Name : Vansh Devgan

Roll No. : 62

Registration No. : 11804339

Email Id : blogwithvansh@gmail.com

Subject : Operating System [CSE 316]

Github-link : https://github.com/vanshdevgan/

Submitted to : Manpreet Singh

Question 10:

Write a C program to solve the following problem: Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO
order,is:
86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130
Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the FCFS disk-scheduling algorithms?

Solution:

Using FCFS-First Come First Serve
The drive is currently serving a request at cylinder at 143

143-86+1470-86+1470-913+1774-913+1774-948+1509-948+1509-1022+1750-1022+1750-130 =7081

> The total seek distance is = 7081

Link To Code of given problem is in My This GitHub repo: https://github.com/vanshdevgan/os_assigment_62.git

Steps to Run It

1- Make Sure You Have Git Installed In Your System

2- Make Sure You Run This In Kali Linux Or Parrot OS

3- Make Sure You Don't Delete Lib Folder as It's Essential For Program To Run

4- Open Terminal & Type
git clone repo_link_should_be_here

5- cd repo_name_here

6- gcc os.c -o main

7- chmod +x main.out

8- ./main.out

> The output will be 7081

We can easily solve this problem with other algorithm SSTF, SCAN and LOOK.

Question-17:

Design a scheduling program to implements a Queue with two levels:

Level 1 : Fixed priority pre-emptive Scheduling
Level 2: Round Robin Scheduling

Solution:

For a Fixed priority pre-emptive Scheduling (Queue 1), the Priority 0 is highest priority. If one
process P1 is scheduled and running, another process P2 with higher priority comes. The New
process (high priority) process P2 pre-empts currently running process P1 and process P1 will go
to second level queue. Time for which process will strictly execute must be considered in the
multiples of 2..
All the processes in second level queue will complete their execution according to round robin
scheduling.
Consider: 1. Queue 2 will be processed after Queue 1 becomes empty.
2. Priority of Queue 2 has lower priority than in Queue 1

1. Fixed priority pre-emptive Scheduling (Queue 1)

* Priority 0 is highest priority.

* Quantum: 4-unit time

* Pre-emptive:

If one process e.g. P1 is scheduled and running, now another process with higher priority comes e.g. P2. New process (high priority)

process P2 pre-empts currently running process P1 and process P1 will go to second level queue.

2. Round Robin Scheduling (Queue 2)

* Quantum: 4 unit time

* All the processes in second level queue will complete their execution according to round robin scheduling.

* Queue 2 will be processed after Queue 1 becomes empty.

* Priority of Queue 2 has lower priority than in Queue 1.

Suppose Queue 1 is empty and currently process from Queue 2 is being executed. Now, If at this time a new process arrives then new process will be part of Queue 1. So, new

process should be scheduled as Queue 1 has higher priority than Queue 2. Again, after Queue 1 becomes empty Queue 2 will resume execution.

```
============================================================
          INPUT FORMAT
===========================================================
<pid>  <arrival_time>  <burst_time>  <priority>
===========================================================
          OUTPUT FORMAT
============================================================
<pid Response_Time Finish_Time Waiting_Time >
============================================================
```

Sample Input :

```
===============================================================
5
1 0 14 2
2 7 8 1
3 3 10 0
4 5 7 2
5 1 5 3
===============================================================
```

Sample Output:

```
==============================================================
1 0 44 30
2 0 31 16
3 0 41 28
4 6 34 22
5 14 35 29
```