Minia University
Faculty of Engineering
Computers and Systems Engineering Department

Title: **Project**

Course: **Operating Systems**

Lab No.: **N/A**

Date: **12 April 2016** ***Due:*** **13 May 2016** ***Time:*** **1 Month**

Show will, and you will have the break you wanted

Introduction:

The assignment of physical processors to processes allows processors to accomplish work. The problem of determining when processors should be assigned a specific process and to which processes is called processor scheduling or CPU scheduling.

When more than one process need to be executed, the operating system must decide which one will run first. The part of the operating system concerned with this decision is called the scheduler, and algorithm it uses is called the scheduling algorithm.

Project Objectives:

1. Proofing that Operating System subject will eventually make you go BANANAS, and we can’t help it.

Project Description:

In this project, you are to implement a C++ application that simulate the short-term scheduler in operating systems. Realize that a real CPU scheduler will probably contain more functionality than in this simulator, this simulator is to only demonstrate how some of CPU scheduling mechanisms work.

A- Simulation Inputs structure:

- **Initial burst time**
  This is the amount of CPU time the process will require. In real life this is not really known, but can be predicted with some degree of accuracy.
- **Delay**
  The time separating the arrival of processes. The amount of time after the (n-1)th process arrives that this process arrives.
- **Priority**
  For prioritized algorithms this is the relative weight of this process. The range is from 0 - 9 where 9 is the lowest priority and 0 is the highest
- **Auxiliary and tracking data**

  a. **PID** (process identifier)
     A Process ID is number to identify a particular process. Each PID is unique to a process.
  b. **Arrival start and finish time**
     Record the time a job arrives, when it is started, and when it is finished. These times allow us to derive three quantifiers.

     i. Response Time: the amount of time a ready job sat in the queue before the process was started.
     ii. Turnaround Tim: the total life of the process in the ready queue.
     iii. Wait Tim: the time this process spent in the ready queue waiting for CPU time.

B- Jobs Scheduler "THE ALGORITHMS":

Schedule jobs and dispatches the job to the first available processor (assume single core processor). This will be according to a pre-selected scheduling algorithm of the following algorithms:

NOTE: ASSUME ALL ALGORITHMS AS NON PREEMPTIVE

1. **First Come First Serve (FCFS)**
   This is the simplest algorithm. It implements a FIFO queue. The first job that arrives in the ready queue is run to completion, and then the next job is selected in arrival order. The FCFS algorithm is neither prioritized nor preemptive.
2. **Shortest Job First (SJF) (Non-preemptive)**
   The SJF algorithm chooses the shortest job. Without preemption jobs run to completion before a new job is selected . If preemption is enabled then the instant a job arrives that is shorter than the one being run the CPU switches to the new shorter job. The processing can be interrupted to run newly arrived shorter jobs.
3. **Round Robin (RR) (Prioritized, Equal time)**
   The Round Robin scheduling algorithm allocates a timeslice to each running process. This is called the quantum and it represents the number of CPU cycles a process gets befor the scheduler searches for a new job to run. Jobs recieve their quantums of CPU time in FCFS order. With priorit scheduling enabled the quantum is multiplied by the magnitude of a processes priority. Thus more important jobs get more CPU time.
4. **Priority Weighted (PRI) (Non-preemptive)**
   The priority scheduling algorithm selects its next job based on the importance of the process. The job that has the highest priority (0 high : 9 low) is run first. If preemption is enabled the new jobs with a higher priority will interrupt the currently executing job. Without preemption the highest priority job is chosen after the active process finishes execution.

5. **BOUNCE +10 Multi-Level queue: implement a 4 multi-level queue scheduler, with queue 1 has the highest priority and queue 4 with the lowest performance.**

C- Real Input

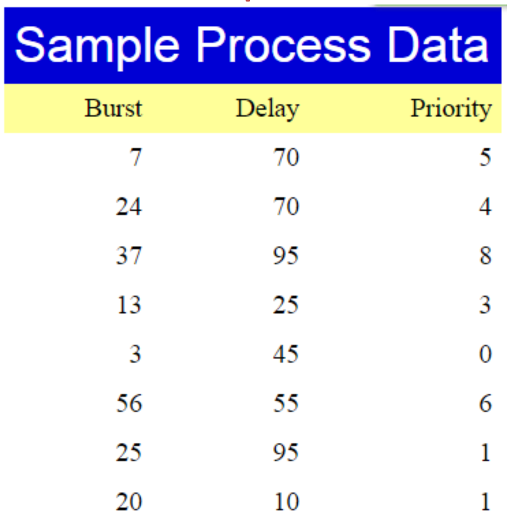
The input is 50 tasks or process. You will find it in the text file “**input (50 tasks).txt**” in the project main folder. For the PID , you can create a random function that will generate a unique ID for each process.

D- Results:

The result will be the mean waiting time, the standard deviation waiting time and the average timing of each set of tasks for each algorithm. Notice that the book uses the average waiting time only, but the average is not a good indicator for performance, hence we measure the mean and standard deviation.

Example: note that this table is not complete; a complete table can be found in the result folder included in the project folder.

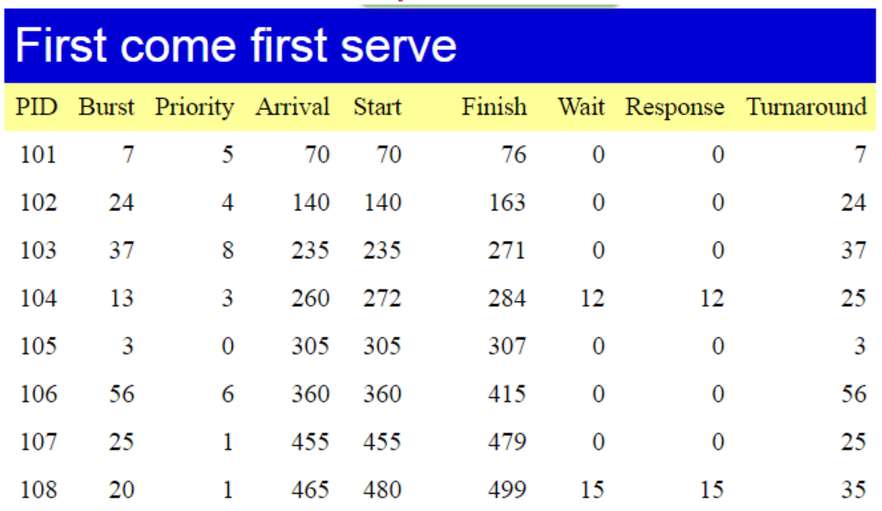
Input



Sample Process Data

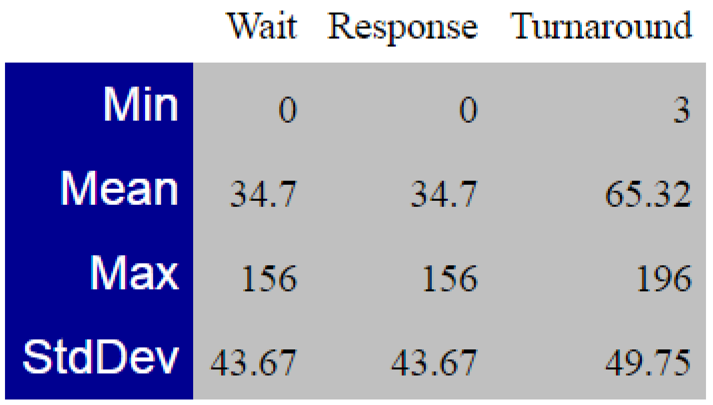
Burst	Delay	Priority
7	70	5
24	70	4
37	95	8
13	25	3
3	45	0
56	55	6
25	95	1
20	10	1

Output for FCFS



First come first serve

PID	Burst	Priority	Arrival	Start	Finish	Wait	Response	Turnaround
101	7	5	70	70	76	0	0	7
102	24	4	140	140	163	0	0	24
103	37	8	235	235	271	0	0	37
104	13	3	260	272	284	12	12	25
105	3	0	305	305	307	0	0	3
106	56	6	360	360	415	0	0	56
107	25	1	455	455	479	0	0	25
108	20	1	465	480	499	15	15	35



	Wait	Response	Turnaround
Min	0	0	3
Mean	34.7	34.7	65.32
Max	156	156	196
StdDev	43.67	43.67	49.75

Tools:

1- Microsoft Visual Studio 2013 or later.

Deliverables:

1- The full Visual Studio project, compress the project folder (you must comment your code otherwise project without comments will lose 20% of the total marks).
2- A comparison of each algorithm performance with respect to mean and standard deviation. Your project should output this to the screen.
3- To verify your results , compare it with the results found in the subfolder output in the main project folder. Note: you will need Microsoft Excel to open these .xls files.

References:

1- http://jimweller.com/jim-weller/jim/java_proc_sched/index.html
2- http://www.tutorialspoint.com/operating_system/os_process_scheduling.htm
3- YouTube playlist: https://www.youtube.com/watch?v=KtuQpQwlmYM&list=PLW1OMpQZxu7xN327CJXp3CsLoNo8eco0l
4- Abraham Silberschatz, Operating System Concepts, 9th edition.

It is common that the first place always gets the prize, but that is for normal people and we are not normal, we are BANANAS. The first place doesn't always mean that you completed 100% of the project. Hence, this T-shirt will be the gift for the team who shows how far they will fly to accomplish this project. So, the more you put efforts in your project, the more likely u r to win this t-shirt. Efforts mean, understanding all of the project parts, writing the code yourself, and doing it your own way, not the way of some other BANANA on the internet.