0930 776:      ora  #$30
292a d002 777:      bne  R891D
292c 6936 778: R891B:  adc  #$36
292e 8d4329 779: R891D:  sta  TAB_DATA
2931 6980 780:      adc  #$80
2933 208b29 781:      jsr  ECHO
2936 ad4329 782:      lda  TAB_DATA
2939 204429 783:      jsr  TAB
293c ae1529 784:      ldx  POPER
293f ad1629 785:      lda  POPER+1
2942 60 786:      rts
2943 00 787: TAB_DATA: dc.b  $00
2944      788: TAB:
2944 ac1829 789:      ldy  ILS_1+1
2947 c001 790:      cpy  #$01
2949 f00c 791:      beq  TAB_1
294b a901 792:      lda  #$01
294d 8d1829 793:      sta  ILS_1+1
2950 ad4329 794:      lda  TAB_DATA
2953 8d1529 795:      sta  POPER

```



```

2956 60    796:      rts
2957      797: TAB_1:
2957 a900  798:      lda  #$00
2959 8d1829 799:      sta  ILS_1+1
295c ad4329 800:      lda  TAB_DATA
295f 8d1629 801:      sta  POPER+1
2962 60    802:      rts
2963      803: BAT_134:
2963      804: ;
2963      805: ;
2963      806: ;
2963      807: ;
2963      808: ;-----
2963      809: ;prints ROUTEEN macro
2963      810: ;this is a slow print to the screen
2963      811: ;usage jsr  prints
2963      812: ; .byte $9b," string to be printed",$ff
2963      813: ;-----
2963      814: ;
2963 68    815: prints:      pla
2964 85b0  816:      sta  $b0
2966 68    817:      pla
2967 85b1  818:      sta  $b1
2969 a001  819:      ldy  #$01
296b b1b0  820: .a:      lda  ($b0),y
296d c9ff  821:      cmp  #$ff
296f f009  822:      beq  .b
2971 208b29 823:      jsr  sprint
2974 208429 824:      jsr  .c
2977 4c6b29 825:      jmp  .a
297a 208429 826: .b:      jsr  .c
297d a5b1  827:      lda  $b1
297f 48    828:      pha
2980 a5b0  829:      lda  $b0
2982 48    830:      pha
2983 60    831:      rts
2984 e6b0  832: .c:      inc  $b0
2986 d002  833:      bne  .d
2988 e6b1  834:      inc  $b1
298a 60    835: .d:      rts
298b      836: ;
298b      837: ;
298b      838: ;
298b      839: ;-----
298b      840: ; PRINT BYTE
298b      841: ;-----
298b      842: echo
298b 85b8  843: sprint:      sta  $b8
298d a904  844:      lda  #$04
298f 8d4a03 845:      sta  $034a
2992 86b6  846:      stx  $b6
2994 84b7  847:      sty  $b7
2996 a206  848:      ldx  #$06

```

```

2998 bd01e4 849:          lda  $e400+1,x
299b 85b3 850:          sta  $b3
299d bd00e4 851:          lda  $e400,x
29a0 85b2 852:          sta  $b2
29a2 a5b8 853:          lda  $b8
29a4 20b329 854:          jsr  .a
29a7 a6b6 855:          ldx  $b6
29a9 a4b7 856:          ldy  $b7
29ab a90c 857:          lda  #$0c
29ad 8d4a03 858:          sta  $034a
29b0 a5b8 859:          lda  $b8
29b2 60 860:          rts
29b3      861: ;
29b3 e6b2 862: .a:          inc  $b2
29b5 6cb200 863:          jmp  ($b2)
29b8 60 864:          rts
29b9      865:
29b9      866: ;
29b9      867: ;
29b9      868: ;
29b9      869: ;
29b9      870: CHECK_POINT:
29b9 a000 871:          ldy  #$00
29bb b91d1f 872: CHKPOI:      lda  VIRTUAL,Y
29be c92e 873:          cmp  #'.'
29c0 f006 874:          beq  CHKPI
29c2 c8 875:          INY
29c3 c000 876:          cpy  #$00
29c5 d0f4 877:          bne  CHKPOI
29c7 60 878:          rts
29c8      879: CHKPI
29c8 88 880:          dey
29c9 8c4a1f 881:          sty  R1F26
29cc 60 882:          rts
29cd      883: ;
29cd      884: ;
29cd      885: START_1
29cd 20b929 886:          jsr  CHECK_POINT
29d0 ac4a1f 887:          ldy  R1F26
29d3 c8 888:          INY
29d4 c8 889:          INY
29d5 ad6326 890:          lda  NODE+1
29d8 991d1f 891:          sta  VIRTUAL,Y
29db ad2402 892:          lda  $0224
29de 8d881f 893:          sta  L1F63+1
29e1 ad2502 894:          lda  $0224+1
29e4 8d891f 895:          sta  L1F63+2
29e7 a05f 896:          LDY  # low CHECKER
29e9 a21f 897:          LDX  # high CHECKER
29eb a907 898:          lda  #7
29ed 205ce4 899:          jsr  $E45C
29f0 206329 900:          jsr  PRINTSI
29f3 20 901:          dc.b  " "

```

```

29f4 819292 902:      dc.b  "•"
2a17 9b 903:      dc.b  $9B
2a18 20fc 904:      dc.b  "ü"
2a1a a0c1f3 905:      dc.b  <+128>," Assembled at____$
2a2d ff 906:      dc.b  -1
2a2e a91f 907:      LDA  # high BEGIN
2a30 201929 908:      jsr  SAVE_CLK
2a33 a900 909:      LDA  # low BEGIN
2a35 201929 910:      jsr  SAVE_CLK
2a38 206329 911:      jsr  PRINTSI
2a3b a0a0a0 912:      dc.b  <+128>,"      "
2a45 fc 913:      dc.b  "ü"
2a46 9b 914:      dc.b  $9B
2a47 20fc 915:      dc.b  "ü"
2a49 a0cee5 916:      dc.b  <+128>," New Memlow_____$
2a5c ff 917:      dc.b  -1
2a5d 918:
2a5d a922 919:      LDA  # high CHECK_END+1
2a5f 201929 920:      jsr  SAVE_CLK
2a62 a966 921:      LDA  # low CHECK_END+1
2a64 201929 922:      jsr  SAVE_CLK
2a67 206329 923:      jsr  PRINTSI
2a6a a0a0a0 924:      dc.b  <+128>,"      "
2a74 fc 925:      dc.b  "ü"
2a75 9b 926:      dc.b  $9B
2a76 20 927:      dc.b  " "
2a77 fc 928:      dc.b  "ü"
2a78 a0d2e5 929:      dc.b  <+128>," Re-init Vbi at____$
2a8b ff 930:      dc.b  -1
2a8c a922 931:      LDA  # high RUN_LOCAL
2a8e 201929 932:      jsr  SAVE_CLK
2a91 a913 933:      LDA  # low RUN_LOCAL
2a93 201929 934:      jsr  SAVE_CLK
2a96 206329 935:      jsr  PRINTSI
2a99 a0a0a0 936:      dc.b  <+128>,"      "
2aa3 fc 937:      dc.b  "ü"
2aa4 9b 938:      dc.b  $9B
2aa5 20 939:      dc.b  " "
2aa6 9a9292 940:      dc.b  "§"
2ac9 9b9b 941:      dc.b  $9B,$9B
2acb ff 942:      dc.b  -1
2acc 943:
2acc 944:
2acc 945:
2acc 946: ;-----
2acc 947: ; This is important code
2acc 948: ;-----
2acc a966 949:      LDA  # low CHECK_END+1 ;[C
2ace 8de702 950:      sta  $02E7
2ad1 a922 951:      LDA  # high CHECK_END+1 ;[C
2ad3 8de802 952:      sta  $02E7+1
2ad6 a50c 953:      lda  $0C
2ad8 8d1122 954:      sta  TRAP_RESET+1

```

```

2adb a50d 955:          lda  $0C+1
2add 8d1222 956:          sta  TRAP_RESET+2
2ae0 a910 957:          LDA   # low TRAP_RESET
2ae2 850c 958:          sta  $0C
2ae4 a922 959:          LDA   # high TRAP_RESET
2ae6 850d 960:          sta  $0C+1
2ae8 a012 961:          ldy  #$12
2aea b91802 962: L2353:      lda  $0218,Y
2aed 99061f 963:          sta  VECTORS,Y
2af0 88 964:          dey
2af1 10f7 965:          bpl  L2353
2af3 a012 966:          ldy  #$12
2af5 ad0002 967:          lda  $0200
2af8 99071f 968:          sta  VECTORS+1,Y
2afb ad0102 969:          lda  $0200+1
2afe 99081f 970:          sta  VECTORS+2,Y
2b01 a960 971:          lda  #$60
2b03 8d001f 972:          sta  CHECK_BEG
2b06 60 973:          rts
2b07 974: ;
2b07 975: ;          .org $06F9
2b07 976: ;
2b07 977: ; Had to do this because xasm does not like to compile les
2b07 978: ; This was a way to move the info into page 6 with extra ju
2b07 979: ;
2b07 980: mypage6:
2b07 a000 981:          ldy  #$00
2b09 b9152b 982: .a:      lda  .data,y
2b0c 99f906 983:          sta  $6f9,y
2b0f c8 984:          iny
2b10 c007 985:          cpy  #$07
2b12 d0f5 986:          bne  .a
2b14 60 987:          rts
2b15 988:
2b15 04534e 989: .data:      dc.b  $04,"SNAP"
2b1a 8a1f 990:          dc.w  JOHN
2b1c 991:
2b1c 992:
2b1c 993: ;
2b1c 994: win_end:      ds.b  0
2b1c 995: ;
2b1c 996: ;
2b1c 997: ;
2b1c 998:

```

End assembly: no errors