Andrew Baca
OS HW 1
September 17th, 2018

<u>**OS HW 1**</u>

CH 1
1.1 (1.1 Book)
What are the three main purposes of an operating system?
The three main purposes of an operating system include:
-Interaction and management of a computer's hardware
-Allocation and management of resources
-Control program that manages execution of users programs to prevent errors and improper use (I/O) and provides a User Interface for the user to abstract unnecessary details

1.2 (1.10 Book)
Give two reasons why caches are useful. What problems do they solve?
Caches are useful to improve performance in the communication between two object by providing a fast means of communication to them using temporary memory as sort of a buffer, opposed to the components having to reach into main memory to communicate.
Caches are also useful when they are implemented in hardware because they are used to hold instructions that are expected to be called, which saves multiple cycles of time, opposed to each instruction having to be pulled from main memory.

1.3 (1.12 Book)
In a multiprogamming and time-sharing environment, several users share the system simultaneously. This can result in various security problems. Name two examples.
One example of a security problem in a multiprogramming/time-sharing environment is that users on the same level of privilege can access and steal somebody elses information such as data, or work that they are doing. They can alter or overwrite another users work.
For example if there was a multiprogramming architecture in our labs, some student may be able to steal another's work
Another example of a security problem in a multiprogramming/time-sharing environment is that some user may have access to resources that the should not have access to such as certain commands or software that they shouldn't have access to, which results in a loss of privacy.
For example, if the whole university used a multiprogramming/time-sharing environment, the students may have access to the back end of a certain database, or some software that they should not have is certain privileges are not given to certain users.
1.4 (1.6 Book)
Which of the following instructions should be privileged: <u>a. Set value of timer.</u> b. Read the clock. c. Clear memory. d. Issue a trap instruction. <u>e. Turn off interrupts.</u> f. Modify entries in device-status table. <u>g. Switch from user to kernel mode. h. Access I/O device?</u>

g. Switch From User to Kernel Mode
h. Access I/O device
a. Set Value of timer
e. Turn Off interrupts

1.5 (1.17 Book)
Define the essential properties of the following types of operating systems:
a. Batch
In a Batch Operating System, the OS has a set of jobs that are set into units. These units get executed one at a time, in first come first serve fashion, and are stored on an output batch based on the type of job job it performs. Generally, the user indirectly interacts with the computer (think punch cards). One job starts once the prior job is finished.

b. Interactive
In an Interactive Operating System, the user is provided with direct communication towards the system. The user gives instructions the OS directly, though an I/O device such as a keyboard, and waits for the results on an output device such as a monitor. Generally, the response time is short

c. Time Sharing
In a time-sharing operating system, the CPU executes multiple jobs by rapidly switching (swapping) between them, making it so that multiple users can interact with multiple programs on the same system. This allows many users to be able to share the computer simultaneously. Time sharing involves job scheduling and CPU scheduling. A time sharing system must provide a file sharing system

d. real Time
Real time operating systems are used when there are necessary time requirements placed on an operating system for the flow of data, generally coming from a sensor. These are usually used in embedded systems to control a device on a dedicated system. Strict time constraints generally come along with the use of a real time OS.

e. Network
A network Operating System is an OS that provides features across a network with a communication scheme to allow different to exchange messages and processes across several computers. The computers act individually although they are aware they are connected to a network.

f. Parallel
A parallel operating system is an OS that uses multiple computers connected to a network to complete processes in parallel with each other. The results of the parallel systems working together will combine the final solution.

g. Distributed
A distributed Operating System consists of several different computer systems are networked so that the user can access various resources that the system contains. Jobs are assigned to the different computers and subsystems based on which system can do the job most efficiently.

h. Clustered
A clustered Operating System gathers multiple CPU’s, and joins the systems together in a loosely coupled manner. They share storage, and are linked by a network. They are meant have high availability, meaning if one core fails, the service will still continue. They can be structured symmetrically or asymmetrically.

I. Handheld
a handheld operating system is an Operating system on a small mobile device, usually a cell phone. They used to have to give up function for size, however, nowadays they are very rich in features.

Apple iOS and Google Android are the most common handheld OS. Combines traditional OS features with new features meant for handheld use.

1.6
Provide the Unix/Linux commands for the following. If multiple options exists, please select one. For choice d, state why you chose it: a. View memory usage b. View a process status c. Print (or ”list”) the author of each file d. one extra command of your choice that you don’t already know.

a. free -m shows the total memory, the used memory, and the free memory, shared, buffered/cached memory, and available mem
b. ps -ef shows all the current processes running on the computer.
c. ls -al will show you the author of all the files in that directory
d. I chose the history command in linux, and what this will do is print out the history of the recent commands used in the shell. I chose this because it is useful to me in the sense that I will not have to look up a bunch of commands on google that I have previously used and forgotten.
I also chose time as one of my Linux commands. If you type in time before any command in linux (such as time ./a.out), which will give you the user time, the system time, and the total time of the process being ran. This is useful for checking efficiency of a certain process.

CH 2

2.1 (2.1 Book)
What is The Purpose Of System Calls?
System calls are used as a means for the user to communicate with the operating system. These are generally low level commands that perform one action at a time. For instance, a small c program to read the first line from a file would involve 2 System Calls to open and close the file, as well as I/O system calls to possibly get the file name from the user, as well as the print statement.

2.2 (2.8 Book)
What is the main advantage of the layered approach to a system design? What are the disadvantages of a layered approach?
The main advantage to the Layered approach is that they are more simple to construct and debug, and they accomplish making the Operating System more modular.
One of the main disadvantages that a layered approach would have is defining what each layer contains and how it is layered, because one layer can only use its lower level layers. They are also less efficient than other types of structures due to multiple layers having to be traversed for certain actions.

2.3 (2.11 Book)
How could a system be designed to allow a choice of operating systems from which to boot? What would the bootstrap program need to do?
Usually OS is stored on a disk, and a bootstrap is located on filmware, and the bootstrap locates the kernel and loads it into main memory. We can store multiple OS’s on different segments of memory in the disk, and modify the bootstrap program to offer an interface asking the user which OS to use. From there, the bootstrap will locate the segment of memory that the OS is stored at on the disk, and load that code segment6 onto the boot block to be executed.

2.4 (2.2 Book)
What are the five major activities of an operating system with regard to process management?
Scheduling Processes and Threads on the CPU
Creating/Deleting both user and system processes
Suspending/resuming processes
Providing Mechanisms for process synchronization
Providing Mechanisms for process communication

2.5 (2.14 Book)
What is the main advantage for an operating system designer of using a virtual-machine architecture? What is the main advantage for the user?
The main advantage for a System Designer to use a virtual machine is that they can make their design usable on multiple platforms on one piece of hardware with the use of a virtual machine.
The main advantage for a user to use a virtual machine architecture is to use applications that are only available to certain operating systems on one host.

2.6
What is an advantage and disadvantage of a modular vs monolithic kernel? Provide an example where a monolithic kernel may be more advantageous than a micro or modular kernel.

	monolithic	**Modular**
Advantage	The main advantage of a monolithic kernel over a modular kernel would be that they are easier to expand upon than a modular kernel because it additions mean you do not have to modify a monolithic kernel as much. Another is reliability. If one process fails, the rest of the OS generally remains untouched	The big advantage of a modular system is that they are very broad, making general use and problem solving in multiple fields easy, for instance, with windows, linux, and mac products,
Disadvantage	One disadvantage of a monolithic system would be that they are usually task specific, o they are not good for general problem solving in multiple fields.	One disadvantage of a modular system opposed to a monolithic system would be that they are not easy to add into/modify.

You may want a monolithic system opposed to a modular system if you are doing something very task specific where you are not going to want to vary any functions/features such as the software for a printer

CH3

3.1 (Book 3.1)
Using the program shown in Figure 3.30, explain what the output will be at Line A.

The output for line a should be:

“Parent: value = 5”

The reason being is that if the pid exist and is greater than 0 (that is the pid of the child process exists), we wait till the child process is done and ended, and the original value is printed

3.2 (Book 3.5)

When a process creates a new process using the fork() operation, which of the following states is shared between the parent process and the child process?
a. Stack b. Heap c. Shared Memory Segments

Shared Memory segments are shared between the child and the parent operation, although the parent chooses which attributes the child gets and uses. They both have their own unique stack and heap

3.3 (Book 3.8)

Describe the differences among short-term, medium-term, and long-term scheduling.

The long term scheduler pulls processes from a pool stored on a mass storage device and loads them into memory for execution, while the short term scheduler chooses a process that is ready for execution, and allocated the CPU to one of them. A key difference between a long term scheduler and a short term scheduler is that the long term scheduler executes much less frequently than the short term scheduler. This is due to the careful selections that the long term scheduler must make, picking a well balanced mix between I/O bound and CPU bound processes. If all processes are CPU bound, then the short term scheduler will rarely have anything to do. Medium term schedulers are different in the sense that they can remove processes from memory, and later be pulled back into memory to resume execution, which is different because the processes can be interrupted and started back up

3.4 (Book 3.15)

Give an example of a situation in which ordinary pipes are more suitable than named pipes and an example of a situation in which named pipes are more suitable than ordinary pipes.

Ordinary pipes are used for one way communication, while named pipes are used for two processes to communicate back and forth with each other. For example, an ordinary pipe would be more ideal for a simple program to read the contents from a text file, and count the frequency of each letter, while a named pipe would be ideal for a live interactive message board during a webinar

3.5 (Book 3.18)

What are the benefits and the disadvantages of each of the following? Consider both the system level and the programmer level. a. Synchronous and asynchronous communication b. Automatic and explicit buffering c. Send by copy and send by reference d. Fixed-sized and variable-sized messages

a. Synchronous communication is good to use when every message that is passed between processes is important, And asynchronous communication is good when every message sent from process to process does not need to be known, for instance constant temperature output from a thermometer. The drawback on synchronous communication happens when all the messages being passed aren’t necessary to stick to the receiving process. The drawback an asynchronous communication happens when all the messages being passed to the receiving process are necessary for proper function.

b. Automatic Buffering occurs when the queue’s length is potentially infinite, meaning that there is not a fixed amount of messages that can be stored in it. This can potentially be inefficient, because depending on how it is implemented, it can be a major waste of memory. Bounded buffering occurs when there is a finite length of buffer memory, so there can still be messages blocked by the sender.

c. Send by copy is advantageous when the sending process does not want the receiving process to alter the message, so it sends a direct copy of the contents of the memory address, however, time and memory space are more affected. Send by Reference is fast and cheap on memory, however, the receiver can possibly alter data.
d. Fixed Sized memory is easier to implement at the hardware level. However, it is harder to implement in the programming level, and variable size is easier to implement at the programming level, but harder to use at the memory level. Fixed sized memory also can be a waste of memort, if you are always going below the fixed size to store data.

3.6
We discussed in class about sandboxing. Define sandboxing and then describe how it may be used to improve resiliency in a web browser. Additionally, define strong and weak resiliency.
Sandboxing is a term used to describe an isolation of a process from other aspects of the computer, so not of the process can escape or access information from outside of the sandbox. This can be used to improve resiliency in a web browser in the sense that if you open a web browser up in a sandbox, nobody on the same network will be able to access any information you wouldn't want them to outside the sandbox, such as files and certain directories. A system with strong resilience means that there is is no way to escape it and access critical information, while a weak resilient system is more prone to having their information accessed by outside sources.
3.7
What are I/O bound and CPU bound processes? B.Describe these two types of processes with examples of how they would be handled by the OS through the process state diagram. C. Also, comment on the size of the desirable time-slice for an OS handling only I/O bound and only CPU-bound processes respectively.
I/O bound processes are processes that spend more time doing I/O interactions rather than computations, while CPU bound processes rarely request I/O interactions and spend most of the execution time doing computations.
b. I/O bound processes will not spend very much time in the ready queue, however, it will be in the waiting queue a lot waiting for input from the device. CPU bound processes will rarely spend time in the waiting queue, and they will go from new → ready → running → terminated, because it is never waiting for messages from an I/O device-status
c. I/O bound processes can generally have a smaller time slice due to the waiting state for I/O interaction, and the use of long term scheduler opposed to short term scheduler. CPU bound processes create an imbalance in the sense that they always use the short term scheduler and dont have to wait for I/O interaction so the time slice will generally be in a whole chink opposed to a bunch of small time slices getting interrupted by I/O interaction.