NAME

ldp — load a process

SYNOPSIS

ldp [-c] [-l syslib] file

DESCRIPTION

Ldp breaks programs up into segments, performs relocation of the text, data, and bss, satisfies external references to shared data segments (data libraries), and shared code segments (public libraries), and provides the machinery for defining symbols (of type common) in segments other than the process data and bss segment.

Input to *ldp* consists of options flags and a specification file. The *-c* option instructs *ldp* to create a new public library rather than updating the current version. The *-l* option causes the system library specified by *sublib* to be used instead of the default (/mrt/syslib). The specification file consists of lines containing a colon (:) terminated keyword and a parameter list. The elements in the parameter list are separated by blanks, tabs, commas, or if enclosed in braces ({}), by new lines. Numerical parameters are assumed to be octal unless terminated by a decimal point. In the discussion of the keywords parameters enclosed in brackets ([]) are optional, *address* means the ascii name of an external symbol, and *ps* is the octal value of the processor status word.

bss: symbol [symbol ...]

The *bss* option permits symbols of type common to be allocated at the beginning if the data segment rather than at the end. This option is useful if one is trying to insure that a set of structures begin on a 32 word boundary. The keyword *bss* is a misnomer since symbols of type bss can not be relocated using this option.

common: sr [-] symbol [symbol ...]

The common option defines a shared segment starting in segmentation register *sr* (if *sr* > 15 ldp will allocate any available registers) and assigns the symbols in the parameter list to the segment. Only symbols of type common (any uninitialized C structure) can be assigned. The size of the segment is equal to the sum of the storage required by the symbols in the parameter list. Up to eight common segments may be specified for supervisor or user processes and two for kernel processes. The common keyword must be repeated for each segment. The segment will contain all zeros when created. Subsequent invocations of the process (via a create message to the process manager) will simply increment the user count on the segment. Ldp allocates a dummy block of 512 bytes at the end of the process data for each common block. This is required in order to associate a unique name with the corresponding segment. If the minus (-) parameter is specified, the common segment will be set up as a stack segment (growing to lower addresses). The first symbol address will be computed according to the C statement:

```
addr = (sr<<13) +8192 - sizeof(symbol);
```

Successive symbols are assigned lower addresses.

copy: n

This specifies the maximum number of invocations of the process. For a process such as the disk driver *n* is usually 1. For the unix supervisor, *n* is normally greater than the number of available processes in the system.

data:	pathname [pathname ...] References to symbols in shared data segments are satisfied. The data segments are placed in the holes in the process address space after all space has been allocated for public libraries, process text, process private data, and stack. Pathnames must begin at the root file system and must be the same in the system under which ldp is running as in the system the process will run.
database: sr [s]	The data and bss sections of the public library are loaded starting in segmentation register *sr.* The *s* parameter causes this segment to be shared by multiple invocations of the library. This is useful for generating system libraries where the text, data, and bss are combined in one sharable, writable segment. This keyword is only meaningful if a public library is being created (*mode:* p).
dcom:	The *dcom* option (for data common) causes the process data segment to be shared (normally the text segment is shared and a new copy of the data segment is made). The segment always appears as the third entry in the PCB segment table for supervisor processes; for kernel processes the data segment will begin at the next 8K byte boundary.
dsect:	sr [-] symbol [symbol ...] The *dsect* keyword is short hand for dummy section. It permits symbols of type common (eg uninitialized C structures) to be assigned values corresponding to segmentation register *sr* (eg if *sr* = 3, the first symbol would have the address 060000). By default, successive symbols are assigned higher addresses. Symbol addresses may cross segmentation register boundaries and rap around. If the minus (-) parameter is given, the address assigned to the first symbol is given by the C statement: addr = (sr<<13) +8192 -sizeof(symbol); Successive symbols are allocated at lower addresses. No logical segment is created by the *dsect* option; it is simply a way of controlling the addresses of structures without knowing their sizes.
emt:	address *Address* specifies the entry point into a kernel process for servicing emt traps from supervisor processes.
entry:	address [ps] *Address* specifies the initial entry point into a supervisor process.
event:	address [ps] *Address* specifies the entry point into a supervisor process for receiving event interrupts.
fault:	address [ps] *Address* specifies the entry point to handle all traps (except emt and bpt from supervisor or kernel mode).
idchar:	c The character *c* is passed the process manager in the process file header. The process manager passes the character to the kernel process tables where it is available for identifying the process with the *ps* program. This option is only meaningful for kernel processes.
ifile:	pathname [pathname] ... [-lx] ... This option specifies the list of object files to be link-edited together to

form the process image. The normal shell syntax may be used (eg *.o abc? x[r-z]*.o). The "-lx" is short hand for /lib/libx.a and is used to specify archive format libraries. The modules are loaded and libraries searched in the order specified in the parameter list. The system library (if a kernel process so requests), any public libraries, /lib/libe.a (for supervisor processes) or /lib/libk.a (for kernel processes), /lib/libs.a, and /lib/libc.a are then searched in order.

interrupt: device entry
or
vector entry
The parameter list specifies the vector addresses and the corresponding entry point for all interrupts the process expects to handle. The specification of the vector address can be by *vector,* the octal address of the interrupt vector, or *device,* a symbolic name for one of the standard DEC peripherals:

rconsole	keyboard part of console teletype
xconsole	printer part of console teletype
pc11r	paper tape reader
pc11p	paper tape punch
kw11p	programmable clock
parity	parity memory
pl	xyplotter
lp	lp11 line printer
ls	ls11 line printer
rf	rf11 disk
rc	rc11 disk
tc	tc11 DEC tape
rk	rk11 disk
tm	tm11 magnetic tape
cr	cr11 card reader
cd	cd11 card reader
cm	cm11 card reader
rp	rp03 disk
tf	telefile rp03 equivalent disk
ta	cassette tape

A process can specify up to 32 interrupts.

mode: ksupD[l][d][i]
Specifies whether the output file is a public library, data library (D), kernel (k), supervisor (s), or user (u) process. The *l* option causes the system library symbol table to be searched before the libraries /lib/libk.a and /lib/libs.a (kernel processes only). The *d* (or *i*) option causes data and bss to be relocated to the data segmentation registers.

ofile: pathname
The process file is deposited in *pathname.*

open:
This keyword specifies that the process expects an open I/O message from the file system whenever a device handled by the process is opened and a close I/O message whenever a device is closed.

pcbbase: sr
The segmentation register allocated to the process PCB is *sr.* The default value is zero.

priority: n

For kernel processes *n* is the actual processor priority at which the processor (bits 7:5 of the ps) is set while the process is executing (3 <= n <= 7). For supervisor processes *n* is the scheduler priority of the process (0<=n <=260).

publib: pathname [pathname ...]
The public libraries specified by the parameter list are searched during the binding phase of process building. The *pathnames* must start from the root file system and be the same in the system under which ldp is running as the system under which the process will run.

share: sr access [symbol symbol ...]
The *share* option permits a process to specify that a shared segment is to be provided by the creating process, and that the shared segment is to appear in the process virtual address space starting in segmentation register *sr*. The shared segment will be given access permission *access* (2 for read only, 6 for read/write), and the symbols (of type common) will be given addresses starting at the beginning of the segment. For supervisor processes, the shared segment will appear in the PCB segment table as the third entry (after the PCB, stack, and text) if the data and stack are combined (see *stack* option) or as the fourth entry (after the data segment) if the data and stack are not combined.

stack: size sr [d]
The stack segment of *size* bytes will be allocated starting in segmentation register *sr* and occupying successive (lower numbered) segmentation registers. The *d* option causes data, bss, and common symbols to be loaded at the top (high address end) of the stack segment. If the *d* option is specified, the size of the stack will be:

data size + bss size + common size + *size.*

The stack pointer will be initialized to point to the appropriate virtual address:

sp = (*sr* <<13) +8192 - data size - bss size;

or

sp = (*sr* <<13) +8192;

The stack segment always appears as the second entry in the PCB segment table.

swap: start nblks
The *swap* keyword declares the process as a valid candidate for supporting system swapping. *Start* is the block number (relative to the start of the logical device defined by *sgen* (e)), and *nblks* is the number of blocks in the swap area. An example for an rk disk might be:

swap: 4000., 872.

Note that there are no checks on logical device between *sgen* and *ldp,* hence an incorrect specification of the logical device could result in a file system being destroyed.

time: n
The length of the process time slice (in 1/60ths second) is set to *n.*

textbase: sr
The starting segmentation register for the text sharable segment of a public library is set to *sr.* If the *database* keyword is not specified, all data and bss will be included in the text segment and thus will be read only. The *textbase* keyword is only meaningful when a public library is being built (*mode:* pD).

The specification file for the unix supervisor is:

```
          mode:          s
          pcbbase:       5
          dcom:
          sp:            1024.,7
          dsect:         7,_u
          ifile:         *.o
          ofile:         /etc/unix
          entry:         start,030000
          fault:         trap,030000
          event:         event,030040
          time:          120.
```

The specification file for the kernel process which handles the dh11 device is:

```
          mode:          kl
          priority: 5
          interrupt:{
                         320,_dhrint
                         324,dhxint
                         314,_dmint
          }
          emt:           dhemt
          event:         dhevent
          ifile:         dh.o dhdm.o dhmch.o
          ofile:         /dev/cdev1
```

FILES

/bin/ld
/bin/sh
a.out

ALSO SEE

ps(1), kprc(e)

DIAGNOSTICS

Error conditions are classified as being either fatal or non-fatal. The following warning messages are output on the occurrence of a non-fatal error

Bad option	A bad keyword in the specification file - the line is ignored.
Event entry point ??	No entry point was specified for events.
Intr. vectors ??	No interrupts were specified for a kernel process.
U: symbol	The symbol *symbol* is undefined