Max Grove
MG6392
HW 19

A Brief Abstract on the fork() Command in UNIX

The fork command is a system call where a process creates a copy of itself. Fork is a process for a process to create new processes. This abstract will discuss the usage of fork, including what occurs in the operating system ("OS") after its call, and other outcomes of fork.

In the UNIX system, fork is the only system call available to create a new process. The fork will create a copy of itself. Typically, the calling process is referred to as the parent process and the resultant new process is called the child process. The parent process, currently in the *running* state, and the child process, created in the *new* state and moving to *ready*, each have independent address spaces, but the same memory images; thus, each process will not be able to view changes that occur in the other (Tanenbaum and Bos 89). These two independent processes will run concurrently as determined by the operating system, going through the *ready*, *running*, and *blocked* states (and/or *block suspended*/*ready suspended*/*exited)* as the OS deems.

The fork function takes no parameters and returns the integer process ID: 0 upon completion of the child process, the positive integer ID of the parent process upon its completion, or -1 is returned to the parent function if no child process was able to be created (IEEE and Open Group - Fork). Both parent and child will run from the fork command onwards.

After fork is called, there are several possible next steps. The parent process may use wait() or pause() to move itself into the *blocked* state until the child process exits. This avoids downsides of the two processes running concurrently (Tanenbaum and Bos 93). Upon the child's completion, the parent moves to *ready*. If the child wishes to execute a new program, it can use exec() to replace the current execution with a new one (IEEE and Open Group - Exec).

Sources

Tanenbaum, Andrew S., and Herbert Bos. *Modern Operating Systems*. 4th ed., Pearson, 2015.

The IEEE, and The Open Group. Fork, pubs.opengroup.org/onlinepubs/9699919799/functions/fork.html. Accessed 20 Nov. 2023.

The IEEE, and The Open Group. Exec, pubs.opengroup.org/onlinepubs/9699919799/functions/exec.html. Accessed 20 Nov. 2023.