README

System Calls

watitx

1. The two arguments are pointers to integers to which waitx will assign the total number of clock ticks during which process was waiting and total number of clock ticks when the process was running.
2. The return values for waitx should be same as that of wait system-call.
3. rtime,etime,ctime,iotime are added to the struct proc and updated at appropriate positions

ps

1. Prints process information of all the active processes.
2. cur_q,n_run etc. are added to the proc structure.

Schedulling Algorithms

FCFS

1. Set SCHEDULER=FCFS

2. This is a non premptive policy.

3. It selects process that was created earliest to run

PBS

1. Set SCHEDULER=PBS

2. It selects the process with the highest priority for execution. In case two or more processes have the same priority, we choose them in a round-robin fashion.

3. The priority of a process can be in the range [0,100], the smaller value will represent higher priority. The default priority of a process is 60.

4. The new system call 'set_priority' can change the priority of a process. The system-call returns the old-priority value of the process.

MLFQ

1. Set SCHEDULER=MLFQ

2. MLFQ scheduler allows processes to move between different priority queues based on their behavior and CPU bursts. If a process uses too much CPU time, it is pushed to a lower priority queue, leaving I/O bound and interactive processes for higher priority queues. Also, to prevent starvation, it implements aging.

3. The struct proc stores the required details.

4. First aging is implemented by checking which process is exceeding waittime. This is done by comparing the ticks with the time since which it was created. If exceeded pushed up.

5. Chooses the process in highest queue that is runnable,Then checks if it has exceeded the time slice.