NAME mount -- mount file system

SYNOPSIS sys mount; special; name / mount = 21.; not in assembler

DESCRIPTION _mount_ announces to the system that a removable file system has been mounted on special file special; from now on, references to file _name_ will refer to the root file on the newly mounted file system. _Special_ and _name_ are pointers to null—terminated strings containing the appropriate path names.

Name must exist already. If it had useful contents, they are inaccessible while the file system is mounted.

Almost always, _name_ should be a directory so that an entire file system, not just one file, may exist on the removable device.

FILES

SEE ALSO umount

DIAGNOSTICS Error bit (c—bit) set if _special_ is inaccessible or _dir_ does not exist.

BUGS At most one removable device can be mounted at a time. The use of this call should be restricted to the super—user.

OWNER ken, dmr

NAME open -- open for reading or writing

SYNOPSIS sys open; name; mode / open = 5.
(descriptor in r0)

DESCRIPTION _open_ opens the file _name_ for reading (if mode is 0) or writing (if _mode_ is non—zero), _name_ is the address of a string of ASCII characters representing a path name, terminated by a null character.

The file descriptor should be saved for subsequent calls to read (or write) and close.

In both the read and write case the file pointer is set to the beginning of the file.

If the last link to an open file is removed, the file is not destroyed until it is closed.

FILES

SEE ALSO creat, read, write, close

DIAGNOSTICS The error bit (c—bit) is set if the file does not exist, if one of the necessary directories does not exist or is unreadable, or if the file is not readable.

B UGS

OWNER ken, dmr

NAME quit -- turn off quit signal

SYNOPSIS sys quit; flag / quit = 26.

DESCRIPTION When flag is 0, this call disables quit signals from the typewriter (ASCII FS). When _flag_ is 1, quits are re—enabled, and cause execution to cease and a core image to be produced. When _flag_ is an address in the program, a quit causes control to be sent to that address.

Quits should be turned off only with due consideration.

FILES

SEE ALSO sys intr turns off interrupts

DIAGNOSTICS

BUGS

OWNER ken, dmr

NAME read -- read from file

SYNOPSIS (file descriptor in r0)

```
sys     read; buffer; nchars        / read = 3.
(nread in r0)
```

DESCRIPTION A file descriptor is a word returned from a successful _open_ call.

Buffer is the location of nchars contiguous bytes into which the input will be placed. It is not guaranteed that all nchars bytes will be read, however; for example if the file refers to a typewriter at most one line will be returned. In any event the number of characters read is returned in r0.

If r0 returns with value 0, then end—of—file has been reached.

FILES

SEE ALSO open

DIAGNOSTICS As , r0 is 0 on return when the end of the file has been reached. If the read was otherwise unsuccessful the error bit (c—bit) is set. Many conditions, all rare, can generate an error: physical I/O errors, bad buffer address, preposterous _nchars_, file descriptor not that of an input file.

BUGS

OWNER ken, dmr

NAME rele -- release processor

SYNOPSIS sys rele / rele = 0; not in assembler

DESCRIPTION This call causes the process to be swapped out immediately if another process wants to run. Its main reason for being is internal to the system, namely to implement timer—runout swaps. However, it can be used beneficially by programs which wish to loop for some reason without consuming more processor time than necessary.

FILES

SEE ALSO

DIAGNOSTICS

BUGS

OWNER ken, dmr

NAME seek -- move read/write pointer

SYNOPSIS (file descriptor in r0)

```
sys      seek; offset; ptrname        / seek = 19.
```

DESCRIPTION The file descriptor refers to a file open for reading or writing. The read (or write) pointer for the file is set as follows:

if _ptrname_ is 0, the pointer is set to _offset_.

if _ptrname_ is 1, the pointer is set to its current location plus _offset_.

if _ptrname_ is 2, the pointer is set to the size of the file plus offset.

FILES

SEE ALSO tell

DIAGNOSTICS The error bit (c—bit) is set for an undefined file descriptor.

BUGS A file can conceptually be as large as 2**20 bytes. Clearly only 2**16 bytes can be addressed by seek. The problem is most acute on the tape files and RK and RF. Something is going to be done about this.

OWNER ken, dmr

NAME setuid -- set process ID

SYNOPSIS (process ID in r0)

```
sys     setuid          / setuid = 23.
```

DESCRIPTION The user ID of the current process is set to the argument in r0. Both the effective and the real user ID are set. This call is only permitted to the super—user.

FILES

SEE ALSO getuid

DIAGNOSTICS Error bit (c—bit) is set if the current user ID is not that of the super—user.

BUGS

OWNER ken, dmr

NAME smdate -- set modified date on file

SYNOPSIS (time to AC—MQ)
sys smdate; file / smdate = 30.; not in assembler

DESCRIPTION File is the address of a null—terminated string giving the name of a file. The modified time of the file is set to the time given in the AC—MQ registers.

This call is allowed only to the super—user.

FILES

SEE ALSO

DIAGNOSTICS Error bit is set if the user is not the super— user or if the file cannot be found.

BUGS

OWNER ken, dmr

NAME stat -- get file status

SYNOPSIS sys stat; name; buf / stat = 18.

DESCRIPTION name points to a null—terminated string naming a file; buf is the address of a 34(10) byte buffer into which information is placed concerning the file. It is unnecessary to have any permissions at all with respect to the file, but all directories leading to the file must be readable.

After stat, buf has the following format:

```
buf, +1             i—number
+2, +3              flags (see below)
+4                  number of links
+5                  user ID of owner size in bytes
+6,+7               size in bytes
+8,+9               first indirect block or contents block
...
+22,+23             eighth indirect block or contents block
+24,+25,+26,+27 creation time
+28,+29, +30,+31 modification time
    +32,+33         unused
```

The flags are as follows:

```
     100000   used (always on)
     040000   directory
     020000   file has been modified (always on)
     010000   large file
     000040   set user ID
     000020   executable
     000010   read, owner
     000004   write, owner
     000002   read, non—owner
     000001   write, non—owner
```

FILES

SEE ALSO fstat

DIAGNOSTICS Error bit (c—bit) is set if the file cannot be found.

BUGS The format is going to change someday.

OWNER ken, dmr

N AME stime -- set time

SYNOPSIS (time in AC—MQ)
sys stime / stime = 25.; not in assembler

DESCRIPTION stime sets the system's idea of the time and date. Only the super— user may use this call.

FILES

SEE ALSO sys time

DIAGNOSTICS Error bit (c—bit) set if user is not the super— user.

BUGS

OWNER ken, dmr

NAME stty -- set mode of typewriter

SYNOPSIS (file descriptor in r0)
sys stty; arg / stty = 31.; not in assembler
arg: dcrsr; dcpsr; mode

DESCRIPTION stty sets mode bits for a typewriter whose file descriptor is passed in r0. First, the system delays until the typewriter is quiescent. Then, the argument dcrsr is placed into the typewriter's reader control and status register, and dcpsr is placed in the printer control and status register. The DC–11 manual must be consulted for the format of these words. For the purpose of this call, the most important role of these arguments is to adjust to the speed of the typewriter.

The mode arguments contains several bits which determine the system's treatment of the typewriter:

200 even (M37 tty) parity allowed
100 odd (non–M37 tty) allowed
040 raw mode: wake up on all characters
020 map CR into LF; echo LF or CR as CR–LF
010 don't echo (half duplex)
004 map upper case to lower case on input (M33 TTY)

Characters with the wrong parity, as determined by bits 200 and 100, are ignored.

In raw mode, every character is passed back immediately to the program. No erase or kill processing is done; the end–of–file character (EOT), the interrupt character (DELETE) and the quit character (FS) are not treated specially.

Mode 020 causes input carriage returns to be turned into new–lines; input of either CR or LF causes CR–LF both to be echoed (used for GE TermiNet 300's).

FILES

SEE ALSO gtty

DIAGNOSTICS The error bit (c–bit) is set if the file descriptor does not refer to a typewriter.

BUGS This call should be used with care. It is all too easy to turn off your typewriter.

OWNER ken, dmr

NAME tell -- get file pointer

SYNOPSIS

```
(file descriptor in r0)
sys     tell; offset; ptrname       / tell = 20.
(value returned in r0)
```

DESCRIPTION The file descriptor refers to an open file. The value returned in r0 is one of:

if _ptrname_ is 0, the value returned is _offset_;

if _ptrname_ is 1, the value is the current pointer plus _offset_

if _ptrname_ is 2, the value returned is the number of bytes in the file plus offset.

FILES

SEE ALSO seek

DIAGNOSTICS The error bit (c—bit) is set if the file descriptor is unknown.

BUGS Tell doesn’t work. Complain if you need it.

OWNER ken, dmr

NAME time -- get time of year

SYNOPSIS

```
sys     time     / time = 13.
(time AC—MQ)
```

DESCRIPTION *time* returns the time since 00:00:00, Jan. 1, 1971, measured in sixtieths of a second. The high order word is in the AC register and the low order is in the MQ.

FILES

SEE ALSO

DIAGNOSTICS

BUGS The chronological—minded user will note that 2**32 sixtieths of a second is only about 2.5 years.

OWNER ken, dmr

NAME umount -- dismount file system

SYNOPSIS sys umount; special / umount = 22.; not in assembler

DESCRIPTION umount announces to the system that special file special is no longer to contain a removable file system. The file associated with the special file reverts to its ordinary interpretation (see mount).

The user must take care that all activity on the file system has ceased.

FILES

SEE ALSO mount

DIAGNOSTICS Error bit (c—bit) set if no file system was mounted on the special file.

BUGS Use of this call should be restricted to the super—user.

OWNER ken, dmr

NAME unlink -- remove directory entry

SYNOPSIS sys unlink; name / unlink = 10.

DESCRIPTION Name points to a null—terminated string. Unlink removes the entry for the file pointed to by name from its directory. If this entry was the last link to the file, the contents of the file are freed and the file is destroyed. If, however, the file was open in any process, the actual destruction is delayed until it is closed, even though the directory entry has disappeared.

FILES

SEE ALSO link

DIAGNOSTICS The error bit (c—bit) is set to indicate that the file does not exist or that its directory cannot be written. Write permission is not required on the file itself. It is also illegal to unlink a directory (except for the super—user).

BUGS Probably write permission should be required to remove the last link to a file, but this gets in other problems (namely, one can donate an undeletable file to someone else).

If the system crashes while a file is waiting to be deleted because it is open, the space is lost.

OWNER ken, dmr

NAME wait -- wait for process to die

SYNOPSIS

```
sys     wait            / wait = 7.
(process ID in r0)
```

DESCRIPTION _wait_ causes its caller to delay until one of its child processes terminates. If any child has already died, return is immediate; if there are no children, return is immediate with the error bit set. In the case of several children several waits are needed to learn of all the deaths.

FILES

SEE ALSO fork

DIAGNOSTICS error bit (c—bit) on if no children not previously waited for.

BUGS A child which dies but is never waited for is not really gone in that it still consumes disk swap and system table space. This can make it impossible to create new processes. The bug can be noticed when several & separators are given to the shell not followed by an command without an ampersand. Ordinarily things clean themselves up when an ordinary command is typed, but it is possible to get into a situation in which no commands are accepted, so no _waits_ are done; the system is then hung.

The fix, probably, is to have a new kind of _fork_ which creates a process for which no _wait_ is necessary (or possible); also to limit the number of active or inactive descendants allowed to a process.

OWNER ken, dmr

NAME write –– write, on file

SYNOPSIS

```
(file descriptor in r0)
sys     write; buffer; nchars        / write = 4.
(number written in r0)
```

DESCRIPTION A file descriptor is a word returned from a successful _open_ or _creat_ call.

buffer is the address of _nchars_ contiguous bytes which are written on the output file. The number of characters actually written is returned in r0. It should be regarded as an error if this is not the same as requested.

For disk and tape files, writes which are multiples of 512 characters long and begin on a 512–byte boundary are more efficient than any others.

FILES

SEE ALSO sys creat, sys open

DIAGNOSTICS The error bit (c–bit) is set on an error: bad descriptor, buffer address, or count. physical I/o errors;

BUGS

OWNER ken, dmr