Operating System Lab questions

2.14

To obtain a statistical profile of the amount of time spent by a program executing different sections of code you would implement periodic timer interrupts which are interrupts that record time spent on sections of a program.

The importance of obtaining such a statistical profile allows for the programmer to optimize the utilization of resources which would determine the amount of time spent on each section of the program.

2.19

It is desirable because of flexibility since policies aren’t static and susceptible to change. If the mechanism is separated it can support the different types of policies according to priority.

2.20

A scenario that it is unclear how to layer two system components that require tight coupling of their functionalities would be situations that involve the virtual memory subsystem and the storage subsystem because they are coupled together. This is because virtual memory system has files mapped into virtual memory space by the system while virtual memory subsystem uses storage system to provide backing for pages that aren’t located in the memory.

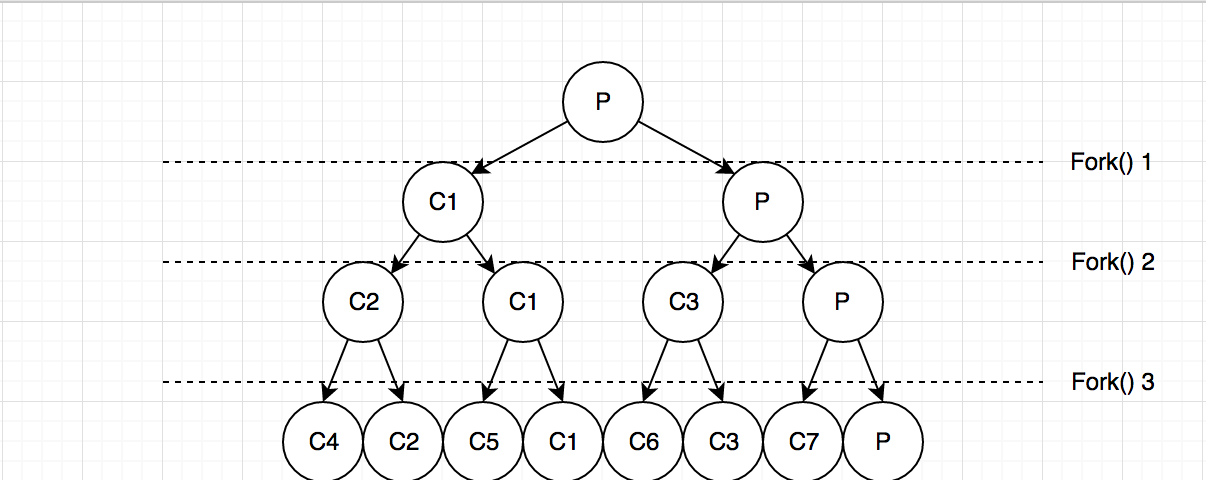
2.21

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Provides efficient communication between client program and services running on user space | Performance is decreased as overhead due to message passing increase. |
| Easy to extend operating system |  |
| Easy to port from one hardware to another due to space |  |
| More secure due to services running on user |  |

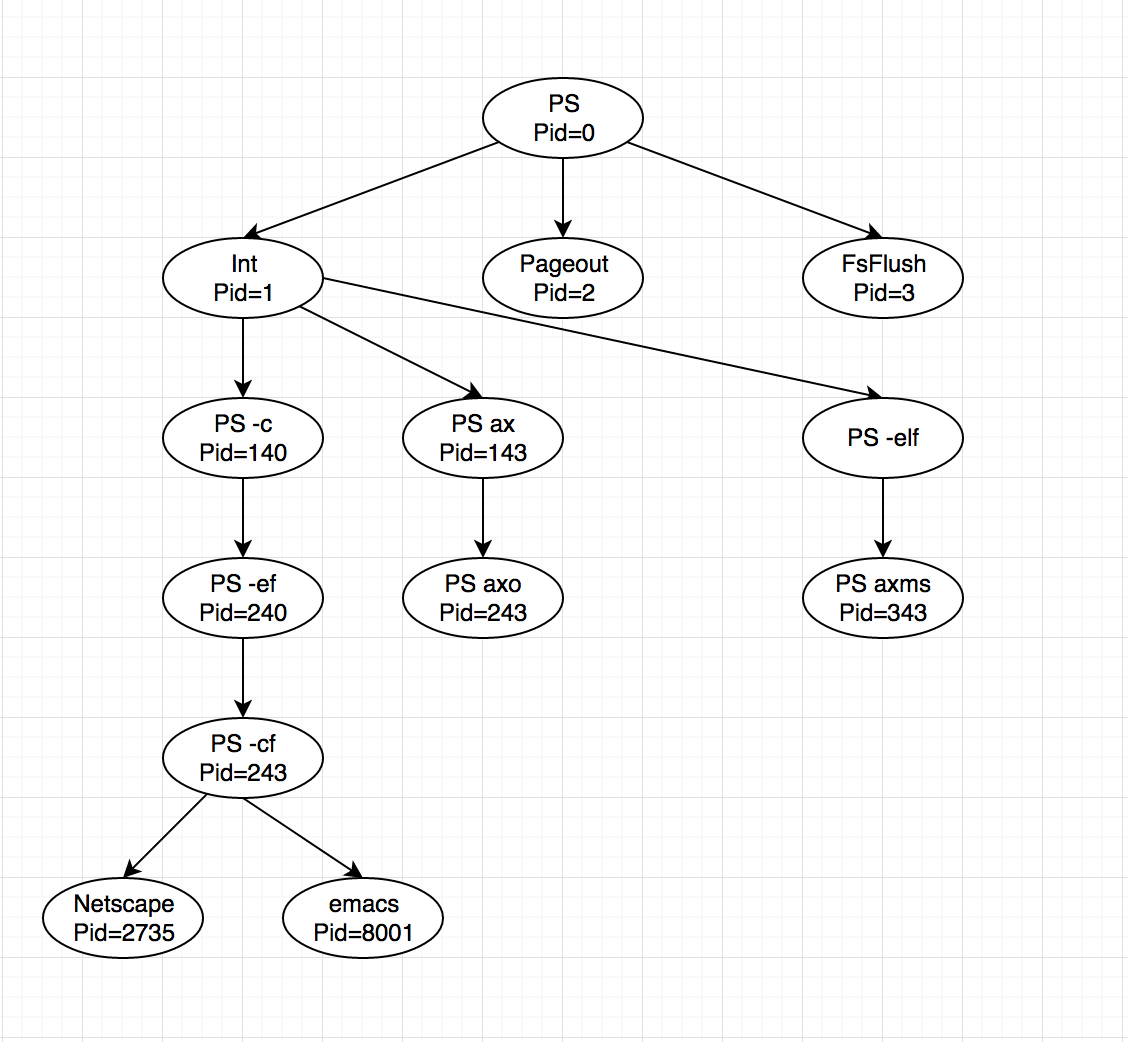
User programs and system services interact in a microkernel architecture through message passing since interaction between user programs and services can’t be done directly

3.2

Using the formula 2n-1 and substituting n for the amount of times we use fork we get 7 child Processes being created and 1 parent process. 23-1 -> 8-1=7



3.10



3.12

24-1 = 15 child processes

15 child processes and 1 parent process so 16 processes in total

3.13

The call to exec() replaces the address space of the process with the program specified as the parameter to exec(). If call to exec() succeeds, the new program is now running and control from the call to exec() never returns. In this scenario, the line printf("Line J"); would never be performed. However, if an error occurs in the call to exec(), the function returns control and therefore the line printf("Line J"); would be performed.

3.14

A = 0   
B = 2603  
C = 2603   
D = 2600

3.17

Because the child is a copy of the parent, any changes the child makes will occur in its copy of the data and won't be reflected in the parent.

X (child) output Y (parent) output

0\*0=0 Child = 0 Parent = 0

1\*-1=-1 Child = -1 Parent = 1

2\*-2=-4 Child = -4 Parent = 2

3\*-3=-9 Child = -9 Parent = 3

4\*-4=-16 Child = -16 Parent = 4