A picture containing icon

Description automatically generatedLogo, company name

Description automatically generated

OS Phase1 Report

Submitted to:

Dr. Ayman Aboelhassan

TA. Muhammad Alaa

TA. Muhammad Hesham

TA. Ahmed Mostafa

TA. Ali Haytham

Submitted by:

Moaaz Tarek 1200871

Salah Mohamed Salah 1200806

Omar Sherif Elzahar 1200476

Hussein Mostafa Elhawary 1200799

Data Structures

## Linked List:

Dynamic List to be used in RR as a circular queue.

Node in List will save pointer to next and previous nodes. It will also contain PCB pointer which is the process details.

A screen shot of a computer program

Description automatically generated

## Priority Queue:

Used to sort in HPF by priority, and in STRN by remaining time.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

This is a process struct that I save data in it from given input file (process.txt)

While PCB struct I save in it all data related to each process when process generator send process to scheduler then any update during running will be modified in PCB struct

Algorithms

All algorithms have near identical skeleton in receiving, deleting and handle the algorithm

A computer screen with text

Description automatically generated

Algorithm, addToDS are pointer to functions passed to this function to handle different data structures in RR ,HPF and SRTN.

# Round Robin:

Start by checking if the list is empty, if it is then return from handling RR.

A black background with white text

Description automatically generated

Then check if there was a PCB that finished and removed form LL if yes goes to next pointer and start the process

A screen shot of a computer program

Description automatically generated

Else it checks if it is time for quantum and sets node if null

A computer screen with white text

Description automatically generated

or iterate circulary and start the process.

A screen shot of a computer program

Description automatically generated

It also ends the process if not finished at quantum time and sets remaining time to synchronize.

A computer screen with text

Description automatically generated

Then if remaining time is less than the quantum it will decrease remaining time each second to synchronize between process and scheduler. Because a process can finish before a quantum time has passed.

A screen shot of a computer

Description automatically generated

### Testcase:

A screenshot of a computer

Description automatically generated

### Output:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

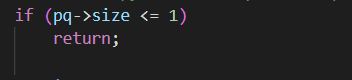
Description automatically generated

### Assumption:

when process is received at quantum time it should start if last running was the previous process.

# Shortest Remaining Time Next:

Start by checking if the Priority Queue is empty, if it is then return from handling SRTN.



If no process is running and a new process arrives start it and process its data

A screen shot of a computer code

Description automatically generated

If a process has finished start a new process with the shortest remaining time

A screen shot of a computer code

Description automatically generated

If a new process arrives that has a shorter remaining time than all other processes stop the current process and start the one that just arrived

A screen shot of a computer program

Description automatically generated

To keep track of the remaining time the scheduler for the Priority Queue to work correctly A black screen with white text

Description automatically generatedTestcase:

**A screenshot of a computer program

Description automatically generated**

Output:

scheduler.log file

**A screenshot of a computer program

Description automatically generated**

Scheduler.perf

**A screenshot of a computer program

Description automatically generated**

# HPF**:**

Start by checking if the Priority Queue is empty, if it is then return from handling HPF.

A black background with white text and numbers

Description automatically generated

If no process is running and a new process arrives start it and process its data

A screen shot of a computer code

Description automatically generated

If a process has finished start a new process with the highest priority

A screen shot of a computer program

Description automatically generated

Testcase:

**A screenshot of a computer program

Description automatically generated**

Output:

scheduler.log file**A screenshot of a computer program

Description automatically generated**

Scheduler.perf

**A screenshot of a computer program

Description automatically generated**

**Assumptions:**

No process should have runtime equal to Zero.

In SRTN if process 2 was running and @ time 11 its remaining time was 4 and process 5 came with a runtime 4 process 2 will remain running then process 5 will run after process 2 finishes.

In HPF if two processes came at same time with same priority it will run according to ascending order of #id.

In RR when process is received at quantum time it should start if last running was the previous process.

|  |  |  |  |
| --- | --- | --- | --- |
| Moaaz Tarek | RR | Process Generator | 20% Skeleton |
| Salah Mohamed | HPF | 80% Skeleton | Process.c |
| Omar Sherif | SRTN | OuputFiles |  |
| Hussein Mostafa | SRTN, HPF | headers |  |

Work Loads:

|  |  |
| --- | --- |
| HPF | 3 hours + 10 debugging |
| SRTN | 40 min + 12 hours debugging |
| RR | 1 hour + 8 debugging |
| Skeleton | 5 hours |
| Process Generator | 1~2 hours |
| Process | 30 mins + 2 hours debugging |
| Headers | 1 hour |
| OutputFiles | 2~3 hours + 4 debugging |