NAME as2 -- assembler pass 2

SYNOPSIS

DESCRIPTION _as2_ is invoked by the assembler _as_ to perform its second pass.

FILES see _as_

SEE ALSO _as_

DIAGNOSTICS see _as_

BUGS

OWNER dmr

NAME ba -- B assembler

SYNOPSIS /etc/ba name

DESCRIPTION ba is invoked by the B command in order to turn the B intermediate code into assembly language.

FILES name.i (input), name.s (output)

SEE ALSO b command, /etc/bc

DIAGNOSTICS

BUGS At the moment, the b command is defunct, and ba is invoked via a command file.

OWNER ken

NAME bc -- B compiler

SYNOPSIS /etc/bc name.b name.i

DESCRIPTION bc is the B compiler proper; it turns B source into intermediate code. It is invoked from the b command.

FILES name.b (input), name.i (intermediate output)

SEE ALSO b (command), /etc/ba

DIAGNOSTICS

BUGS The b command is defunct at the moment; bc is called from a command file.

OWNER ken

NAME bilib -- B interpreter library

SYNOPSIS

DESCRIPTION *bilib* is the library of B runtime operators. It is searched during the loading of a B—compiled program.

Standard *B* subroutines are contained in /etc/libb.a.

FILES

SEE ALSO b (command); ar, ld

DIAGNOSTICS

BUGS The following assignment binary operators are missing: b102 (=|) b103 (=&), b104 (===), b105 (=!=), b106 (=<=), b107 (=<), b110 (=>=), b111 (=>=), b112 (=>>), b113 (=<<), b120 (=/).

OWNER ken, dmr

NAME bos, maki, rom, vcboot, msys, et al

SYNOPSIS

DESCRIPTION On the RF disk, the highest 16K words are reserved for stand—alone programs. These 16K words are allocated as follows:

bos	(1K)
Warm UNIX	(6K)
Cold UNIX	(6K)
unassigned	(3K)

The UNIX read only memory (ROM) is home cut with 2 programs of 16 words each. The first (address 173700) reads bos from the RF disk into core location 54000 and transfers to 54000. The other ROM program (address 173740) reads a DECtape sitting in the end—zone on drive 0 into core location 0 and transfers to 0. This latter operation is compatible with part of DEC's standard ROM. The disassembled code for the UNIX ROM follows:

```
173700: mov     $177472,r0          12700;177472
        mov     $3,-(r0)            12740;3
        mov     $140000,-(r0)       12740;140000
        mov     $54000,-(r0)        12740;54000
        mov     $-2000,-(r0)        12740;176000
        mov     $5,-(r0)            12740;5
        tstb    (r0)                105710
        bge     .-2                 2376
        jmp     *$5400Q             137;54000

173740: mov     $177350,r0          12700;177350
        clr     -(r0)               5040
        mov     r0,-(r0)            10040
        mov     $3,-(r0)            12740;3
        tstb    (r0)                105710
        bge     .-2                 2376
        tst     *$177350            5737;177350
        bne     .                   1377
        movb    $5,(r0)             112710;5
        tstb    (r0)                105710
        bge     .-2                 2376
        clr     pc                  5007
```

The program bos (Bootstrap Operating System) examines the console switchs and executes one of several internal programs depending on the setting. If no setting is recognizable, bos loops waiting for a recognizable setting. The following settings are currently recognized:

173700

73700 Will read Warm UNIX from the RF into core location 0 and transfer to 400.

1 Will read Cold UNIX from the RF into core location 0 and transfer to 400.

2 Will read the unassigned 3K program into core location 0 and transfer to 400.

10 Will dump 12K words of memory from core location 0 onto DECtape drive 7.

0 Will load a standard UNIX binary paper tape into core location 0 and transfer to 0.

57500 Will load the standard DEC absolute and binary loaders and transfer to 57500.

Thus we come to the UNIX warm boot procedure: put 173700 into the switches, push load address and then push start. The alternate switch setting of 73700 that will load warm UNIX is used as a signal to bring up a single user system for special purposes. See /etc/init.

Cold boots can be accomplished with the Cold UNIX program, but they're not. Thus the Cold UNIX slot on the RF may have any program desired. This slot is, however, used during a cold boot. Mount the UNIX INIT DECtape on drive 0 positioned in the end—zone. Put 173740 into the switches. Push load address. Put 1 into the switches. Push start. This reads a program called vcboot from the tape into core location 0 and transfers to it. vcboot then reads 16K words from the DECtape (blocks 1—32) and copies the data to the highest 16K words of the RF. Thus this initializes the read—only part of the RF. vcboot then reads in bos and executes it. bos then reads in Cold UNIX and executes that. Cold UNIX halts for a last chance before it completely initializes the RF file system. Push continue, and Cold UNIX will initialize the RF. It then sets into execution a user program that reads the DECtape for initialization files starting from block 33. When this is done, the program executes /etc/init which should have been on the tape.

The INIT tape is made by the program maki running under UNIX. maki writes vcboot on block 0 of /dev/tap7 It then copies the RF 16K words (using /dev/rf0) onto blocks 1 thru 32. It has internally a list of files to be copied from block 33 on. This list follows:

/etc/init
/bin/chmod

/bin/chown
/bin/cp
/bin/ln
/bin/ls
/bin/mkdir
/bin/mv
/bin/rm
/bin/rmdir
/bin/sh
/bin/stat
/bin/tap

Thus this is the set of programs available after a cold boot. /etc/init and /bin/sh are mandatory. /bin/tap and /bin/mkdir are used to load up the file system. The rest of the programs are frosting. As soon as possible, an _sdate_ should be done.

The last link in this incestuous daisy chain is the program msys

msys char file

will copy the file file onto the RF read only slot specified by the characacter _char_. Char is taken from the following set:

_b_ _bos_
_u_ Warm UNIX
_1_ Cold UNIX
_2_ unassigned

Due to their rarity of use, and msys are maintained offline and must be reassembled before used.

FILES /dev/rf0, /dev/tapn

SEE ALSO /etc/init, /bin/tap, /bin/sh, /bin/mkdir, bppt format

DIAGNOSTICS

BUGS The files /bin/mount, /bin/sdate, and /bin/date should be included in the initialization list of _maki_.

OWNER ken

NAME brt1, brt2 -- B runtime routines

SYNOPSIS

DESCRIPTION The first of these routines must be loaded first in an executable B program; the second must be loaded last, after all other routines. They are not in /etc/bilib only because having them separate is the easiest way to assure the order of loading.

FILES

SEE ALSO b command, bilib

DIAGNOSTICS

BUGS

OWNER ken

NAME f1, f2; f3, f4 -- Fortran compiler

SYNOPSIS

DESCRIPTION These programs represent the four phases of a Fortran compilation:

    f1: specification statements
    f2: common and equivalence allocation
    f3: executable statements
    f4: cleanup

Each exec's the next; the first is called by the for command.

FILES f.tmpl, f.tmp2, f.tmp3

SEE ALSO for

DIAGNOSTICS

BUGS Besides the fact that there is a good deal of the Fortran language missing, there is no for command; Fortran is invoked via a command file.

OWNER ken, dmr

NAME glob -- global

SYNOPSIS

DESCRIPTION _glob_ is used to expand arguments to the shell containing "*" or "?". It is passed the argument list containing the metacharacters; _glob_ expands the list and calls the command itself.

FILES

SEE ALSO sh

DIAGNOSTICS "No match", "no command"

BUGS _glob_ will only load a command from /bin. Also if any "*" or "?" argument fails to generate matches, "No match is typed and the command is not executed.

OWNER dmr

NAME init -- process initialization

SYNOPSIS

DESCRIPTION

init is invoked inside UNIX as the last step in the boot procedure. It first carries out several housekeeping duties: it must change the modes of the tape files and the RK disk file to 17, because if the system crashed while a tap or rk command was in progress, these files would be inaccessible; it also truncates the file /tmp/utmp, which contains a list of UNIX users, again as a recovery measure in case of a crash. Directory usr is assigned via sys mount as resident on the RK disk.

init then forks several times so as to create one process for each typewriter channel on which a user may log in. Each process changes the mode of its typewriter to 15 (read/write owner, write—only non—owner; this guards against random users stealing input) and the owner to the super—user. Then the typewriter is opened for reading and writing. Since these opens are for the first files open in the process, they receive the file descriptors 0 and 1, the standard input and output file descriptors. It is likely that no one is dialled in when the read open takes place; therefore the process waits until someone calls. At this point, init types its "login: message and reads the response, which is looked up in the password file. The password file contains each user's name, password, numerical user ID, default working directory, and default shell. If the lookup is successful and the user can supply his password, the owner of the typewriter is changed to the appropriate user ID. An entry is made in /tmp/utmp for this user to maintain an up—to—date list of users. Then the user ID of the process is changed appropriately, the current directory is set, and the appropriate program to be used as the Shell is executed.

At some point the process will terminate, either because the login was successful but the user has now logged out, or because the login was unsuccessful. The parent routine of all the children of init has meanwhile been waiting for such an event. When return takes place from the sys init simply forks again, and the child process again awaits a user.

There is a fine point involved in reading the login message. UNIX is presently set up to handle automatically two types of terminals: 150 baud, full duplex terminals with the line—feed function (typically, the Model 37 Teletype terminal), and 300 baud, full duplex terminals with only the line—space function (typically the GE TermiNet terminal). The latter type identifies itself by sending a line—break (long space) signal at login time. Therefore, if a null character is received during reading of the login line, the typewriter mode is set to accommodate this terminal and the "login:" message is typed again (because it was garbled the first time).

Init, upon first entry, checks the switches for 73700. If this combination is set, will open /dev/tty as standard input and output and directly execute /bin/sh. In this manner, UNIX can be brought up with a minimum of hardware and software.

FILES /tmp/utmp, /dev/tty0 ... /dev/ttyn

SEE ALSO sh

DIAGNOSTICS "No directory", "No shell". There are also some halts if basic I/O files cannot be found in /dev.

BUGS

OWNER ken, dmr

NAME kbd -- keyboard map

SYNOPSIS cat /etc/kbd

DESCRIPTION kbd contains a map to the keyboard for model 37 Teletype terminals with the extended character set feature. If kbd is printed on such a terminal, the following will appear:

<[1234567890—_]^\ >qwertyuiop@ asdfghjkl;: zxcvbnm,./

... [rest deleted --DMR 1998]

FILES

SEE ALSO

DIAGNOSTICS

BUGS

OWNER jfo

NAME liba.a -- assembly language library

SYNOPSIS

DESCRIPTION This library is the standard location for assembly—language subroutines of general use. A section of this manual is devoted to its contents.

This library is searched when the link editor ld encounters the "—l" argument.

FILES

SEE ALSO ld; library manual

DIAGNOSTICS

BUGS

OWNER dmr, ken

NAME libb.a. -- B library

SYNOPSIS

DESCRIPTION This library contains all B—callable subroutines of general utility. Its contents are detailed in the library sectionof the B manual. At present its contents are:

```
char
getchr
putchr
exit
printf
seek
setuid
stat
time
unlink
wait
lchar
chdir
chmod
chown
close
creat
execl
execv
fork.
fstat
getuid
intr
link
makdir
open
read
write
ctime
```

FILES

SEE ALSO b

DIAGNOSTICS

BUGS

OWNER ken, dmr

NAME /etc/libf.a –– Fortran library

SYNOPSIS

DESCRIPTION This library contains all the Frotran runtime routines. Many are. missing.

FILES

SEE ALSO f1, f2, f3, f4

DIAGNOSTI CS

BUGS Will be renamed, and libf.a reserved for subroutines and functions.

OWNER ken, dmr

NAME logging in and logging out

SYNOPSIS

DESCRIPTION UNIX must be called from an appropriate terminal. The two general classes of terminals which UNIX supports are typified by the 37 Teletype on the one hand and the GE TermiNet 300 and Memorex 1240 on the other. The principal difference is the baud rate (150 vs. 300) and the treatment of the carriage return character. Most terminals operating at 150, 300, or 1200 baud using the ASCII character set either work (more or less) at the moment or can be used by special arrangement. In particular, special arrangement is necessary for terminals which do not generate lower—case ASCII characters.

It is also necessary to have a valid UNIX user ID and (if desired) password. These may be obtained, together with the telephone number, from the system administrators.

The same telephone number serves terminals operating at both the standard speeds. When a connection is established via a 150—baud terminal (e.g. TTY 37) UNIX types out "login:" ; you respond with your user name, and, if a mask is typed, with a password. If the login was successful, the @ character is typed by the Shell to indicate login is complete and commands may be issued. A message of the day may be typed if there are any announcements. Also, if there is a file called mailbox , you are notified that someone has sent you mail. (See the mail command.)

From a 300—baud terminal, the procedure is slightly different. Such terminals often have a full—duplex switch, which should be turned on (or conversely, half—duplex should be turned off). When a connection with UNIX is established, a few garbage characters are typed (these are the login: message at the wrong speed). You should depress the "break" key; this is a speed—independent signal to UNIX that a 300—baud terminal is in use. It will type login: (at the correct speed this time) and from then on the procedure is the same as described above.

Logging out is simple by comparison (in fact, sometimes too simple). Simply generate an end—of—file at Shell level using the EOT character; the "login:" message will appear again to indicate that you may log in again.

It is also possible to log out simply by hanging up the terminal; this simulates an end–of–file on the typewriter.

FILES

SEE ALSO init

DIAGNOSTICS

BUGS Hanging up on programs which never read the typewriter or which ignore end–of–files is very dangerous; in the worst cases, the programs can only be halted by restarting the system.

OWNER ken, dmr

NAME msh -- mini-shell

SYNOPSIS

DESCRIPTION *msh* is a heavily simplified version of the Shell. It reads one line from the standard input file, interprets it as a command, and calls the command.

The mini-shell supports few of the advanced features of the Shell; none of the following characters is special:

    > < $ \ ; &

However, "*" and "?" are recognized and *glob* is called. The main use of *msh* is to provide a command-executing facility for various interactive sub-systems.

FILES

SEE ALSO sh, glob

DIAGNOSTICS

BUGS

OWNER ken, dmr

NAME suftab -- suffix table

SYNOPSIS

DESCRIPTION suftab is a table of suffixes used to guide hyphenation in roff. Its first 12 words are not used (see a.out format) Its next 26 words point to the beginning of the subtables for each of the 26 initial letters of a suffix. The first entry for each suffix is a count of the number of bytes in the suffix. The second byte of each entry is a flag indicating the type of suffix. The suffix itself follows; the high bits of each letter indicate where the hyphens come. The table for each initial suffix letter ends with a zero count byte.

FILES

SEE ALSO roff

DIAGNOSTICS

BUGS

OWNER jfo, dmr, ken

NAME tabs -- tab stop set

SYNOPSIS cat /etc/tabs

DESCRIPTION When printed on a suitable terminal, this file will set tab stops at columns 8, 16, 24, 32, ... Suitable terminals include the Teletype model 37 and the GE TermiNet 300.

Since UNIX times delays assuming tabs set every 8, this has become a defacto 'standard.'

FILES

SEE ALSO

DIAGNOSTICS

BUGS

OWNER ken