Operating Systems I

CS 474

Homework #1

Due date: 9/30/2020 at 11:59 pm
Instructions:
1) To complete assignment one, you need to read Chapters 1, 2, 3, and 4 of the textbook.
2) HW must be submitted on typed pdf or word document.
All questions need to be worked on and answered individually. No collaboration of any form is permitted and will amount to plagiarism if performed.

Questions from Chapter 1

1.1 What are the three main purposes of an operating system? (6 pts)

1.15 Describe the differences between symmetric and asymmetric multiprocessing. What are three advantages and one disadvantage of multiprocessor systems? (8 pts)

1.19 What is the purpose of interrupts? What are the differences between a trap and an interrupt? Can traps be generated intentionally by a user program? If so, for what purpose? (9 pts)

Questions from Chapter 2

2.1 What is the purpose of system calls? (5 pts)

2.13 Describe three general methods for passing parameters to the operating system. (6 pts)

2.21 What is the main advantage of the microkernel approach to system design? How do user programs and system services interact in a microkernel architecture? What are the disadvantages of using the microkernel approach? (9 pts)

Questions from Chapter 3

3.8 Describe the differences among short-term, medium-term, and long-term scheduling. (7 pts)

3.9 Describe the actions taken by a kernel to context-switch between processes. (6 pts)

3.12 Including the initial parent process, how many processes are created by the program as shown. (10 pts)

```
#include<stdio.h>
#include<unistd.h>
int main()
{
      int i;
      for (i=0; i<4; i++)
         fork();

      return 0;
  }
```

Q4. Assume that the following program contains no syntax errors. As it executes it will create one or more processes. Simulate the execution of this program and show how processes are created (7+8 pts)

```
#include<stdio.h>
main()
{
int m=10, n=5,count=1, mult=1;
while(count <3)
{
    if(m != 0)
    {
          m = fork(); n = n+25;
     }
     else
     {
           m = fork(); n = n+20; mult = mult*n;
      }
      printf(“ n = %d       mult  = %d”, n, mult);
      count =count + 1;
   }
}
```

1) What is total number of processes? Show your work.
2) What will this program print on the screen when it executes?

Questions from Chapter 4

4.7 Under what circumstances does a multithreaded solution using multiple kernel threads provide better performance than a single-threaded solution on a single-processor system? (6 pts)

4.10 Which of the following components of program state are shared across threads in a multithreaded process? (4 pts)

a, Register values

b, Heap memory

c, Global variables

d, Stack memory

4.14 Consider a multiprocessor system and a multithreaded program written using the many-to-many threading model. Let the number of user-level threads in the program be greater than the number of processors in the system. Discuss the performance implications of the following scenarios. (9 pts)

1), The number of kernel threads allocated to the program is less than the number of processors.

2), The number of kernel threads allocated to the program is equal to the number of processors.

3), The number of kernel threads allocated to the program is greater than the number of processors but less than the number of user- level threads.