COSC3407N21W - Operating Systems

Assignment #1

Due Friday January 30th,2021 (11:55pm)

Type your answers and submit them in a Word or PDF document to the CMS.

Marks for each question are indicated.

Answers should be in your own words – not cut and pasted (or copied) from the textbook. Material copied verbatim from the textbook will be penalized. TYPE ALL YOUR ANSWERS.

Q1: **The issue of resource utilization shows up in different forms in different types of operating systems. List what resources must be managed carefully in the following settings [6 marks]:**

a) **Mainframe or minicomputer systems**
b) **Workstations connected to servers**
c) **Mobile computers**

Q2: **Describe the differences between symmetric and asymmetric multiprocessing. [2 marks]**

Q3: **What are three advantages and one disadvantage of multiprocessor systems? [6 marks]**

Q4: **What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose? [6 marks]**

Q5: **Describe the mechanism for enforcing memory protection in order to prevent a program from modifying the memory associate with other programs. [3 marks]**

Q6: **Identify several advantages and several disadvantages of open-source operating systems. Include the types of people who would find each aspect to be an advantage or a disadvantage. [8 marks]**

Q7: **What is the separation of mechanism and policy and why is it desirable? [4 marks]**

Q8: **How are iOS and Android operating systems similar? How are they different? [4 marks]**

Q9: Explain why Java programs running on Android systems do not use the standard Java API and virtual machine. [2 marks]

Q10: Define short-term, medium-term, and long-term scheduling and explain the differences between each. [6 marks]

Q11: Describe what actions are taken by a kernel when if performs a context-switch between processes. [2 marks]

Q12: Explain the circumstances under which the line of code marked printf("LINE J") in the code below will be reached. [1 mark]

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
int main()
{
pid t pid;
        /* fork a child process */
        pid = fork();

        if (pid < 0) { /* error occurred */
                fprintf(stderr, "Fork Failed");
                return 1;
        }
        else if (pid == 0) { /* child process */
                execlp("/bin/ls","ls",NULL);
                printf("LINE J");
        }
        else { /* parent process */
                /* parent will wait for the child to complete */
                wait(NULL);
                printf("Child Complete");
        }
        return 0;
}
```