Thatcher Lane

CS 321

Assignment 3

02/23/18

1.20)

Fork()- In case of fork condition no memory allocation is there thus the resources are limited

Exec()- this could easily fail if the path name of the file that one wants to execute is incorrect, causing there to be more symbolic links within the translation of the path or the file

Unlink()- If the user does not have write permissions to the unlink file, or if the unlink file does not exist this could fail.

1.28)

When a system call is executed it puts the current running process on hold in order to begin the system call. The process that was previously running is saved to registers with its status being saved so that it can be resumed. In the case of system calls, after the execution of a system call, it is possible that instead of the earlier process, a new process can be started, resulting in further delays of the previous operation. With this in mind it is important for programmers to know which library wall will result in system calls so that they can prevent speed deterioration. So this means that it is important for programmers to know what will result in a system call because it could fundamentally change the logic of the program.

2.11)

A single thread process is incapable of forking while waiting for input from a keyboard. While waiting for input the process is put on hold, which prevents fork() from being used. After the input is received the process resumes and fork can be used. But because a fork cannot be created while the process is on hold it is impossible to have the same problem as we find in the multi thread program.

2.12)

2.13)

2.15)

2.28)

2.32)