**Deliverable A**

Aaron Borjas, Ty Heim, Diana Diaz

3/21/2022

CS475W

**System requirements**

**Process entity support (Process Table, PCB, etc):**

Processes will be contained in a class with attributes that hold all the data stored in a process control block and the process table will be a map from IDs to PCBs

**Description of multicore process distribution approach:**

1. Load Sharing: Ready processes will be stored in a global queue and a processor will grab the next available ready process upon completion of a process.

**Description of Uniprocessor multiprogramming scheduling:**

1. Round Robin: The selection function is constant, where whichever process is first in the list will be executed. The selection function is used once every time quantum, which is noted by the preemptive decision mode. Throughput (processor utilization) can be low if time quantum is low, however RR allows short response times for short processes. There is not much overhead, just for clock interrupts and for performing scheduling and dispatching. Each process is treated fairly by RR, because they are each given the same amount of time to execute. Starvation is not possible with RR.
2. First Come First Served: The process that has been in the ready queue the longest will be chosen next and executed completely. The decision mode for FCFS is non-preemptive, where the process running continues to run until it terminates or until it blocks (eg. To do a IO call). If the variance in process execution times is high, the response time is high. Additionally, similar to Round Robin, there is minimal overhead, which is a plus for this method. Unfortunately, FCFS is not advantageous for shorter processes and I/O bound processes. Starvation is not possible.
3. Shortest Process Next: Shorter processes skip to the front of the queue. The decision mode is also non-preemptive, like FCFS. For SPN, throughput is high on the CPU. Response time for short processes is good. The overhead on the CPU can be high, though, but longer processes are penalized. Additionally, starvation is possible with SPN, which is not ideal.

**Diagram of multilevel queue configuration & scheduling algorithms:**

**Queues:**

New: Processes being made

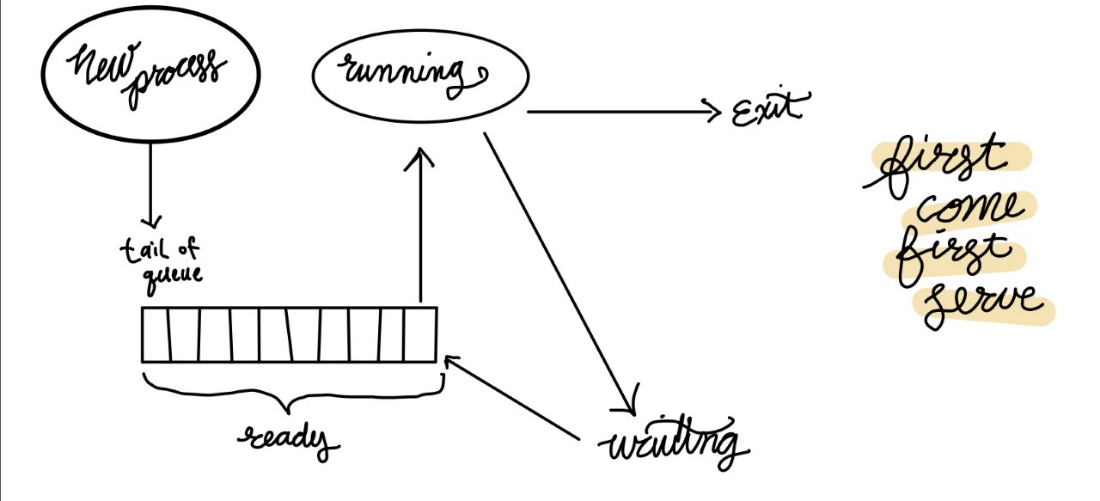
Ready: Processes ready to happen

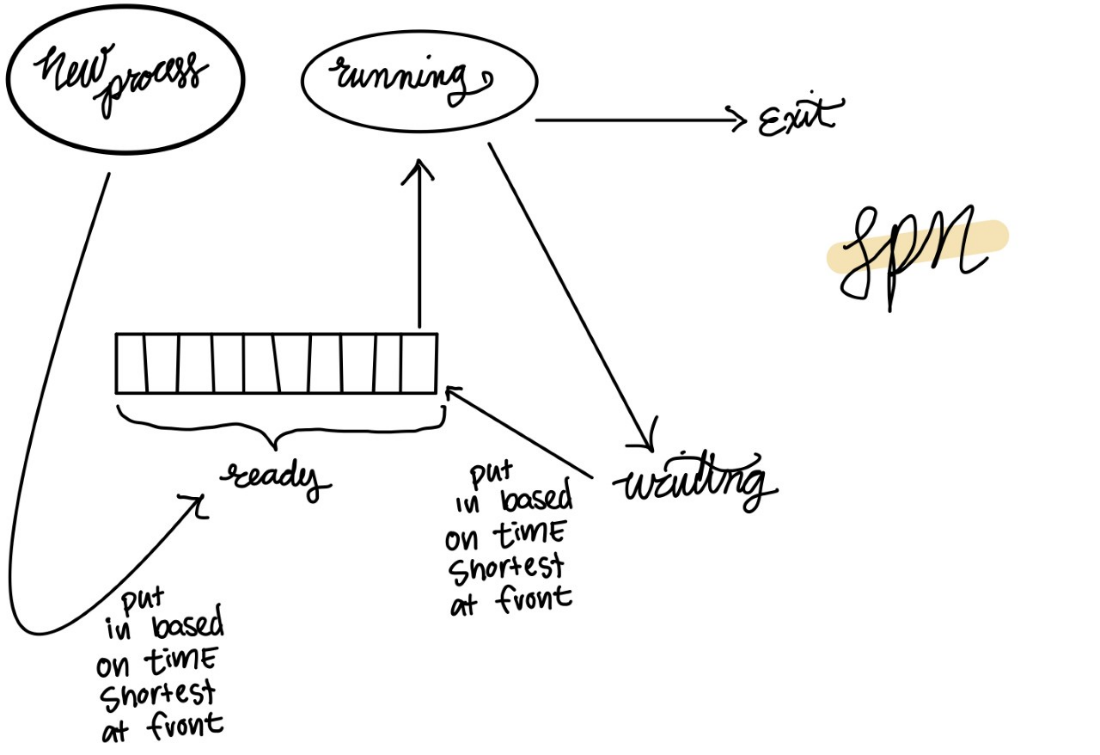
Blocked: Processes waiting on I/O -> I/O will be executing while they are in waiting queue

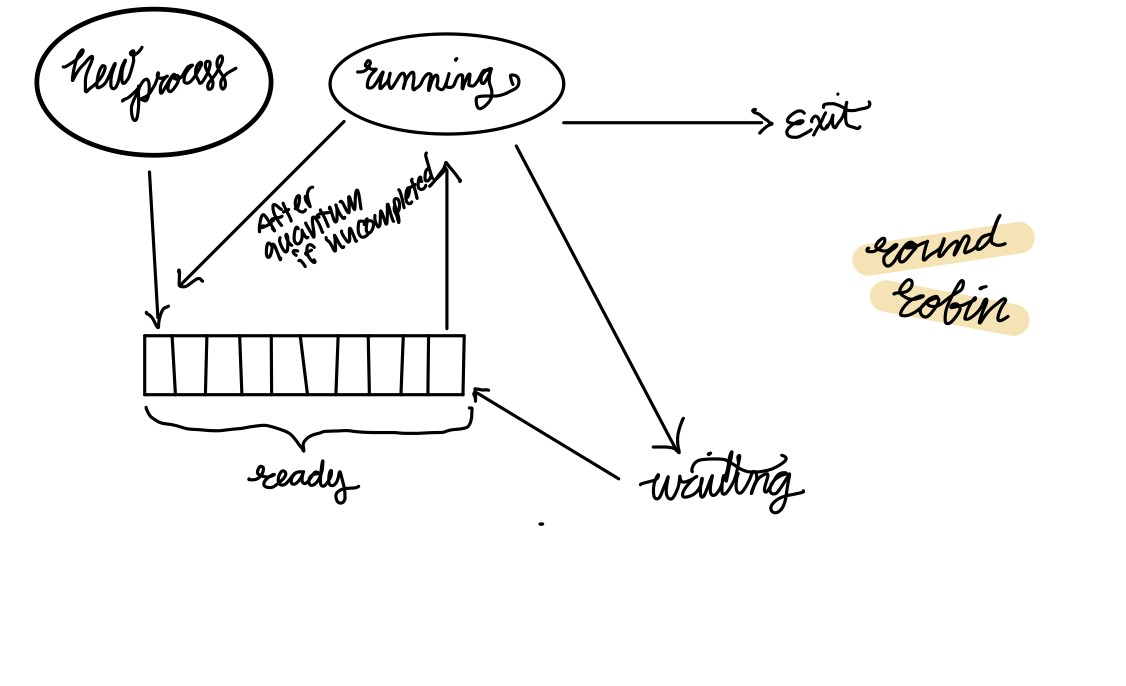
Running: Processes being executed by a processor

Exit: After all execution is completed

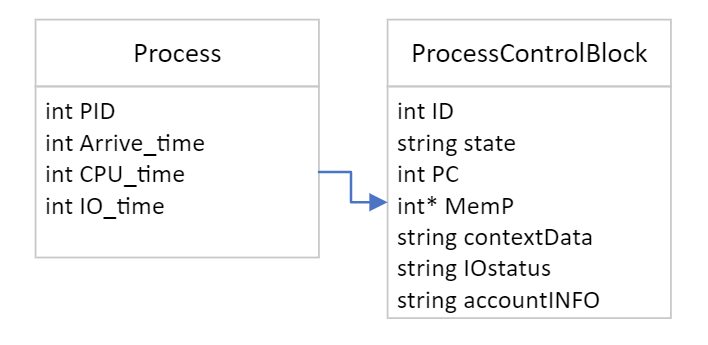
**Algorithms**







**UML Diagrams :**



**Project Management Plan:**

**Deliverables each team member is responsible for?** We are all responsible for them all.

**Milestone deliverable dates:** By Thursday having code done. Thursday and Friday will be for graphs and analysis