

NAME

fork — spawn new process

SYNOPSIS

fork ()

DESCRIPTION

Fork is the only way new processes are created. The new process's core image is a copy of that of the caller of *fork*. The only distinction is the return location and the fact that *r0* in the old (parent) process contains the process ID of the new (child) process. (In the new process, *r0* contains a 0). This process ID is used by *wait*(2).

The two returning processes share all open files that existed before the call. In particular, this is the way that standard input and output files are passed and how pipes are set up.

In the child process, the external integer *par_uid* contains the process ID of the parent process.

From C, the returned value is 0 in the child process. The value returned to the parent process is the child's process ID; however, a return of -1 indicates inability to create a new process.

The return locations in the old and new processes differ by one word. The C-bit is set in the old process if a new process could not be created.

SEE ALSO

wait(2), *exec*(2)

DIAGNOSTICS

The error bit (c-bit) is set in the old process if a new process could not be created because of lack of process space. From C, a return of -1 (not just negative) indicates an error.

ASSEMBLER

(fork = 2.)

sys fork

(new process return)

(old process return)

The return locations in the old and new processes differ by one word.