John Escobal April 17, 2016

Austen Herrick

**Lab Assignment Number 1 – cs459/559**

**Goals**:

1. Get a basic understanding of operating systems by implementing a simple batch sequencer.
2. Become familiar with the Java dynamic class loader.
3. Program a doubly linked list of processes.
4. Implement a simple command driven interface.

**Part 1** Pseudo-code your command interpreter program; a few lines for each command will suffice.

**Part 2** Answer the following questions

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?

To avoid this error we used Character.toUpperCase to always capitalize the first character of the string input.

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

Doubly linked list offers easy implementation of many operations, whereas singly linked list requires more info for the same operation. And with Operating Systems requiring less information is optimal.

3. Define a primary characteristic of each of the following categories of operating systems.

a. Batch An operating system that automatically keeps executing the jobs in a batch

b. Time Share An Operating System that enables many people, located at various terminals, to use a particular computer system at the same time.

c. Real Time The level of its consistency concerning the amount of time it takes to accept and complete and application’s task.

d. Distributed An operating system that uses physically separated computational nodes that hold a specific software subset of the complete OS.

e. Handheld A specially designed operating system for mobile/handheld devices that can function in touchscreen environments.

f. Embedded An operating system that is compact, efficient, and reliable operating systems.

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

According to java concept of the day “Recovering from **Error** is not possible. The only solution to errors is to terminate the execution. Where as you can recover from **Exception** by using either try-catch blocks or throwing exception back to caller.”

5. Which activities related to process management did we implement in this lab?

In this lab we added processes to a list, removed processes from a list, also changed the priority of processes in a list.

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

The command interpreters are separate from the kernel so that the OS doesn’t have to know anything about the commands.

7. In our “batch system”, what part of it would be considered the kernel? In our batch system the kernel is the class with the main method that takes user input and interprets it.

8. How did we use polymorphism in this project?

We used polymorphism through Java reflection in this project.