PALESTINE POLYTECHNIC UNIVERSITY
COLLEGE OF INFORMATION TECHNOLOGY AND COMPUTER ENGINEERING
OPERATING SYSTEMS 7505
FINAL PROJECT DESCRIPTION

1- Students should start working on the implementation of this project as soon as possible so as finish the requirements on time and with no delays.
2- You have 2 problems to solve using C/C++ or Java programming language. Or any other language of your choice.
3- The project is out of 100 (50 % for part 1 and 50 % for part 2).
4- You should work in person (single student), **however you may share and discuss ideas with your collogues, if so you should state clearly in your project report with whom students you discussed and shared ideas. Otherwise, similar ideas will be considered as plagiarism**
5- The program code must be written in good manner and must be documented. An execution file also with source code should be attached in your project files.
6- **Deadline is 1-5-2020. No late submission is allowed after this date.**
7- **Any copy or plagiarism will yield to the failure in the whole course of all students participating in this illegal action.**

Projects Input File for both parts 1 and 2.

The input file for the 2 project parts will contain the following information:

- Physical memory size = msize… (Example 4096 bytes) Power of 2 (will be used in Part 2)
- Page size = Frame size = psize = fsize = ….. (example = 512 bytes). Power of 2 (will be used in Part 2)
- Round Robin Quantum = Q = ….. (example 10).
- Context Switch = CS = ……. (example 1)
- Each process information as :

  Process ID Arrival Time CPU Burst Size in Bytes (used in project 2)

Sample file format process.txt:

```
4096
512
10
1
0    3    10   8192
1    0    12   2048
2    1    3    512
3    5    21   4096
4    9    7    1024
     .
     .
```

PALESTINE POLYTECHNIC UNIVERSITY
COLLEGE OF INFORMATION TECHNOLOGY AND COMPUTER ENGINEERING
OPERATING SYSTEMS 7505
FINAL PROJECT DESCRIPTION

Part1: Simulate a CPU scheduler: 50 %.

- Each student should work separately to design and implement this project. All required explanations should be included inside your project and should be very clear.
- The idea of the project is to receive a list of processes n processes (dynamic number of process) with different arrival times and different CPU bursts. Also, each process will have a size in bytes (will be used in Part 2).
- You should write a program that will read these processes from a file called ***processes.txt*** that contains the list of process that should be scheduled. A sample input is like the file below.
- Your simulator should build the PCB for each process with required data structure. Your design for the PCB should be clear and clarified inside the source code of your simulator or as a sperate documentation file using word or pdf format.
- Your simulator should be able to report the following results on the screen as soon as the last process is finished running using FCFS, SJF (non-preemptive), and RR with Q = y that will be given in the text file as well.

a. (10 %) A Gantt Chart that shows the execution times of the process, similar to what we had in the lecture examples.
b. (10 %) Finish time of each process, Waiting time for each process, Turnaround time for each process. Assume a CS = x time units that will be input from the text file as well.
c. (10 %) The averages of the required criterions in part a. Average Waiting Time ….
d. (10 %) The CPU utilization.
e. Documentation and report.

Bonus (15%)
A bonus of 15 points will be given if you implement the project using threads and dynamic process creation and execution.

PALESTINE POLYTECHNIC UNIVERSITY
COLLEGE OF INFORMATION TECHNOLOGY AND COMPUTER ENGINEERING
OPERATING SYSTEMS 7505
FINAL PROJECT DESCRIPTION

Part2 : Simulate a Paging Memory Manager: 20 %.

- Each student should work separately to design and implement this project. All required explanations should be included inside your project and should be very clear.
- This project can be done using any programming language like C/C++ or JAVA.
- The idea of the project is to receive a list of processes (n processes) with different sizes in bytes.
- You should write a program that will read these processes from a file called ***Processes.txt*** that contains the list of process that should be loaded into physical memory of size msize. A sample input is like the file in Part 1.
- Your simulator should build the page table for each process with required data structure. Your design for the Page table should be clear and clarified inside the source code of your simulator or as a sperate documentation file using word or pdf format.
- Should use the memory size and the page/frame size as given in the input file.
- Assume that each page of each process will be a allocated in a free frame using a random number generator that gives randomly the free frame number that will be used to allocate each page of each process. All frames are free at the beginning of the run of the simulator.
- Your simulator should be able to report the following results on the screen as soon as all 5 processes are loaded into memory frames.
  a. (20 %) The page table of each of the 5 processes.
  b. (10 %) The memory map of the physical memory showing filled frames and free frames. Similar to the map as in the slide's examples.
  c. (10 %) Your simulator should be able to accept interactively a **logical address** from the user and map it to a physical address based on the resulted allocation of pages in the physical frames as parts a and b above. (Example: logical address of 1050 is in physical address 2074 assuming a frame/page size of 1024 and page 1 is loaded in frame 2. That is your output will be in the format

     $$1050 \rightarrow (p, d) = (1, 26) \rightarrow (f, d) = (2, 26) \rightarrow 2074$$

  d. (10 %) Documentation.