

!A

*** End of Pass 1

*** End of Pass 2

```
0800      1          ttl "Memory Move Source Code, MM.L"
0800      2      ;
0800      3      ;
0800      4      ; MM.L
0800      5      ;
0800      6      ;
0800      7      ; Memory Move Source Code
0800      8      ;
0800      9      ; 2024 February 14
0800     10      ;
0800     11      ;
0800     12      ; DOS 4.5, Build 06
0800     13      ;
0800     14      ; 2024 February 14
0800     15      ;
0800     16      ;
0800     17      ; Start of Source Code: 0x4000
0800     18      ; Start of Symbol List: 0x7800
0800     19      ;
0800     20      ;
0800     21      ; Copyright (c) 2024 February 14 by
0800     22      ; Walland Philip Vrbancic Jr
0800     23      ;
0800     24      ; 6223 East Peabody Street
0800     25      ; Long Beach, California 90808
0800     26      ; Unitied States of America
0800     27      ;
0800     28      ; All Rights Reserved
0800     29      ;
0800     30      ; This software is the confidential and
0800     31      ; proprietary intellectual property of
0800     32      ; Walland Philip Vrbancic Jr
0800     33      ;
0800     34      ;
0800     35      ; CALL MM,S%,D%,N%
0800     36      ;
0800     37      ;     FA = Entry address of routine
0800     38      ;
0800     39      ;     S% = Source address
0800     40      ;
0800     41      ;         Required in command line
0800     42      ;
0800     43      ;     D% = Destination address
0800     44      ;
0800     45      ;         Required in command line
0800     46      ;
0800     47      ;     N% = Number of bytes to move
0800     48      ;
0800     49      ;         Required in command line
0800     50      ;
0800     51      ; This code will execute at any address.
0800     52      ;
0800     53      ;
00EC     54  VAR%ADR  epz $EC
00EE     55  N%VAL    epz $EE
0800     56      ;
00FA     57  S%PTR    epz $FA
00FC     58  D%PTR    epz $FC
0800     59      ;
0800     60          enz
```

```

0800          61 ;
0000          62 ZERO      equ $00
0800          63 ;
DEBE          64 CHKCOM    equ $DEBE
DFE3          65 PTRGET    equ $DFE3
0800          66 ;
0800          67 ;
0900          68          org $900
0900          69          obj $900
0900          70          usr
0900          71 ;
0900          72 ;
0900          73 ; Get S% variable address and start address.
0900          74 ;
0900 20 BE DE    75          jsr CHKCOM
0903 20 E3 DF    76          jsr PTRGET
0906          77 ;
0906 85 EC      78          sta VAR%ADR
0908 84 ED      79          sty VAR%ADR+1
090A          80 ;
090A A0 01      81          ldy #1
090C          82 ;
090C B1 EC      83          lda (VAR%ADR),Y
090E 85 FA      84          sta S%PTR
0910          85 ;
0910 88         86          dey
0911          87 ;
0911 B1 EC      88          lda (VAR%ADR),Y
0913 85 FB      89          sta S%PTR+1
0915          90 ;
0915          91 ;
0915          92 ; Get D% variable address and destination address.
0915          93 ;
0915 20 BE DE    94          jsr CHKCOM
0918 20 E3 DF    95          jsr PTRGET
091B          96 ;
091B 85 EC      97          sta VAR%ADR
091D 84 ED      98          sty VAR%ADR+1
091F          99 ;
091F A0 01     100          ldy #1
0921          101 ;
0921 B1 EC      102          lda (VAR%ADR),Y
0923 85 FC      103          sta D%PTR
0925          104 ;
0925 88         105          dey
0926          106 ;
0926 B1 EC      107          lda (VAR%ADR),Y
0928 85 FD      108          sta D%PTR+1
092A          109 ;
092A          110 ;
092A          111 ; Get N% variable address and number of bytes to move.
092A          112 ;
092A 20 BE DE    113          jsr CHKCOM
092D 20 E3 DF    114          jsr PTRGET
0930          115 ;
0930 85 EC      116          sta VAR%ADR
0932 84 ED      117          sty VAR%ADR+1
0934          118 ;
0934 A0 01      119          ldy #1
0936          120 ;
0936 B1 EC      121          lda (VAR%ADR),Y

```

```
0938 85 EE      122      sta N%VAL
093A           123      ;
093A 88         124      dey
093B           125      ;
093B B1 EC      126      lda (VAR%ADR),Y
093D 85 EF      127      sta N%VAL+1
093F           128      ;
093F           129      ;
093F           130      ; Move data from source to destination.
093F           131      ;
093F B1 FA      132      ^1      lda (S%PTR),Y
0941 91 FC      133      sta (D%PTR),Y
0943           134      ;
0943 A5 EE      135      lda N%VAL
0945 D0 06      136      bne >2
0947           137      ;
0947 A5 EF      138      lda N%VAL+1
0949 F0 04      139      beq >3
094B           140      ;
094B C6 EF      141      dec N%VAL+1
094D           142      ;
094D C6 EE      143      ^2      dec N%VAL
094F           144      ;
094F 05 EE      145      ^3      ora N%VAL
0951 F0 09      146      beq >4
0953           147      ;
0953 C8         148      iny
0954 D0 E9      149      bne <1
0956           150      ;
0956 E6 FB      151      inc S%PTR+1
0958 E6 FD      152      inc D%PTR+1
095A           153      ;
095A D0 E3      154      bne <1      ; always taken
095C           155      ;
095C 60         156      ^4      rts
095D           157      ;
095D           158      ;
```

BSAVE MM,A\$0900,B,L\$005D

```
095D           159      usr MM
095D           160      ;
095D           161      ;
095D           162      stt "MM Symbol Table"
095D           163      ;
095D           164      ;
095D           165      end 111
```

*** End of Assembly

Symbol List starts at 0x7800, ends at 0x785A, used 0x005A, remaining 0x3F46

Symbols unsorted:

VAR%ADR	00EC	N%VAL	00EE	S%PTR	00FA	D%PTR	00FC	ZERO	0000
CHKCOM	DEBE	PTRGET	DFE3						

Symbols alphabetically sorted:

CHKCOM	DEBE	D%PTR	00FC	N%VAL	00EE	PTRGET	DFE3	S%PTR	00FA
VAR%ADR	00EC	ZERO	0000						

Symbols numerically sorted:

ZERO	0000	VAR%ADR	00EC	N%VAL	00EE	S%PTR	00FA	D%PTR	00FC
CHKCOM	DEBE	PTRGET	DFE3						