**CS 3343 Operating Systems Assignment 5 14 points**

**Due March 13 at 5pm**

**One homework assignment submission per student. Microsoft Word format only. No AI or GPT use. Cite all references.**

**Email your answers to me at** [**harringp@nsuok.edu**](mailto:harringp@nsuok.edu)

**Send emails exclusively via** [**https://mail.google.com/**](https://mail.google.com/)

**Sign in with your NSU email and password**

**Chapter 5:**

**Part 1: Use the slides or textbook definitions to answer the following (8 points)**

1. When do CPU scheduling decisions take place? Is the long term or short term scheduler used? **(Ch 5 Slide 7, Appendix C 22)**

According to the slides, CPU scheduling decisions take place…

* When a process switches from running to waiting state
* When a process switches from running to ready state
* When a process switches from waiting to ready
* And when a process terminates

In the Appendix C, it says what has been referred to as CPU scheduling corresponds closely to the short-term scheduling of Chapter 3. This is the only information I can find on the CPU scheduling and scheduler being used in the book. So, I believe it uses the short-term scheduler.

1. What is dispatch latency? **(Book Pg 203)**

Dispatch latency is the time it takes for the dispatcher to stop one process and start another running.

1. What is involved in switching to another process? **(Book Pg 202-205)**

The CPU scheduler and the Dispatcher are involved. There is also a scheduling algorithm that is involved which involves things like CPU utilization, throughput, turnaround time, waiting time, and response time.

1. Compare and contrast First come first served, shortest job first, priority, and round robin scheduling. **(Book Pg 206-212)**

**First Come First Served** = Process that requests the CPU first is allocated to the CPU first. This can lead to a heavy process taking up a lot of time that could instead be used for lighter processes first. The Gantt Chart shows P1, P2, and P3, with P1 having a 24 burst, P2 with a 3 burst, and P3 with a 3 burst. This leads to a waiting time of 17 milliseconds, whereas if it was reversed as P3, P2, P1, then it would be 3 milliseconds.

**Shortest Job First** = This takes the shortest processes and orders them first. If a process burst time is tied, then FCFS is used. It is considered to be more optimal than FCFS

**Priority** = With this method, a priority is attached to each process. The CPU is allocated to the highest priority first. Equal priorities are once again tied to FCFS.

**Round Robin Scheduling** = Similar to FCFS, however, there is preemption involved that enables the system to switch between processes. The ready queue in this algorithm is treated as a roundtable rather than a straight line. This leads to a Gantt Chart that looks like P1 = 24, P2 = 3, P3 = 3. And the order is P1 for 4 milliseconds, P2 for 3, P3 for 3, and then it goes back to P1 again in intervals of 4 milliseconds until it’s finished.

1. What are the two types of thread scheduling? **(Book Pg 217-218)**

Thread scheduling involves contention scope. We have process contention scope and system-contention scope.

PCS = on systems implementing the many-to-one and many-to-many models, the thread library schedules user level threads to run on an available LWP.

SCS = kernel thread scheduled onto an available CPU. In using this, there is competition for the CPU among all threads in the system. Systems using one-to-one model like Windows schedule threads only using SCS

1. What types of scheduling can be done with multiple CPUs? **(Book Pg 220)**

More complex when multiple CPUs are involved. We can use asymmetric or symmetric multiprocessing to help alleviate the issues. In Asymmetric, there is a master server that accesses all the system data structures. In Symmetric, each processor is self-scheduling.

1. What are multicore processors and give examples of multicore multithreaded capable operating systems? **(Book Pg 221)**

Multicore processors are multiple computing cores on the same physical chip. This is used in most contemporary computer hardware nowadays. So, an example of a capable operating system would be Windows. Inside almost every Windows PC nowadays, there is a multicore processor doing work.

1. List the windows thread/process priority levels (based on windows XP): **(Book Pg 239-240)**

For Windows XP, there are priority classes and relative priorities. These can form together to create a table with the priority classes for the columns and the relative priorities as the rows.

Priority Classes:

* Real Time
* High
* Above Normal
* Normal
* Below Normal
* Idle

Relative Priorities

* Time Critical
* Highest
* Above Normal
* Normal
* Below Normal
* Lowest
* Idle

**Part 2: Java programming (6 points):**

Modify the java program code below so that it runs as a standalone program and also makes a thread of high priority. Print to the screen when a thread is created. Include screenshots of your running program.

public class Assignment5Thread implements Runnable

{

public void run() {

Thread.currentThread().setPriority(Thread.NORM\_PRIORITY + 3);

}

}

public class Assignment5

{

    public static void main(String[] args)

    {

        Thread thread = new Thread(new Assignment5Thread());

        thread.start();

    }

}

class Assignment5Thread implements Runnable

{

     public void run()

     {

        //I changed the thread priority from NORM to MAX

        Thread.currentThread().setPriority(Thread.MAX\_PRIORITY);

         System.out.println("-- New Thread Created --\n" + "Priority: " + Thread.currentThread().getPriority());

     }

}

//I found a good article from GeeksForGeeks that helped out with this assignment.

//LINK: https://www.geeksforgeeks.org/runnable-interface-in-java/?ref=lbp

