# CST 352 – Final

Wilsonville, Spring 2018

This is a take home final. You must turn in your completed final on **Blackboard** by mid-night on **Thursday 6/14/2018**. You may use any and all resources available to you to complete the final. You will be expected to turn in your own work. Do not copy/paste from a source (for example from the internet), instead read the source, understand the material and then put your answer into your own words.

## Section 1 – Terminology and Concepts (60 points)

Answer each question in this section with enough detail that the instructor understands that *you* understand the term or concept.

1. (10 points) Contrast the following terms in your own words:
   1. Process vs Thread

Both processes and threads are abstractions of physical space on some kind of memory (CPU, Main Memory). A process is the space that code can function in. A program that declares an array and does something to it is an example. A thread, on the other hand is needed to actually execute this process. The act of actually putting a variable in a process in the heap is an example of a task for the thread. The process is like a record, while the thread is like the head of the record player actually reading each note.

* 1. Principle vs Subject

Both principles and subjects access resources on a computer. Operating systems try to separate these accessors in order to better secure themselves. Subjects are the things that actually access the resources. An example here is a program requiring memory, or access to a file before running. Principles are the people, or groups of people, who run the programs. Principles make use of the subjects, so it makes sense to restrict the user first, instead of trying to restrict the subjects themselves.

* 1. Device Controller vs Device Driver

Both device controllers and device drivers sit in between devices and either the user or data on a computer. The main difference is where they sit. Running in the operating system, device drivers deal with what the system needs to do with the information from the device. Device controllers sit between the device itself and the bus. This piece of hardware is responsible for getting the information over the bus to where it can be accessed by the device driver.

* 1. Internal vs External Fragmentation

Fragmentation occurs when memory segments are allocated by different programs in such a way that leaves gaps in memory. This is the same for both internal and external fragmentation. External fragmentation occurs when leftover chunks of memory are too small to be used by another process. Internal fragmentation occurs when the end of a page doesn’t get fully used. Internal refers to the fragmentation relative to the process itself. External refers to a new process trying to fit in memory.

* 1. CHS vs LBA Addressing

Both these techniques refer to how we read data off a disk. CHS (Cylinder/Head/Sector) addressing is where the disk is divided into cylinder, head, and sector sections. Data can be addressed by using those three locations. LBA (Logical Block Addressing) is just where every sector is looked at like a gigantic array.

1. (5 points) What is virtual memory and why do operating systems include it?

Virtual memory is address space given to a program. Operating systems map this memory to physical storage. This gives the operating system a chance to catch the program if it reaches out of the allocated space. A page fault is generated by the CPU if a program tries to reach out of its bounds. This could definitely be seen as a safety precaution as well as a crash precaution.

1. (5 points) What is a deadlock? Describe a scenario where one could happen. What is an operating system technique for allowing programmers to avoid deadlocks?

A deadlock is when two threads are waiting on each other. This results in neither thread doing anything and an infinite loop of waiting. An example would be if Thread A is waiting for mutex 1 to become available before releasing mutex 2. Meanwhile Thread B is waiting mutex 2 to release mutex 1. Since neither mutex will ever become available, both threads sit idle. A quick fix for a situation like this is to give threads timeouts when waiting for another mutex. General careful design is another technique.

1. (10 points) Describe the producer/consumer pattern. Use at least the following terms in your description: ~~thread~~, ~~synchronization, mutex~~, ~~event~~, ~~queue~~. Include a diagram to illustrate your explanation.

The producer/consumer pattern is used for threads to communicate with each other without possible deadlocks. Synchronization of two different kinds of threads results in efficient operations happening in tandem. Producers are only allowed insert data or requests into a queue. Consumers have access to this same queue but can only remove data or process requests. Events must be added to the queue to make sure a consumer doesn’t take something that’s not there or a producer place something into a full queue. A mutex is used to make sure that nothing is being done to the queue while its attributes are changing. An example of this process is below.

Queue

Event Capacity;

Event Items;

The queue will lock the mutex whenever updating the Capacity or Items Events. These are updated as things are put into the queue or removed.

1. (10 points) Describe the 3 layers of a modern file system. What is the role of each layer?

Physical File System – This is the system that is responsible for writing data to the physical drive. This is system that takes care of writing blocks of data to cache, which then is translated to main memory, and some non-volatile disk. This is the layer that actually saves things to a place that will exist after power off.

Logical File System - This is the place where applications interact with the file system. An example here would be when I save this document in Microsoft Word. This system abstracts away the physical storage, and makes a common system for applications to use. I could save the same file from Notepad++ and things would work the same (except for Microsoft’s pesky .docx format).

Virtual File System – This system is where different physical file systems come together into one common ground. Mount points of different disks come into play here when operating systems have to read from a physical disk. An example here is how Windows mounts drives with a letter, such as the C: drive.

1. (5 points) Contrast the First-Fit and Best-Fit algorithms. What is the significant difference? What are the pros/cons of each?

The main difference in First-Fit and Best-Fit algorithms is where the memory is allocated. In First-Fit, the first chunk of memory that will hold the requested amount is used. In Best-Fit, the smallest chunk memory is used. First Fit has been shown to be the best approach in simulations. One advantage of First-Fit is the speed in which memory can be allocated and accessed. Best-Fit has the advantage of leaving smaller unallocated chunks of memory behind. The main disadvantage of First-Fit is leaving bigger unallocated chunks, while the disadvantage of Best-Fit is it being slower.

1. (5 points) What are the potential consequences to data storage when a computer crashes. Describe an operating system technique for mitigating these risks.

When a computer crashes, the main two consequences are data loss and data corruption. Data loss occurs when data is being written to a disk as is interrupted before being committed to non-volatile memory. Data corruption occurs when only some of the data gets committed to non-volatile memory and an application reads it incorrectly later. ACID is a technique used to mitigate this risk.

“**A**ll or nothing” – Make sure all data is written, or write none at all

“**C**onsistent” – Leave the system in a consistent state

“**I**solated” – Transactions shouldn’t affect other ones until committed

“**D**urable” – Once committed, data is final

1. (5 points) What is a security context? When and how is it established? How is it used by the operating system?

Security context is implemented by the security system. This system is responsible for what users (and user groups) have access to. When a person signs in as “guest”, for example, they won’t have access to the same files as a person with full administrator rights. This difference in access is the security context, which is created upon login. The operating system can use this context to run applications differently, according to who is logged in. Ultimately, the operating system uses security context to ensure its own wellbeing, as well as give the administrator customization.

1. (5 points) Describe how session management combines various components of the operating system together to provide a secure and usable experience.

After the terminal server asks the session manager for a session the security system can be used to make sure the correct security context is used in the session. This ensures that the user’s session won’t read or write to places that user shouldn’t be. The file system also comes into play here as it is used to create the user’s directories in the session. A shell can also be used in the session, which would add much more usability. The terminal that the user sees can also be in the session, which again, adds to the usability. In short, session management encapsulates a lot of essential elements of an operating system.

## Section 2 – Problem Solving (40 points)

For each of the questions in this section, read the challenge and describe the operating system techniques that are used to solve the challenge. Imagine that the visionary OS designer had anticipated the challenges and built the OS to solve them before the computer was set up. Be specific in your answer and use examples whenever possible.

1. (10 points) A person wants to run multiple applications at the same time. Each application needs to be able to share a single processor. Bugs in one application should not impact the other applications.

From the Operating System’s standpoint, there are a number of things that can be done to allow the user to run multiple applications at once. An application crashing needs to be isolated as well.

1) Memory allocation – Each application should be run with a process. This isolates the “space” an application can run in by only allocating a specific block of memory to each one.

2) Security – The operating system will have to make sure that the processes running don’t reach out of their memory space. If they do, the OS should have higher precedence and be able to stop the process in question. This would solve the issue with one application’s crash not affecting the other applications. This is also where the logical file system can have checks to make sure no application writes or reads data from places it shouldn’t be. A check at the logical file system can stop an application before data gets written to disk, or even the virtual file system.

3) Threading – Because the computer is only a single core processor the OS will have to lock and unlock a mutex for the thread that will be running each process. Each process will gain access to the one thread as the OS sees fit. This should make the multiple processes look as if they’re running in tandem. If a single processor can be split into multiple threads, each process could get its own (including the OS itself). This takes careful design to successfully synchronize processes. This is also where the OS needs to make sure that a processes isn’t waiting or editing any resource that another process might need.

4) Context switching – A run queue can be used to stack thread objects in a row to make sure each is run successfully.

1. (10 points) A system administrator needs to add more storage capacity to a computer, without downtime. Since the computer was originally purchased, new types of hard disks have come out with much greater capacity per disk (more and larger sectors).

In short, the administrator needs to plug in (or attach via network) a new hard disk. This requires a lot from an operating system’s point of view.

1) Device Controller – At the lowest level, the new hard disk itself needs to be able to communicate its information to the computer. Whether it be over a bus or a network, this part of the device itself will transmit information to the machine in question.

2) Bus – For simplicity, this is the part that carries the data back and forth from the hard disk to the computer.

3) Device Driver – The manufacturer of the hard disk will need to provide this application. This application will launch as a process and enable the operating system to send/receive information from the drive.

4) Physical file system – This part of the OS will interact with reading/writing the data from the new hard disk.

5) Virtual file system – This part of the OS is the step in between the physical file system and the logical file system (explained in question 5)

6) Logical file system – This is the part of the OS that actually connects with the application the user needs. This is where the application will have access to the newly added storage.

Device Driver

New hard disk

Device controller

1. (10 points) A computer is accessed by many people at a time and over time. The computer is used to stored and edit confidential documents in a number of formats. Each person expects to share their documents only with other people of their choice.

Operating System

Application displaying new data storage

Bus

The first thing the operating system would have to do here is implement sessions. This is done by making users log in and allowing access to files based on the login information. This involves a couple of OS elements.

1) Session manager – This part of the OS is responsible for using a security system and file system to generate a session for the user.

2) Session – This is the context that the user sees. The file system provides a directory path, the security system creates a user, and the session manager provides a terminal. These three elements, along with a shell make up a user’s session. The session is also used as a whole to prevent simultaneous users from overwriting each other.

3) File System – This is used to abstract away the interaction with reading and writing data. Usually divided into three layers.

4) Security system – This part is used to verify credentials entered by the user and determine which files each user has access to. This is the part that would be responsible for keeping user 1’s documents away from user 2.

5) File metadata – This is where users would be able to enable sharing with specific other users. For example, User 1 can sign on and set Document1.txt’s metadata to include User 2, 3, and 4. When User 2 signs on, they’ll be able to see Document1.txt. User 5 won’t be able to. This can be implemented at the file system level, as the filesystem is responsible for creating the directories (and files) that a user sees in their session.

6) User groups – This can also be used to share documents between a range of users. The security system would be responsible for making sure “Group X” is only made up of particular users. File metadata would be used to make sure that the entire group has access.

7) Shared directories – If needed, the file system can implement some shared directories that every user has access to. An example here would be every user needed access to a directory containing a common file. As the file system creates directories for the user in a session, it would just make this directory fully accessible.

1. (10 points) A person has purchased a new laptop after their previous one suffered catastrophic failure at a local coffee shop. They bring their new equipment home and are greatly relieved to discover that their wireless mouse, keyboard, 45” screen and printer work seamlessly with their new laptop. Within 10 minutes they are leaving a positive review on the laptop vendor’s website.

This problem is best solved with the use of device drivers and device controllers.

1) Device controllers – Device controllers are in charge of getting the data to and from the actual device itself. In the case of the mouse, the device driver that sends/receives data is still intact on the mouse itself.

2) Device drivers – Device drivers are in charge of exchanging data over the communication means between the external equipment and the computer. In this case, the drivers may have been lost on the old machine, but a quick download can replace them (if they’re not included in the OS already). An example here is the device driver for the keyboard recognizing which button was pressed by the user.

3) Operating system abstraction – The operating system as a whole should show the user the same I/O results, regardless of what keyboard, mouse, or monitor is being used. While a lot of things are going in an operating system, as long as the device driver is intact things should work well.

Monitor

(with device controller)

Applications

Device Driver

Device Driver

Device Driver

Device Driver

External Equipment

(each with device controllers)

Everything working fine before coffee spill.

External Equipment

(each with device controllers)

Monitor

(with device controller)

After coffee spill (no computer)

Monitor

(with device controller)

Applications

Device Driver

Device Driver

Device Driver

Device Driver

External Equipment

(each with device controllers)

After new computer, with same device drivers

The combination of device drivers and device controllers abstract away the need for the equipment to “care” about the OS they’re sending and receiving data to/from.