OPERATING SYSTEM LAB 3

2021 / 04 / 27
Student: HAOXIANG YANG
ID :302485

--[**TASK**] ----------

Create a configuration file in which all processes run an average of 2000 milliseconds with a standard deviation of zero, and which are blocked for input or output every 500 milliseconds. Run the simulation for 10000 milliseconds with 2 processes. Examine the two output files. Try again for 5 processes. Try again for 10 processes. Explain what's happening.

Introduction:

Preemptive scheduling: Preemptive Scheduling is defined as the scheduling which is done when the process changes from running state to ready state or from waiting for the state to ready state

Non-Preemptive scheduling: Non-preemptive Scheduling is used when a process terminates, or a process switches from running to waiting state.

I/O block:

all programs perform I/O. A scheduler clearly has a decision to make when a job initiates an I/O request, because the currently running job won't be using the CPU during the I/O; it is blocked waiting for I/O completion. If the I/O is sent to a hard disk drive, the process might be blocked for a few milliseconds or longer, depending on the current I/O load of the drive. Thus, the scheduler should probably schedule another job on the CPU at that time. The scheduler also has to make a decision when the I/O completes. When that occurs, an interrupt is raised, and the OS runs and moves the process that issued the I/O from blocked back to the ready state.

First Come First Serve (FCFS):

- Jobs are executed on first come, first serve basis.
- It is a non-preemptive, pre-emptive scheduling algorithm.
- Easy to understand and implement.
- Its implementation is based on FIFO queue.
- Poor in performance as average wait time is high.

Batch mode: execution of series of programs in a way that the human intervention is unnecessary.

Observed results and explanations:

```
// # of Process
numprocess 2

// mean deivation
meandev 2000

// standard deviation
standdev 0

// process     # I/O blocking
process 500
process 500


// duration of the simulation in milliseconds
runtime 10000
```

```
Scheduling Type: Batch (Nonpreemptive)
Scheduling Name: First-Come First-Served
Simulation Run Time: 4000
Mean: 2000
Standard Deviation: 0
Process #       CPU Time        IO Blocking     CPU Completed   CPU Blocked
0               2000 (ms)       500 (ms)        2000 (ms)       3 times
1               2000 (ms)       500 (ms)        2000 (ms)       3 times
```

```
Process: 0 registered... (2000 500 0 0)
Process: 0 I/O blocked... (2000 500 500 500)
Process: 1 registered... (2000 500 0 0)
Process: 1 I/O blocked... (2000 500 500 500)
Process: 0 registered... (2000 500 500 500)
Process: 0 I/O blocked... (2000 500 1000 1000)
Process: 1 registered... (2000 500 500 500)
Process: 1 I/O blocked... (2000 500 1000 1000)
Process: 0 registered... (2000 500 1000 1000)
Process: 0 I/O blocked... (2000 500 1500 1500)
Process: 1 registered... (2000 500 1000 1000)
Process: 1 I/O blocked... (2000 500 1500 1500)
Process: 0 registered... (2000 500 1500 1500)
Process: 0 completed... (2000 500 2000 2000)
Process: 1 registered... (2000 500 1500 1500)
Process: 1 completed... (2000 500 2000 2000)
```

Conclusion:

As we observed that there are 2 processes runs, each process runs an mean time 2000ms.Therefore, whole simulation runs 4000ms,which not beyond 10000ms (duration of simulation).Because the processes were scheduled based on NonPreemptive scheduling and FCFS(First Come First Served) algorithm, when a process is being executed only after the previous run process is I/O blocked and when 2nd process is I/O blocked we return to execute the one blocked before running the current process. so, there is a switch in every 500ms between running status and waiting status, which means the current executing process is being blocked and previous blocked process will continue running. When final execute ends, then means get into terminated status, the process completes its job. Note that each blockade lasts 500ms.

```
// # of Process
numprocess 5

// mean deivation
meandev 2000

// standard deviation
standdev 0

// process     # I/O blocking
process 500
process 500
process 500
process 500
process 500

// duration of the simulation in milliseconds
runtime 10000
```

```
Scheduling Type: Batch (Nonpreemptive)
Scheduling Name: First-Come First-Served
Simulation Run Time: 10000
Mean: 2000
Standard Deviation: 0
Process #       CPU Time        IO Blocking     CPU Completed   CPU Blocked
0               2000 (ms)       500 (ms)        2000 (ms)       3 times
1               2000 (ms)       500 (ms)        2000 (ms)       3 times
2               2000 (ms)       500 (ms)        2000 (ms)       3 times
3               2000 (ms)       500 (ms)        2000 (ms)       3 times
4               2000 (ms)       500 (ms)        2000 (ms)       3 times
```

```
 1 Process: 0 registered... (2000 500 0 0)
 2 Process: 0 I/O blocked... (2000 500 500 500)
 3 Process: 1 registered... (2000 500 0 0)
 4 Process: 1 I/O blocked... (2000 500 500 500)
 5 Process: 0 registered... (2000 500 500 500)
 6 Process: 0 I/O blocked... (2000 500 1000 1000)
 7 Process: 1 registered... (2000 500 500 500)
 8 Process: 1 I/O blocked... (2000 500 1000 1000)
 9 Process: 0 registered... (2000 500 1000 1000)
10 Process: 0 I/O blocked... (2000 500 1500 1500)
11 Process: 1 registered... (2000 500 1000 1000)
12 Process: 1 I/O blocked... (2000 500 1500 1500)
13 Process: 0 registered... (2000 500 1500 1500)
14 Process: 0 completed... (2000 500 2000 2000)
15 Process: 1 registered... (2000 500 1500 1500)
16 Process: 1 completed... (2000 500 2000 2000)
17 Process: 2 registered... (2000 500 0 0)
18 Process: 2 I/O blocked... (2000 500 500 500)
19 Process: 3 registered... (2000 500 0 0)
20 Process: 3 I/O blocked... (2000 500 500 500)
21 Process: 2 registered... (2000 500 500 500)
22 Process: 2 I/O blocked... (2000 500 1000 1000)
23 Process: 3 registered... (2000 500 500 500)
24 Process: 3 I/O blocked... (2000 500 1000 1000)
25 Process: 2 registered... (2000 500 1000 1000)
26 Process: 2 I/O blocked... (2000 500 1500 1500)
27 Process: 3 registered... (2000 500 1000 1000)
28 Process: 3 I/O blocked... (2000 500 1500 1500)
29 Process: 2 registered... (2000 500 1500 1500)
30 Process: 2 completed... (2000 500 2000 2000)
31 Process: 3 registered... (2000 500 1500 1500)
32 Process: 3 completed... (2000 500 2000 2000)
33 Process: 4 registered... (2000 500 0 0)
34 Process: 4 I/O blocked... (2000 500 500 500)
35 Process: 4 registered... (2000 500 500 500)
36 Process: 4 I/O blocked... (2000 500 1000 1000)
37 Process: 4 registered... (2000 500 1000 1000)
38 Process: 4 I/O blocked... (2000 500 1500 1500)
39 Process: 4 registered... (2000 500 1500 1500)
```

Conclusion:

As I observed that we have 5 processes and whole all processes running time equals to duration of simulation is 10000ms,each process runs 2000ms.Because the processes were scheduled based on Non-Preemptive scheduling and FCFS(First Come First Served) algorithm, when a process is being executed only after the previous run process is I/O blocked and when 2nd process is I/O blocked we return to execute the one blocked before running the current process. So, there is a switch in every 500ms between running status and waiting status, which means the current executing process is being blocked and previous blocked process will continue running. The blockade lasts 500ms. It can be noticed that the processes were executed and switched in pairs. They executed regular after the completion of 1st pair of processes, and next pair of processes will start executing and so on. Notice that in “Summary-Processes”, the 4th process (counting from 0 to 4) is executed in

single, because it's the only one exist (only one not execute in pair) after 2 previous pairs were finish, there are no more other process left. One more thing, there is not message inform that 4th process is completed. I guess that since duration of simulation is 10000ms, it may not enough to show this message, we may need 1 more 500ms to show it.

```
// # of Process
numprocess 10

// mean deivation
meandev 2000

// standard deviation
standdev 0

// process    # I/O blocking
process 500
process 500
process 500
process 500
process 500
process 500
process 500
process 500
process 500
process 500

// duration of the simulation in milliseconds
runtime 10000
```

```
Scheduling Type: Batch (Nonpreemptive)
Scheduling Name: First-Come First-Served
Simulation Run Time: 10000
Mean: 2000
Standard Deviation: 0
Process #       CPU Time        IO Blocking     CPU Completed   CPU Blocked
0               2000 (ms)       500 (ms)        2000 (ms)       3 times
1               2000 (ms)       500 (ms)        2000 (ms)       3 times
2               2000 (ms)       500 (ms)        2000 (ms)       3 times
3               2000 (ms)       500 (ms)        2000 (ms)       3 times
4               2000 (ms)       500 (ms)        1000 (ms)       2 times
5               2000 (ms)       500 (ms)        1000 (ms)       1 times
6               2000 (ms)       500 (ms)        0 (ms)          0 times
7               2000 (ms)       500 (ms)        0 (ms)          0 times
8               2000 (ms)       500 (ms)        0 (ms)          0 times
9               2000 (ms)       500 (ms)        0 (ms)          0 times
```

```
Process: 0 registered... (2000 500 0 0)
Process: 0 I/O blocked... (2000 500 500 500)
Process: 1 registered... (2000 500 0 0)
Process: 1 I/O blocked... (2000 500 500 500)
Process: 0 registered... (2000 500 500 500)
Process: 0 I/O blocked... (2000 500 1000 1000)
Process: 1 registered... (2000 500 500 500)
Process: 1 I/O blocked... (2000 500 1000 1000)
Process: 0 registered... (2000 500 1000 1000)
Process: 0 I/O blocked... (2000 500 1500 1500)
Process: 1 registered... (2000 500 1000 1000)
Process: 1 I/O blocked... (2000 500 1500 1500)
Process: 0 registered... (2000 500 1500 1500)
Process: 0 completed... (2000 500 2000 2000)
Process: 1 registered... (2000 500 1500 1500)
Process: 1 completed... (2000 500 2000 2000)
Process: 2 registered... (2000 500 0 0)
Process: 2 I/O blocked... (2000 500 500 500)
Process: 3 registered... (2000 500 0 0)
Process: 3 I/O blocked... (2000 500 500 500)
Process: 2 registered... (2000 500 500 500)
Process: 2 I/O blocked... (2000 500 1000 1000)
Process: 3 registered... (2000 500 500 500)
Process: 3 I/O blocked... (2000 500 1000 1000)
Process: 2 registered... (2000 500 1000 1000)
Process: 2 I/O blocked... (2000 500 1500 1500)
Process: 3 registered... (2000 500 1000 1000)
Process: 3 I/O blocked... (2000 500 1500 1500)
Process: 2 registered... (2000 500 1500 1500)
Process: 2 completed... (2000 500 2000 2000)
Process: 3 registered... (2000 500 1500 1500)
Process: 3 completed... (2000 500 2000 2000)
Process: 4 registered... (2000 500 0 0)
```

```
34 Process: 4 I/O blocked... (2000 500 500 500)
35 Process: 5 registered... (2000 500 0 0)
36 Process: 5 I/O blocked... (2000 500 500 500)
37 Process: 4 registered... (2000 500 500 500)
38 Process: 4 I/O blocked... (2000 500 1000 1000)
39 Process: 5 registered... (2000 500 500 500)
```

Conclusion:

As I observed that in this case we have 10 processes. The duration of simulation 10000ms is not enough to execute all processes. We need at least 20000ms (2000ms * 10) to run all 10 processes. As I wrote in case#2, here also have a switch in every 500ms between running status and waiting status, which means the current executing process is being blocked and previous blocked process will continue running. The blockade lasts 500ms. It can be noticed that the processes were executed and switched in pairs. Scheduler were scheduled based on Non-Preemptive scheduling and FCFS (First Come First Served) algorithm.

We can see only first 4 processes (2 pairs (from 0 to 3)) were completed. Actually, we can see that 6 processes, four processes (No.0 to 3) show completed and 2 processes (process 4 and process 5) are being proceed. Because of actual simulation time out of the range we set, the rest of processes didn't start.

The explanation of the process 5.

After we run the simulation for 10000 milliseconds with 2 processes, 5 process and 10 process. We can get the process state transfer.
All process will be in queue in ready state and waiting for execution, but only one process can be execute in Single processor system each time, so ready processes can be queued by multiple priorities (the process is completed by the I / O operation and enters the ready state, it is queued to the high priority queue).
If A process requests I / O operation , it enters a Blocked state until the I / O operation is completed.

So we can see from the summary process of 5 process that after 500(ms), process0 enter the Blocked state for 500(ms),
and at the same time, the second process, process 1 start into Running state.

And after process0 exit from Blocked state, it has the highest priority in Ready state, so when process1 into Blocked state, process 0 enter Running state and continue to execute.
If after running a specific time, the CPU time of the process has reached the required runtime for a process, it enter Terminated state, and it shows the completed at (2000 500 2000 2000),the process 0 is done, then process 1 is also finished , at the same time the process 2 enter the blocked sate, and the same time process 3 are enter into the running state, after process 3 into blocked state, process 2 are get into running state. And so on for the next steps.