Introduction to Operating Systems

COP 4600

Name and ID:
Hunter Morera
U79718512

Homework #1

Due date: 09/03/2017 at 11:59 pm

Q1. What are the three main purposes of an operating system? (**1 point)**

Answer:

The three main purposes of the operating system are one, to make using the computer easier to use by providing a friendly UI. Two, execute users programs and provide services to applications to make solving problems easier. Third, the OS manages the computers resources such as hardware (CPU, disk drives, printer, keyboard) to use them efficiently.

Q2. What is the main advantage for an operating-system designer of using virtual-machine architecture? What is the main advantage for a user? **(1 point)**

Answer:

Virtual machines make the system easier to debug and make security problems easier to solve. Also they provide a good platform for OS research because many different operating systems can run on one physical system. For the user it provides the ability to run more than one OS without having to purchase more hardware, you can have more than one "machine" on a single physical machine.

Q3. Describe the actions taken by a kernel to context-switching between processes. **(1 point)**

Answer:

The process for a context switch by the kernel is as follows, the kernel saves the context of the old process in its PCB and loads the saved context of the new process that is scheduled to run. Essentially context switching requires performing a state of save of the current process and a state of restore of a different one.

Q4. What are the benefits and the disadvantages of each of the following? **(3 points)**

a. Synchronous and asynchronous buffering

Answer:

One advantage of synchronous buffering is that it allows for a rendezvous between the sending and receiving processes which thereby synchronizing the processes. A disadvantage of the synchronous buffering is that there is only one communication at a time and if rendezvous is not required then the processes are blocked until the communication is completed and in the meantime its unable to complete any other actions. Now with asynchronous buffering the advantage is that process flow is not restricted or blocked while waiting for a response. However, a disadvantage of asynchronous is that processes must be able to handle multiple communications which requires more functions to be implemented in the system.

b. Automatic and explicit buffering

Answer:

An advantage of automatic buffering is it provides a queue with indefinite length, this ensures the sender will never have to block while waiting to copy a message. A disadvantage of automatic buffering is that there are no specifics on how automatic buffering will be provided, one scheme may reserve sufficiently large memory where much of the memory is wasted. An advantage of explicit buffering is it specifies how large the buffer is which means it is less likely memory will be wasted with explicit buffering. However a disadvantage to explicit buffering is that the sender may be blocked while waiting for available space in the queue.

c. Fixed-sized and variable-sized messages

Answer:

The advantage of fixed size messages is that because we know the size of them we know how many will fit in a buffer of a specific size. The disadvantage is that if we have a buffer that is too small it will not hold the entire message. With a variable sized message we don't know the size so we don't know how many our buffer will hold. This means we will have to have a system in place to be able to transfer these by first putting them in shared memory which is a disadvantage, however, because of this our system is more adept at handling different size messages which makes it more flexible in it's use.

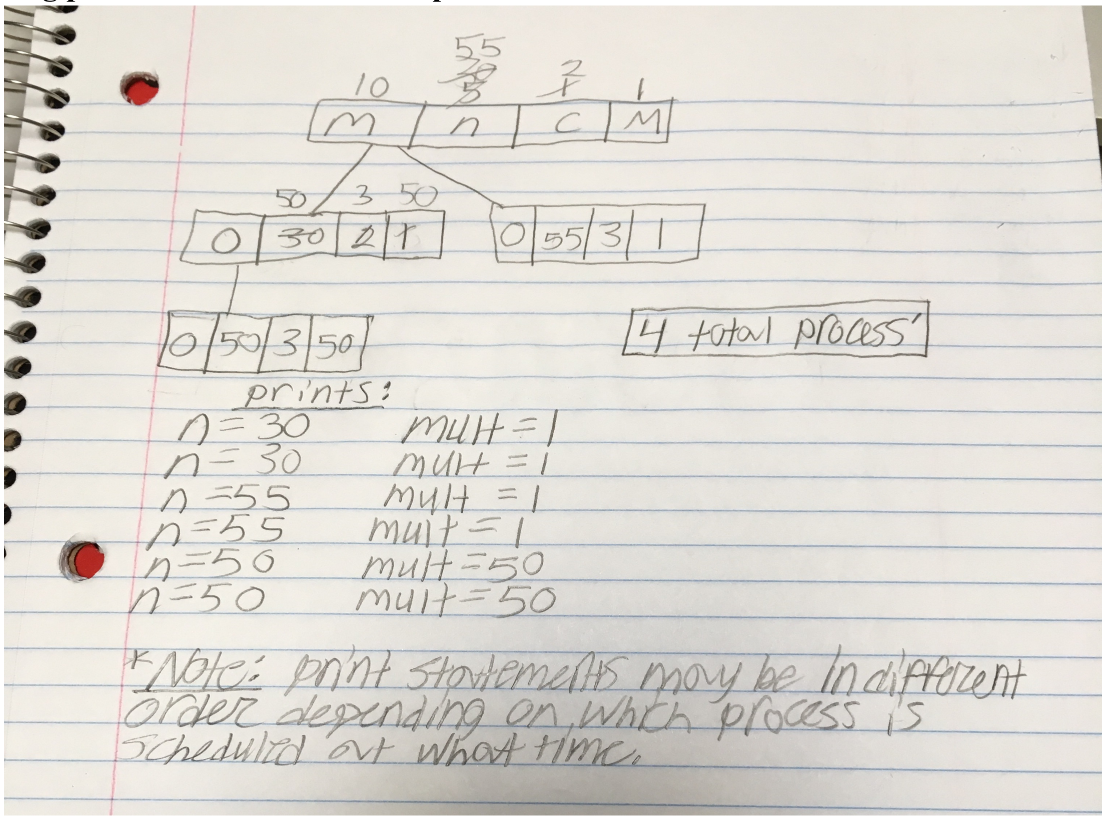
Q5. Assume that the following program contains no syntax errors. As it executes it will create one or more processes**. (4 points)**
Simulate the execution of this program and show how processes are created

```
#include<stdio.h>
main()
{
int m=10, n=5,count=1, mult=1;
while(count <3)
{
    if(m != 0)
    {
        m = fork(); n = n+25;
    }
    else
    {
        m = fork(); n = n+20; mult = mult*n;
    }
    printf(" n = %d      mult  = %d", n, mult);
    count =count + 1;
  }
}
```

What is total number of processes? Show your work.
What will this program print on the screen when it executes?

<u>As you can see from the image below there will be a total of 4 processes, this results in the following being printed to the screen in no particular order:</u>

<u>n = 30 mult = 1</u>
<u>n = 30 mult = 1</u>
<u>n = 55 mult = 1</u>
<u>n = 55 mult = 1</u>
<u>n = 50 mult = 50</u>
<u>n = 50 mult = 50</u>



prints:
n = 30 mult = 1
n = 30 mult = 1
n = 55 mult = 1
n = 55 mult = 1
n = 50 mult = 50
n = 50 mult = 50

*Note: print statements may be in different order depending on which process is scheduled at what time.