CS 459 Operating Systems

Lab 1

1. **The java class loader will cause a software failure when the name of a class to run is spelled with a different case. For example, suppose you want to dynamically load and run a class saved as Tower.class. If you spell it as tower.class, the batch sequencer will bomb. What did you do to avoid this error?**

We used toUpperCase and toLowercase on the arguments strings to avoid case sensitive spelling errors.

1. **Most operating systems use doubly linked lists instead of singly linked lists. Why do you suppose that this is the case?**

You can start in the middle of the list and work either forward or backwards.

1. **Define a primary characteristic of each of the following categories of operating systems.**
   1. **Batch**

Small resident kernel sequences jobs for one or more batch streams using a job control interface.

* 1. **Time Share**

Multi-user responsiveness. User response time should be < 1 second.

* 1. **Real Time**

Responds to multiple users at workstations. Scheduling is based on real time clock “slices”

* 1. **Distributed**

Geographically separate.

* 1. **Handheld**

Slow, limited in processor speed, available power and user interfaces.

* 1. **Embedded**

Special purpose buttons and displays.

1. **What is the difference between a Java Exception and a Java Error?**

When an error accrues, you can’t recover from an error. However, you can sometimes recover when using throwable. Essentially, errors are something you don’t want to have happen. When an exception accrues, you can recover from it. Exceptions aren’t a bad thing and sometimes you want to happen.

1. **Which activities is related to process management did we implement in this lab?**

We used the priority class that allowed us to choose which program would need the most or the least amount of process resources. We could also use the Remove command to terminate a process.

1. **Why are command interpreters separate from the kernel? How could we have implemented our “batch system” with the command interpreter separate?**

Command interpreter communicates with the command line interface, the kernel communicates with the OS and hardware.

1. **In our “batch system”, what part of it would be considered the kernel?**

It should be our new OS console we created. This is where our kernel will accept new commands we create and communicate those commands with the OS

1. **How did we use polymorphism in this project?**

We used Java reflections that when we were using CLI, our commands could communicate with the OS and the programs.