Report

Anshul Padhi, 2018114013

To Do:
1.waitx(int *wtime, int *rtime)-A syscall similar to wait(). waitx gives wtime and rtime pointer which signify waiting and runtime to a zombie child.
2.getpinfo(struct procstat *p)-Assigns values to the fields of the struct procstat for which gives details on the current user process.
3.FCFS scheduling-Selects the process with lowest creation time. The selected process runs non-preemptively.
4.PBS scheduling-Selects the process with highest priority. Lower the priority value, higher the priority. The processes with same priority are scheduler in RR format.
5.MLFQ scheduling-Has 5 queues, q1, q2, q3 ,q4, q5 with q1 given highest priority and q5 the lowest. Each queue has a maximum cpu ticks. It has been set as {1, 2, 4, 8, 16}. Processes are initially added to the first queue. Once they use up their allotted time they shift to a lower priority queue. If a process has been waiting too long it moves up to a higher priority queue.

To test the scheduling algorithms I made a program which forks a number of times and in each fork runs some computations in a loop.
On testing it was found that MLFQ gave the best time of about 5k ticks, whereas PBS ran in 6k ticks and FCFS ran in about 7k ticks.