

## **FORK(II)**

## **FORK(II)**

### **NAME**

**fork** — spawn new process

### **SYNOPSIS**

(fork = 2.)

**sys fork**

(new process return)

(old process return)

**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. This process ID is used by *wait*.

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 also how pipes are set up.

From C, the child process receives a 0 return, and the parent receives a non-zero number which is the process ID of the child; a return of -1 indicates inability to create a new process.

### **SEE ALSO**

**wait (II)**, **exec (II)**

### **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.