Interlide: Process API

Ryan Liu

July 10, 2025

To create a process in UNIX, use a pair of system calls: fork() and exec(). A third call, wait() can be used by a process wishing to wait for a process it has created to complete.

4.1 The fork() System Call

fork() is lwk weird. We learned from Tanenbaum that fork() creates a clone of the current process. But furthermore, the process that is created doesn't start running at main(), but it comes into life as if it had called fork() itself — a true clone, a copy of the array of bytes representing the entirety of the state of the parent process.

But it still isn't an *exact* copy, like how a clone is now another individual. The value that fork() returns to the parent and child is different. While the parent receives the PID of the child, the child is returned a 0. This is useful because now we know which is parent and which is child.

4.2 Adding wait() System Call

This is actually not deterministic and can lead to race conditions, unless we use wait() of waitpid()

4.3 Finally, the exec() System Call

There are 6 variants of exec(): exec1(), execle(), execlp(), execv(), execvp(). Read the man pages