

NAME

dup - duplicate an open file descriptor

SYNOPSIS

```
dup (fildes)
struct { char lobyte; char hibyte; } fildes;
```

DESCRIPTION

Given a file descriptor returned from an *open*, *pipe*, or *creat* call, *dup* will allocate another file descriptor synonymous with the original. The original file descriptor must be placed in the low byte of *fildes*, i.e. *fildes.lobyte*, the high byte of i.e. *fildes.hibyte*, must be 0. The new file descriptor is returned. Normally the first available file descriptor is returned, however, the system can be forced to assign file descriptors starting at some number other than zero by setting the high byte of *fildes*, i.e. *fildes.hibyte*, to the desired starting point. The system attempts to find a non-allocated file descriptor \geq *fildes.hibyte*. If all of the file descriptors beyond the start point are used, the user is returned an error even if there are file descriptors with smaller values available. A process may have up to 20 file descriptors open at a time and the file descriptors will be assigned as numbers from zero to nineteen.

Dup is used more to reassign the value of file descriptors than to genuinely duplicate a file descriptor. Since the algorithm to allocate file descriptors returns the lowest available value, combinations of *dup* and *close* can be used to manipulate file descriptors in a general way. This is handy for manipulating standard input and/or standard output.

SEE ALSO

open(2), *close*(2), *creat*(2), *pipe*(2), *ioctl*(2)

DIAGNOSTICS

The error bit (c-bit) is set if: the given file descriptor is invalid; there are already too many open files; there are no more file descriptors beyond the value specified in the high byte of *fildes*. From C, a -1 returned value indicates an error.

ASSEMBLER

```
(dup = 41.; not in assembler)
(file descriptor in r0)
sys dup
```

1)

2)

3)

4)

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100