Operating Systems Lab

Endsemester Exam

Descriptive Part

-Maanas Bhardwaj

-AU1940047

-Section-2

Question 2_Part-A

Round-Robin Scheduling-

In Round-Robin scheduling algorithm, each process gets executed for a specific time (time quantum) in a cyclic manner. So basically processes are alligned in a ques on the basis of their arrival time but after a fixed amount of time (time quantum), the CPU shifts to next process and the process which was being executed earlier sent to the end of the queue.

Modified Round-Robin Scheduling-

Normally, RR approach is not prioritised in the analysis, resulting in even worse CPU performance. The Modified RR algorithm improves the efficiency of the traditional RR algorithm by taking into account all factors such as process priority, burst time, and arrival time to arrange them in the ready queue and generate TQ accordingly, reducing Waiting time, Turn-around time, and the number of context switches.

Code Explanation of RR

Firstly the initialization of variables is done.

PN= Total number of processe

X== Number of processes completed

Time_quantum== for time quantum (fixed time)

Turnaround_time

Arrival_time

Average_turnaround_time

Average_ time

First, we will ask the user for total number of processes and quantum time.

Implementation of for loop is done to go through the processes.

If the burst time of the given process< quantum time, that means the process is completed and hence we increament the counter variable.

If the burst time is not less than the quantum time, then we simply subtract the quantum time from its burst time.

This will go on for every process. Then the calculation for average burst time and average turnaround time is done.

Pseudocode Explaination of Modified RR (given in the reference Research Paper)

When a computer starts up, it utilises a certain time quantum (TQ). TQ changes when new programmes are added to the system. When a large number of processes arrive at the same time, the algorithm sorts them by burst times (BT). The median of the burst times is then computed. Because the median represents the midpoint value, if this number is used as TQ, half of the programmes stored into the system will be fully run. When this figure is determined again for the next cycle, half of the remaining processes will be run, and so on.

The mean and median of the burst times are compared, and the one that is bigger is multiplied by the greatest burst time in order to achieve a better time quantum. The time quantum is then calculated by taking the square root of this number. When processes arrive at various times, they are initially organised based on the arrival timings. After that, a score is computed for each procedure. This score indicates the importance of a specific procedure.

The score is calculated as-

[(0.7 * priority of process) + (0.3 * remaining time of processing)].

Question-3

Producer Consumer Problem

The Producer-Consumer Problem consists of a Producer who procuces something and one Consumer who consumes what Producer been poducing. Both Producer and Consumer share same fixed-size memory buffer.

The job of Producer is to prodce data and store it in a buffer, and then generate the data again. The Consumer has the job to consume the data produced by the Producer from the buffer.

Mutex

Through Mutex, either producer or consumer can proceed at a time. Untill buffer is filled by Condumer, the Producer has to wait and vice-versa.

Semaphore

Semaphore is generalized Mutex.

Binary Semaphore-

0 and 1 are the only values it can have. Its value is initialized to 1.

Counting Semaphore-

It can have values other than 0 and 1.

Pseudocode

The pseudocode has been writen in the code file.

The wait operation decreament the value of semaphore and the signal operation increaments the value.

So when producer does some task the value of full will increase and empty will decrease. And when the consumer performs the process is done vice-verca.