[EOPSY]: Lab - task 3

***2020 / 05 / 11***

***Student: Gao Ning***

***Supervisor: Figat Maksym***

--[ **My 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.

1. **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.

1. **Observed results and explanations:**
   1. **Run the simulation for 10000 milliseconds with 2 processes.**

**Configuration**

**A screenshot of a cell phone

Description automatically generated**

**Summary-results:**

**A screenshot of a cell phone

Description automatically generated**

**Summary-processes:**

**A close up of text on a black background

Description automatically generated**

**Conclusion:**

# As I can see 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 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. When final execute ends, then means get into terminated status, the process completes its job. Note that each blockade lasts 500ms.

* 1. **Run the simulation for 10000 milliseconds with 5 processes.**

**Configuration:**

**A screenshot of a cell phone

Description automatically generated**

**Summary-results:**

**A screen shot of a smart phone

Description automatically generated**

**Summary-processes:**

**A picture containing table, sitting, black, computer

Description automatically generated**

**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.

* 1. **Run the simulation for 10000 milliseconds with 10 processes.**

**Configuration:**

**A screenshot of a cell phone

Description automatically generated**

**Summary-results:**

**A screenshot of a computer screen

Description automatically generated**

**Summary-processes:**

**A screenshot of a computer

Description automatically generated**

**Conclusion:**

As I observed that in this case, we have all the same setting as case#1 and case#2 except here 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.