UNPL - 1080

**Bell Laboratories** Cover Sheet for Technical Memorandum

*The information contained herein is for the use of employees of Bell Laboratories and is not for publication. (See GEI 13.9-3)*

Title- **UNIX facilities for the Development of Programs written in Macro-11**

Date- **December 5, 1975**

TM- **75-8234-14**

Other Keywords-


| Author | Location | Extension | Charging Case- **49170-420** |
|---|---|---|---|
| **P. D. Jensen** | **MH 2D-443** | **7498** | Filing Case- **40952-1** |



*ABSTRACT*

This memo describes the initial set of facilities available under UNIX for use in developing programs written in the DOS Macro-11 language, and BOS and BOS application programs in particular. The facilities include a Harvard version of the Macro-11 assembler and linker, a utility to read and write DOS PIP format tapes, and a utility to convert between DOS and UNIX representations of a load module. Possible extensions to these basic facilities are noted.


| Pages Text 5 | Other 7 | Total 12 | |
|---|---|---|---|
| No. Figures 1 | No. Tables 0 | No. Refs. 1 | |



SEE REVERSE SIDE FOR DISTRIBUTION LIST

BELL TELEPHONE LABORATORIES, INC.

TM-75-8234-14

DISTRIBUTION
(REFER GEI 13.9-3)

COMPLETE MEMORANDUM TO

CORRESPONDENCE FILES

OFFICIAL FILE COPY
PLUS ONE COPY FOR
EACH ADDITIONAL FILING
CASE REFERENCED

DATE FILE COPY
(FORM E-1328)

10 REFERENCE COPIES

*ALBRECHT,ERWIN H
*BAYER,DOUGLAS L
<BLINN,JAMES C
*BRENZA,KENNETH J
*CARDOZA,WAYNE M
<CRUM,TED A
DEL RIESGO,C J
DICKERT,L S JR
*DUFF,THOMAS G
*FOLEY,GEORGE E
<FREEMAN,K GLENN
FRITSCH,JAMES T
GIUNTA,JOHN A
GLOVER,MRS S G
<HANSEN,MRS G J
*HATTER,WAYNE E JR
*HOLTMAN,JAMES P
HOWSON,L
HUNT,J M III
<KEESE,W M
<KELLY,L J
<KERNIGHAN,BRIAN W
KESSELL,DAVID L
*KLEIN,MISS R L
*LEKKOS,A A
*LOYD,D G
LUDERER,GOTTFRIED W R
*LYCKLAMA,HEINZ
MARANZANO,JOSEPH F
MARTIN,R L
<MORGAN,S P
*REHERT,ALLEN F
*RHODES,S P
<RITACCO,J E
*ROBINSON,HERMAN E
*SIX,FREDERICK B
*SOBIN,DAVID L
TAGUE,BEFKLEY A
*UNDERWOOD,R W
WATSON,D S
40 NAMES

COVER SHEET ONLY TO

CORRESPONDENCE FILES

4 COPIES PLUS ONE

COVER SHEET ONLY TC

COPY FOR EACH FILING
CASE

ACKERMAN,A F
AHERN,PETER L
AHO,A V
AHRENS,RAINER B
ALBERTS,BARBARA A
ALCALAY,DAVID
ALLISON,CHARLES E
ALMQUIST,M L JR
ALT,MISS DOROTHY L
AMOSS,JOHN J
AMRON,I
ANDERSON,M M
ANDERSON,MS K J
ANTOLICK,DAVID R
ARCHER,RUSSELL E JR
ARNOLD,GEORGE W
ARNOLD,S L
ARTHURS,EDWARD
ATAL,B S
AVERILL,R M JR
BACCASH,MISS J M
BADURA,DENNIS C
BASEIL,RICHARD J
BAUGH,C R
<BECKER,R A
BERING,D E
BERNSTEIN,LAWRENCE
BERTH,R P
BEYER,JEAN-DAVID
BICKFORD,N B
BILOWOS,RICHARD M
BIRCHALL,R H
BIREN,MRS IRMA B
BISHOP,MISS V L
BLAZIER,S D
BLUM,MRS MARION
BLY,JOSEPH A
BODEN,F J
BONANNI,LORENZO E
BOURNE,STEPHEN R
BOWERS,J L
BOWYER,L RAY
BOYCE,W M
BRANDT,K P
BRANDT,RICHARD B
BROWN,COLIN W
BROWN,W STANLEY
BROWN,WILLIAM R
BUCHANAN,D N E
BULLEY,RAYMOND M
BURNETTE,W A
BYRNE,EDWARD R
BZOWY,D E
<CALESSO,GIULIO L
CAMPBELL,STEPHEN T
CANADAY,RUDD H
CANNON,T B
CARLSON,MRS H V

COVER SHEET ONLY TO

CARRAN,J H
CARR,DAVID C
CASPERS,MRS BARBARA E
CAVINESS,JOHN D
CHAFFEE,N F
CHAMBERS,J M
CHAMBERS,MRS B C
CHEN,EDWARD
CHEN,STEPHEN
CHERRY,MS L L
CHIANG,T C
CHODROW,MARK M
CHRISTENSEN,C
CHRIST,C W JR
CLAYTON,D F
CLOUTIER,J E
COBEN,ROBERT M
COHEN,HARVEY
COKE,MISS E U
<COLE,LOUIS M
COLE,M O
COLLUM,DONALD J
COOK,THOMAS J
COOPER,A E
COPP,DAVID H
COREY,D A
CORNELL,R G
COSTELLO,PETER E
COSTON,WALTER P
CRAGUN,D W
CRANE,ROGERICK F
CRUME,LARRY L
CUILWIK,ANTHONY
CUNNINGHAM,STEPHEN J
D ANDREA,MRS LOUISE A
D STEFAN,D J
DAVIS,D R
DAVIS,R D
DE JAGER,D S
DESENDORF,JUDITH
DEUTSCH,DAVID N
DICKMAN,B N
DIMMICK,JAMES O
DOLAN,MRS MARIE T
DOLOTTA,T A
DONNELLY,MISS MARY M
DONOFRIO,L J
DOWDEN,D C
DOWD,PATRICK G
DRAKE,MRS I
DRUMMOND,R E
DRYBURGH,MRS P F
DUDLEY,MRS E H
DWYER,T J
EDDY,MICHAEL R
EDELSON,D
EDMUNDS,T W
EIGEN,D
EITELBACH,DAVID L
ELY,T C
ERDLE,K W

COVER SHEET ONLY TO

ERRICHIELLO,PHILIP M
ESSERMAN,ALAN R
ESTOCK,R G
FABISCH,MICHAEL P
FACTOR,R M
FARBANISH,R S
FARGO,GEORGE A
FEDER,J
FELDMAN,STUART I
FELS,ALLEN M
FELTON,W A
FIORE,MRS RHODA J
FISCHER,H B
FLANDRENA,R J
FLEISCHER,HERBERT I
FLYNN,MS M L
FOCARILE,J P
FONG,KENNETH T
FORTNEY,V J
FOUNTOUKIDIS,A
FOX,PHYLLIS
FOX,R T
FOY,J C
FRANK,H G
FRANK,MISS A J
FRANK,RUDOLPH J
FRANZ,ANN M
FRASER,A G
FREEDMAN,M I
FREEMAN,R DON
FREIDENREICH,MRS B
FRETWELL,LYMAN J
FROST,H BONNELL
FULTON,ALAN W
GALLANT,R J
GANNON,T F
GARAY,D J
GARCIA,R F
GATES,G W
GAY,FRANCIS A
GEARY,M J
GEERS,THOMAS J JR
GEFNER,JAMES R
GERARD,ALLAN
GERGOWITZ,E B
GEYLING,F T
GIBB,KENNETH R
GIBSON,H T JR
GIBSON,T A
GIMPEL,JAMES F
GITHENS,JOHN A
GLASER,W A
GLASSER,ALAN L
GLUCK,F
GOGUEN,MS NANCY
GOLABEK,MISS R
GOLDSMITH,L D
GOLDSTEIN,A JAY
GORMAN,JAMES E
GRAHAM,R L
GRANT,GEORGE E

COVER SHEET ONLY TO

GRAVEMAN,R F
GRAYSON,C F JR
GREENBAUM,H J
GREENHALGH,H WAIN
GROSS,ARTHUR G
GRUCHOWSKI,M P
GUERRIERO,JOSEPH R
GUIDI,PIER V
HAENSCHKE,DETLEV G
HAFER,E H
HAGGERTY,JOSEPH F
HAIGHT,R C
HALE,A L
HALL,ANDREW L JR
HALL,HENRY I
HALL,JOE T
HALL,MILTON S JR
HALL,W G
HALPIN,THOMAS
HAMILTON,PATRICIA
HANNAH,JUDY R
HARKNESS,C J
HARRINGTON,T DORSEY
HARRISON,NEAL T
HARRIS,CHARLES S
HARUTA,K
HAUSE,A D
HEATH,SIDNEY F III
HEDRICK,MISS ELLEN L
HEIDER,B H
HELD,RICHARD W
HERGENHAN,C B
HEROLD,JOHN W
HESS,MILTON S
HESTER,S D
HINDERLITER,R G
HOALST,BLAINE C
HONIG,W L
HOWARD,PHYLLIS A
HOYT,WILLIAM F
<HUMCKE,D J
HUNNICUTT,CHARLES F
HUPKA,MRS FLORENCE
HYMAN,B
IERLEY,W H
IFFLAND,FREDERICK C
IMAGNA,CLYDE P
IPPOLITI,O D
IRVINE,M M
IRWIN,G C
IVIE,EVAN L
JACKOWSKI,D J
JACKSON,JAMES H
JACOBS,H S
JAKUBEK,RAYMOND J
JARVIS,JOHN F
JENSEN,PAUL L
JESSOP,WARREN H
JOHNSON,STEPHEN C
JOHNSTON,WALTER E JR
JONES,B R
...
532 TOTAL

* NAMED BY AUTHOR > CITED AS REFERENCE < REQUESTED BY READER (NAMES WITHOUT PREFIX WERE SELECTED USING THE AUTHOR'S SUBJECT OR ORGANIZATIONAL SPECIFICATION AS GIVEN BELOW)

MERCURY SPECIFICATION.....

COMPLETE MEMO TO:
4561-SUP 4133-SUP 8234-SUP

COVER SHEET TO:
8234

COPLAS = COMPUTING/PROGRAMMING LANGUAGES/ASSEMBLY
UNNL* = UNIX/NEWSLETTER,MEETING ANNOUNCEMENTS,REPORTS
UNPL* = UNIX/PROGRAMMING LANGUAGES

---

TO GET A COMPLETE COPY:

- BE SURE YOUR CORRECT ADDRESS IS GIVEN ON THE OTHER SIDE.
- FOLD THIS SHEET IN HALF WITH THIS SIDE OUT AND STAPLE.
- CIRCLE THE ADDRESS AT RIGHT. USE NO ENVELOPE.

JENSEN,P D
MH 2D443

TM-75-8234-14
TOTAL PAGES 12

PLEASE SEND A COMPLETE COPY TO THE ADDRESS SHOWN ON THE OTHER SIDE
NO ENVELOPE WILL BE NEEDED IF YOU SIMPLY STAPLE THIS COVER SHEET TO THE COMPLETE COPY.
IF COPIES ARE NO LONGER AVAILABLE PLEASE FORWARD THIS REQUEST TO THE CORRESPONDENCE FILES.

Bell Laboratories

Subject: **UNIX facilities for the Development of Programs written in Macro-11**
Case- 49170-420 -- File- 40952-1

date: **December 5, 1975**

from: **P. D. Jensen**

TM: **75-8234-14**

*MEMORANDUM FOR FILE*

## 1. Introduction

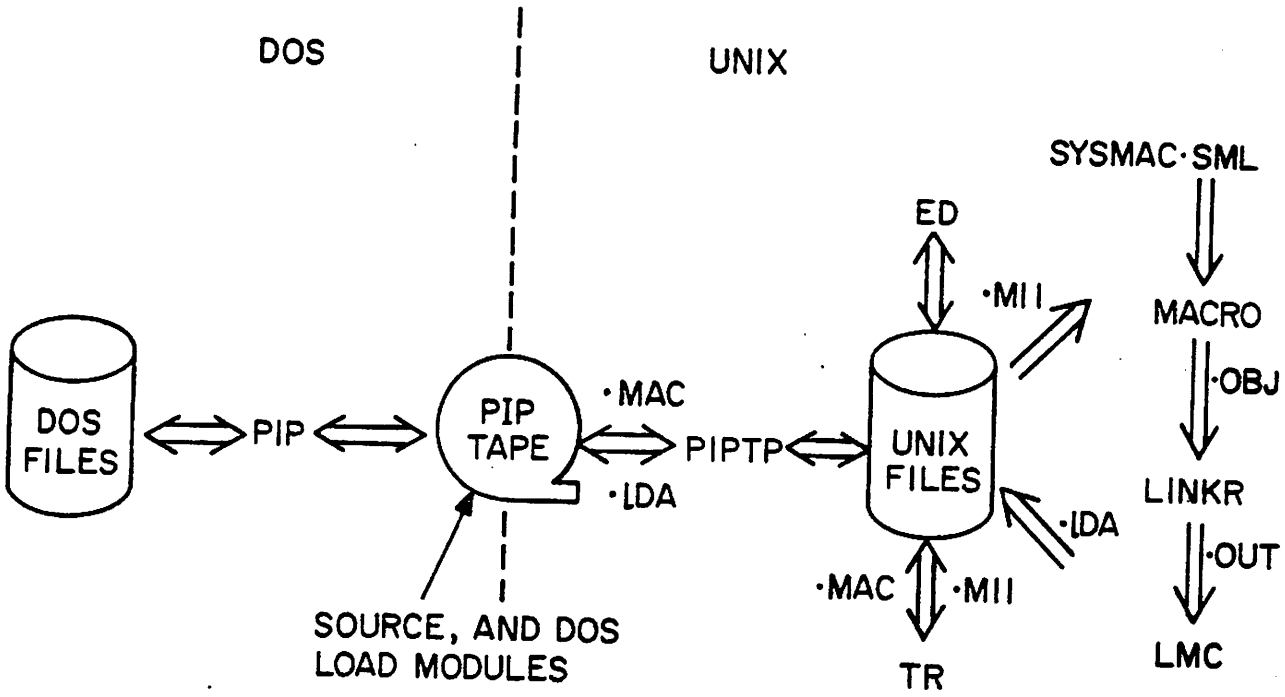
The basic single user facilities provided by DOS for program development and documentation are somewhat restrictive, especially where several PDP-11 machines are being used by a group of people. In order to provide the more abundant facilities available under UNIX, eventually including the C language, a means of assembling and linking Macro-11 programs intended for DOS or BOS environments has been provided under UNIX.

An early version of the Macro-11 assembler and Linker, modified to run under UNIX and produce UNIX executable code, is now available.* In addition, a minimal set of utilities to get back and forth to a DOS environment has been written, as well as a specialized interface for use with BOS based projects.

This memo describes those facilities, the known incompatibilities and restrictions, and notes possible extensions.

The basic flow of programs through these facilities is shown in figure 1.

*Developed originally by R. Bowering for use by DEC, and installed by him on Harvards UNIX. Modified for symbol table output and extended use of UNIX facilities by Harvard.




UNIX DEVELOPMENT FOR MACRO-11 PROGRAMS

FIGURE 1

Source programs may be transported to UNIX files by a DOS format PIP tape, edited (perhaps translated to lower case for ease of editing), assembled using an appropriate system macro file, linked, then converted back to a DOS format load module. The updated source and the load modules may then be placed on a PIP format tape for transportation back to a DOS environment. The object files, although in the same basic format, are not generally compatible with those created under DOS.

The next section discusses the use of these facilities (Piptp, Macro, Linkr, and Lmc), plus a specialized interface with BOS (Blip). Hints on smooth conversion procedures are given in section 3, and planned extensions plus user manual pages for the current facilities are given in the appendices.

## 2. Facilities

*A. Macro Assembler*

Assembling under "Macro" is very nearly source compatible with DOS version 9 Macro-11 except for a few options (notably global symbols of .enabl/.dsabl). The source file may be in either upper or lower case, although editing would be more painful if upper case is used, especially from upper case terminals (remember UNIX maps upper case to lower case on input for such terminals). However, character constants are assembled as they are written, but may be forced to upper case by use of a .dsabl lc to map all input to upper case thereafter (including the listing). The default of this option will probably be reversed in the future.

The amount of symbol table space can be extended n*1024 words by the -xs:n switch. If n does not include a period, it is taken as octal. A listing generally requires a symbol table extension of 4K. Heavy use of macros may require 8 to 12K.

The file /usr/lib/sysmac.sml is accessed by the standard .mcall directive. One of the following macro libraries should be thus named. If a different macro file is needed, the -sm:fname switch will read the macros from fname.sml in the current directory. Existing libraries in /usr/lib are:

- The standard DOS system macros for use in execution under DOS. (dosmac.sml)
- A simple set of macros for the UNIX system calls. (umac.sml)
- Macros for execution under BOS. These include macros providing structured control statements as well as the BOS system calls.* (bosmac.sml)

Additional macro packages will probably be created to simulate other environments under UNIX. See the Appendix on enhancements.

DOS Psect attributes RO (read only) and RW (read write) correspond to prv (private) and shr (shared) under UNIX; the I (instruction) and D (Data) space attributes have been replaced by UNIX ins, dat, and bss segment attributes. *However*, none of the above attributes can be conveyed by the DOS load module, so they should be avoided. See Linkr for a further discussion of .psect problems.

The switches -nl, -li, -en, -ds do not appear to work correctly, although they are accepted. The corresponding directives do work properly.

The output of macro is a DOS binary formatted file with block type codes similar to DOS 9, except that code and relocation are combined into one block rather than two, apparently to handle RSX-11's more complex relocation. Thus object comparisons under DOS will fail.

*B. Linkr*

Linking, if done with care, can duplicate the memory allocation of the DOS linker. Linkr does *not* support overlays however.

Psects and the way in which memory is allocated provide the largest potential incompatibility (especially when comparisons between DOS and UNIX created programs are needed). Named psects are allocated in the order encountered rather than alphabetically as in DOS version 9. This may be overcome by providing a dummy program which simply lists the psects and csects in alphabetical order, along with their respective attributes. This should be the first program linked. Blank csects are allocated as encountered rather than concatenated as in DOS 9. The -dc switch (DOS compatible) forces blank csects to be global and concatenated. These two measures should give the same memory allocation as DOS 9, when that is required.

---

*"Macros for structured Programming under Macro-11 - File 39499-41", Memo for File by W. M. Cardoza dated Nov 7, 1975.

The bottom and top load switches are available, but should not be used if the program is to execute under UNIX since UNIX requires an entry point of zero. The combination of an initialized asect and a top or bottom load can not be handled, and causes a wipeout of the Linkr.

ODT and DDT programs for use under UNIX are not currently available, and the ODT transfer address will *not* be conveyed to the DOS load module. Instead, put a jump in your program to ODT, and link it in normally.

Since no library facility exists, the number of modules which can be linked is severely limited by the argument passing of the shell, which usually allows about 45 names. Some type of indirection will be provided as soon as possible.

The output of the linker is a UNIX .out file with two or three words of the 8-word header used in non-standard fashion.* Type 407 files are normally produced and all psects will be allocated in the UNIX data segment (it's read/write), with text and bss segments empty. Word 7 will be the module's entry point (which must be zero for successful execution under UNIX), and word 6 (a UNIX spare) will contain the load point if the top or bottom load switch was used. No space will be reserved in the data segment to reflect a non-zero load point. Thus "load point" bytes are injected before the data segment when creating a core image.

Type 410 files are produced if read-only psects occurred, and 411 files if separate I and D space was requested. Neither can currently be mapped to a DOS load module, and the 411 code is not standard UNIX. Use the -b:0 switch to force a type 407 file if necessary.

*C. Load Module Conversion*

If a module is to be loaded by a DOS loader, the UNIX .out file produced by the Linkr may be converted to the format of a DOS load module by the Lmc command. The DOS communications directory is complete for non-overlay programs with the exception of the ODT transfer address and the module identification (.ident of first module linked). The time and date reflect that of the last modification of the UNIX file. Only type 407 files may be converted.

Due to the different manner in which the formatted binary records are formed, a DOS load module created by Lmc will not directly compare with one created by the DOS linker. In order to do binary compares, both must be mapped to UNIX .out format. Happily, Lmc will also convert from a DOS load module to a UNIX .out file. Note that since the output of the UNIX Linkr usually contains a symbol table, this table must first be stripped (see the UNIX Strip command or the -ns switch of Linkr) from the UNIX .out file before comparing with the output of Lmc.

*D. PIP Tape Utility*

The Piptp command may be used to read, write, or list a DOS PIP format mag tape. Source and load module files are usefully transferred. The object files produced by Macro under UNIX may be transferred, but they will not be accepted by the DOS linker, and vice versa.

Since Pip tapes are multi-file, creation of multiple file output tape requires an appropriate tape driver in UNIX. The standard UNIX driver allows multi-file tapes to be read, but not written. A new standard tape driver will be available shortly from USG.

Piptp currently works only with mag tape, but programs dealing with DECtape and RK05 disks have been located, and will probably be incorporated.

*E. BOS Library Program*

Blip is a program to provide minimal capabilities to replace a module on an attached BOS disk, or add/replace a module via a CATLDR "tape option" tape containing just one module.

---

*UNIX Programmer's Manual, a.out formats in Section V.

For an attached BOS disk, you must know what the module number is currently. Also, the system library directory is not changed - hence the date associated with the last update of a module will *not* be properly updated.

### 3. Conversion Hints

It is *very* important that a methodical conversion procedure should be used in order to isolate and resolve any differences in assembling and linking a program under DOS and under UNIX. Your first efforts should be to try and duplicate under UNIX, as exactly as you can, the debugged programs you have under DOS. Any differences should be resolved *before* any changes are made to your programs. Otherwise, problems in the conversion may be obscured and will be much more costly to track down. The procedure for conversion involves a bit of extra work, but is quite straightforward.

First, the source *and* the corresponding DOS load module should be brought over via a PIP tape, along with the link map. The programs should be assembled and any flags checked. Known problems occur with .enabl gbl or .dsabl gbl which receive A flags. Remove the directive and supply .globl's if necessary. Similarly, .enabl reg is flagged, but is unnecessary. Psect attributes are not likely to cause trouble, but do differ. Local symbol blocks sometimes cause flags, but the reason is not yet understood.

When the assembly errors have been reconciled, look at your link map from DOS. If you use named psects or csects, add an extra module consisting only of .psect statements in alphabetical order, with the appropriate attributes. When linking, specify the module names in the same order as under DOS preceded by the dummy module, and use the -dc and -ns switches of Linkr. The resulting link maps should agree. If not, check the module order, psect order, and psect attributes. Also check that you brought over the load module which indeed corresponds to your source!

When the link map from UNIX agrees with the map from DOS, convert the DOS load module to a UNIX format with Lmc and compare it with the Linkr output. If they fail to match in the first 16 bytes, you may have forgotten to Strip the new load module, or perhaps did not use the same bottom/top load switch (remember the values are octal). If the first word of the module is not 407 octal, use a bottom load switch to force the output type to 407. Mismatches later in the module may be due to ascii constants in lower case rather than upper case. Use Octal Dump (od) to find the problem.

Finally, when everything compares, *before* you make any changes to your programs, make sure you can get all of your source code back to DOS *easily*. The multi-file format of DOS PIP tapes requires a non-standard UNIX tape driver, and you may have to modify Piptp to match yours. It is very painful to move programs one per tape in the event you need them back under DOS.

The dummy module of psects and the -dc switch can probably be discarded once the conversion is complete, unless your programs are dependent on the order of allocation. But the effort to obtain a bit for bit comparison is not much additional work, and the security of knowing the conversion was successful more than pays for it.

Paul Jensen

**P. D. Jensen**

Attachments
Appendix A
Appendix B

## *APPENDIX A*

The following list is more of a wish list than any commitment to implement the features. Priorities and schedules will be set according to the needs of users.

*Macro*

Process .dsabl gbl for IAS compatibility.

Make listing headers and footers in official PR format and provide a directive to provide pident name, etc.

Add macros to emulate common BOS (and/or DOS) system macros, connecting with run time routines.

*Linkr*

Add an archive library capability.

Provide odt debugging facilities for use under UNIX (DOS).

Add relocation information so the file may be further linked by the UNIX Ld command.

*C Compiler*

Provide for the ability to write C routines and link them in with code and data written in Macro-11.

Supply necessary run time routines needed to support C language features not related to UNIX.

Add C callable routines to interface with BOS (DOS) system calls.

*Blip*

Provide System Library Directory update capabilities.

Add ability to transmit a load module over a link to a connected test system.

Allow access to UNIX commands from the console of a test system.

*Piptp*

Option to convert the case of source files.

Should work with DOS DECtapes or disks.

*Appendix B*

User Manual Pages for Macro, Linkr, Piptp, Lmc, and Blip

**NAME**

macro - assembler for the MACRO-11 language under Unix

**SYNOPSIS**

**macro** [option1 option2 ...] file1 file2 ... filen

**DESCRIPTION**

*Macro* assembles the concatenation of the specified files, terminating when a ".end" statement is encountered. The resulting object file is named filen.obj. If a file arg does not contain a ".", file.m11 will be sought before file itself.

Options, if desired, may appear anywhere in the command, and are chosen from the following:

**-ls** Produce an assembly listing named filen.lst
**-cr** Produce a cross-reference listing named filen.lst. If the -ls option is also included, the cross reference listing will follow the assembly listing in filen.lst
**-lp** Same as -ls, but spools filen.lst for printing
**-no** No object file is produced. This is useful for syntax checking or producing the listing only.
**-xs:n** Allocates nK words of extra space for symbol table and macro storage. n must be octal and less than 20
**-ns** No symbol output in object modules. Thus ddt will not know of symbols from this assembly
**-sm:fname** System Macros will be read from file "./fname.sml" instead of the standard file /usr/lib/sysmac.sml

**FILES**

/usr/lib/sysmac.sml - system macro library (for .mcall use)
/bin/lpr - spooler
/usr/lib/macxrf - cross-reference post-processor
filen.xrf - cross-reference temporary file

**SEE ALSO**

DEC Macro-11 Manual

**BUGS**

- .enabl/.dsabl for REG and GLB are unknown and give A flags.
- switches for li nl fl en ds exist, yet do not work.
- Psect attributes are different. RO and RW become SHR and PRV, and I, D, HGH, and LOW are not supported. Unix segment attributes INS, DAT, and BSS work in their own fashion, but are not treated correctly by other programs.
- Listing header and footer formats should be changed to official BTL PR formats.
- Undiagnosed troubles with local symbols have been experienced.

**NAME**

linkr - linking loader for Macro-11 under Unix

**SYNOPSIS**

**linkr** [option1 option2 ...] file1 file2 ... filen

**DESCRIPTION**

*Linkr* link-loads the specified files in the order given, producing an executable file named filen.out. If a file arg does not contain a ".", file.obj will be sought before file itself.

Options, if desired, may appear anywhere in the command, and are chosen from the following:

**-ls** Produce a load map named filen.map
**-cr** Produce a global cross-reference listing named filen.map. If the -ls option is also included, the cross-reference listing will follow the load map in filen.map
**-lp** Same as -ls, but spools filen.map for printing
**-no** No .out file is produced. This is useful for syntax checking or producing the listing only
**-xs:n** Allocates nK words of extra space for symbol table. n must be octal and less than 20
**-ns** no symbol output with load module. Thus .out file is "stripped"
**-go** only global symbols are output in the .out file
**-t:n** top load with next free byte after program being n(octal)
**-b:n** bottom load with load point starting at n(octal)
**-dc** DOS compatible option to force contiguous blank csect

**FILES**

/bin/lpr - spooler
/usr/lib/macxrf - cross-reference post-processor
filen.xrf - cross-reference temporary file

**SEE ALSO**

DEC Linkr-11 Manual

**BUGS**

- Initialized Asects and top/bottom switch cause wipeout
- No overlay capabilities exist
- No transfer address switch exists
- Order of psect allocation is in order encountered, unlike DOS 9 which puts them in alphabetical order. The -dc switch does only part of the job
- No relocation information is produced
- No support of ddt or odt is provided
- No library capability exists

**NAME**

piptp – manipulate PIP format magtapes under UNIX

**SYNOPSIS**

**piptp –rlw**[n] [name ...]

**DESCRIPTION**

*Piptp* enables the user to read magnetic tape files which were generated under DOS in PIP format. These files are stored as UNIX files for editing under UNIX and then may be rewritten on magnetic tape in PIP format. Files which have been created under UNIX may also be written out in PIP format. The list option may be used in conjunction with the read or write options.

**–r** reads files from a PIP format magnetic tape into the current directory. The first record of each file is a 7 word header which includes the file name and extension stored in RADIX50 format. The file name and extension are extracted and the rest of the header information is discarded. If that file does not exist, it is created. If it already exists, it is overwritten. If no file names are input, the entire tape is read.

**–l** lists the file names from a PIP format tape.

**–w** writes UNIX files onto magnetic tape in PIP format. File names may be specified in any way acceptable to UNIX including the use of path names. File names are restricted to six characters and file extensions are restricted to three characters. A header record is created for each file. The header includes the file name and extension in RADIX50 format (any path names are discarded). The date of last modification is also included along with an arbitrary user identification code (UIC) of (1,1) and a protection code of 233. This code allows any accesss by the owner. The user group and all others may read or execute but not delete or write the file. This is the default code provided by the DOS system.

**–n** is a number 0-3 specifying a tape drive. Drive 0 is the default.

**FILES**

/dev/rmt[0-3] - mag tape drives 0-3
/dev/rmt[4-7] - same drives with no rewind

**BUGS**

– **requires a non-standard mag tape driver to be able to create multi-file output tapes.**

– present version reads entire tape on input. Don't specify file names.

– changes may be necessary due to the handling of EOF status from the mag tape driver.

**AUTHOR**

Pat Dowd

**NAME**

lmc - load module conversion between Unix and DOS

**SYNOPSIS**

**lmc** file1file2

**DESCRIPTION**

The first file is expected to be either a DOS or Unix load module, and is converted to a load module of the other type, becoming file2.

Only Unix type 407 files are acceptable as input or produced as output. No relocation or symbol table will be included in a Unix type output, ie the file will be 'stripped'.

**SEE ALSO**

a.out(V), piptp(VI)

**BUGS**

- It should be possible to convert type 410 files.
- No .ident or odt transfer address is put in the DOS communication directory.
- The Unix file is not in a canonical form. This may cause problems in comparisons.

**NAME**

blip - bos library replace function

**SYNOPSIS**

**blip** file.out device modnumber

**DESCRIPTION**

*Blip* creates a BOS load module (header plus 8 modblocks) from file.out (Unix type 407) and this will either replace a load module on a BOS system pack, or be placed on tape in the CATLDR tape format.

file.out will normally be the output of Linkr.

The device should be **p3** (**p4**) for a library on the BOS RP03 (RP04), **m0** (**m1**) for a tape on transport mt0 (mt1), or a Unix file containing an image of a BOS library.

The module number (modnumber) parameter should be:

a) When the device parameter is a disk (p3 or p4):
The number of the module to be replaced on the BOS system pack. This should be gotten from a current CATLDR directory listing of the BOS system pack.

b) When the device parameter is mag tape (m0 or m1):
The module number parameter is irrelevant, but must appear in the command string. Therefore use 0.

**FILES**

/dev/bos0 - the logical BOS system pack
/dev/rmt0,1 - mag tape

**BUGS**

- The command format should have order of device and file.out reversed.
- Date of last change is not reflected in the system library directory entry for the module.
- CATLDR cannot be replaced on disk without overwriting the SLD.
- The module number should not be required for tape.
- greatly improved functional capabilities are needed.