Lecture Assignment

Week 9

November 23, 2022

Edward Nafornita

Student ID: 110076381

Questions selected were Q6 and Q10 from Lec09 Manual.

6. Between zombie processes and orphan processes, zombie processes are much more dangerous. Orphan processes are processes in which the parent process that initiated the child process ends earlier than the child process, thus leaving the child process without a parent process to report to. Whereas a zombie process is a process which has completed its instructions however never exited leaving it dormant (or stilling running) in the process table. The process table is a place in memory which is allocated for the sole purpose of executing processes. The process which is more dangerous would be a zombie process as it will continue to occupy space in the process table. This is problematic since a zombie process may have a loop which allocates space for some values, if this process never terminates then it will continuously allocate memory in the process table until the process table is full. This means that your computer will run out of usable memory which obviously would be bad for your computer as it will not be able to process anything once the memory overfills. An orphan process may become a zombie process if the kernel cannot take control as it's parent process, however the kernel will always attempt to gain control of an orphan so that it doesn't become a zombie process. This is not the case for a zombie process.

10. For a process to have a child the process must use the system call "fork()". This will create the parent-child process hierarchy. However, for a process to contain a grandchild the parent must initiate the system call fork() to create the child process then in the child process, it must initiate fork() also. This secondary fork() command will create a grandchild in the process hierarchy.