Instructions

The UNIX "Fork" command has been a critical aspect of network programming over the years. I would like you to write a (roughly) one page summary of what the function does and what activities you'd expect to see in the operating system after calling this function from a program. Please pay careful attention to the various states that a process goes through and explain which states you'd expect each process to be in and why.

Answers

fork() creates a new process by duplicating the calling process. The new process is referred to as the child process. The calling process is referred to as the parent process.

1. If fork() returns a negative value, the creation of a child process was unsuccessful.
2. If fork() returns a zero, this is the newly created child process.
3. If fork() returns a positive value, this is the the process ID of the child process to the parent.

We can determine the processes by the following:

```
pid = fork();
if (pid < 0) { // error occurred
      …
}else if (pid == 0) { // child process
      …
}else { // parent process
      …
}
```

After a new child process is created, both processes will execute the next instruction following the fork(). After calling fork() :

1. The kernel might switch to another process halfway through a process. In order to prevent this, we need to note that system calls are atomic so that the kernel will not switch to another process halfway through a system call. We should use system call in the process.
2. In Unix, there is asynchronous notification is implemented by signals. That is used for one process to interrupt another (like kill function).