**Tasks:**

* Lab 4: Process management
* Review Process management algorithms

**Definitions:**

**(**in terms for this lab not formal definitions)

* *Preemptive: When a new process arrives and has a higher priority than the current running process, Then the current running process is paused and put back in the waiting queue, and the new process starts executing. In most systems it will automatically interrupt the current running job when the new job arrives, however some systems use a pre-determined interval (a certain number of time units), to check for new jobs.*
* *Non preemptive: Take no action when a process with higher priority arrives, when a new process with a higher precedence arrives current running process need to finish first before the new process goes*

*Oxford definition:*

Preempt – to take action in order to prevent (an anticipated event) from happening; forestall.

* *Metric:* a metric is a numeric value which measures a particular element in a given system ,

Examples: average waiting time(ms,s,min), burst time/Execution(ms,s,min),

Metric definitions

* Burst time(bt): The amount of time a process will need to execute for(when executed multiple times)
* Execution time(et): The amount of time the process will need to complete all given tasks
* Arrival time(at): The first point in time the system knows about the process
* Start time(st): the time at which the process starts executing
* Completion time(ct):the time at which the process finishes executing, = st + et (non-pre-emptive only)
* Turnaround time(tat):the total time the process is waiting to finish its execution, = ct - at
* Avg turnaround time(att): = tat/(number of jobs), or Average(tat)
* Priority: A value which represents the precedence of the process, usually numeric. (depending on the specific system can be ascending or descending).
  + Example: the numbers 1 through 4, in descending order, meaning the value 1 has higher precedence than 2 , 2 is higher than 3 , 4 is the lowest.

**Algorithms Pseudo code:**

* First Come - first serve (fcfs): non-preemptive

Steps:

1) Check process queue, for jobs that have arrived and are waiting to execute

2)Execute the next job in the waiting queue to completion,

3)Repeat until job queue is empty.

* Shortest job Next(SJN): non-preemptive

Steps:

1)check process queue for jobs that have arrived and are waiting to execute.

2)Of the processes that are waiting, compare the execution time

3) Execute the next process that has the shortest execution time.

4)Repeat until no processes remain in the waiting queue

* Priority: preemptive

Steps:

Notes: Use 1-time unit as minimum execution time/ (minimum check time)

2 methods to check,

* + - System event handler,(not easy to do in a simulation)
    - Use a Quantum(pre determined length of time)

Pseudo Algorithm

1)check processes in the queue for jobs that have arrived and are waiting (every 1 time unit)

2)Of the jobs that have arrived and are waiting, compare the priority value,

3)Execute the next job that has the highest priority, for the next 1-time unit

4)repeat until job queue is empty