CAIRO UNIVERSITY

FACULTY OF ENGINEERING

CREDIT HOURS SYSTEM

os project phase1

Team number: 4

Team members:

1. Ziad amr salah 1190103
2. Nada tarek 1180504
3. Ahmed sameh 1162351
4. Mariam essam 1180027

Workload distribution:

1. Ziad amr: scheduler (sjf, hpf, RR, mlfl), process.c, debugging.
2. Nada tarek: process generator, process.c, debugging.
3. Ahmed sameh: process generator.
4. Mariam essam: scheduler(sjf,hpf).

Assumptions:

1. In RR and mlfl we don’t need to wait till .next quantum to schedule a process if nothing is running
2. In mlfl we itreate over multi level queue levels even when nothing is running and the process starts to get ran from whichever queue level the scheduler was at the time.

Discussion of the data structures used in the project:

1. Priority queue made with a linked list for both hpf, sjf
2. Circular linked list for round robin
3. Chained linked list for mlfl, chained linked list consists of a pointer to linked list and a pointer to next chaine

Your algorithm explanation:

1. Sjf: use a priority queue with prio as runtime, peek at the first process in queue, pop it and run it and repeat.
2. Hpf: use a priority queue with prio as priority, peek at the first process in queue, pop it and run it. While running check head of queue to see if prio is higher if so switch and repeat this.
3. RR: insert process at tail of a circular linked list, run head of process for a quantum (or remaining time if less than the default quantum), if a process is finished remove it from list and then advance head.
4. Mlfl: start at the head of the chained linkedlist iterate and grab first available process, run it for a quantum (or remaining time if less than the default quantum), when its finished grab next available one, repeat until null is left(ran all process for that level). Increament level (with looping if reached min prio level) and repeat.

Data structures used:

1. Priority queue: with queue priority as runtime for sjf and queue priority as priority level for hpf
2. Circular linkedlist: for round robin.
3. Used priority queue as a regular linked list (all queue priorities =0) for mlfl.

Results:

Testcase:

#id arrival runtime priority

1 1 6 5

2 2 10 0

3 19 14 6

4 20 7 4

Sjf:

#At time x process y state arr w total z remain y wait k

At time 1 process 1 started arr 1 total 6 remain 6 wait 0

At time 7 process 1 finished arr 1 total 6 remain 0 wait 0 TA 6 WTA 1.00

At time 7 process 2 started arr 2 total 10 remain 10 wait 5

At time 17 process 2 finished arr 2 total 10 remain 0 wait 5 TA 15 WTA 1.50

At time 19 process 3 started arr 19 total 14 remain 14 wait 0

At time 33 process 3 finished arr 19 total 14 remain 0 wait 0 TA 14 WTA 1.00

At time 33 process 4 started arr 20 total 7 remain 7 wait 13

At time 40 process 4 finished arr 20 total 7 remain 0 wait 13 TA 20 WTA 2.86

CPU utilization= 92.50 % Avg

WTA =1.59

Avg Waiting =4.50

Hpf:

#At time x process y state arr w total z remain y wait k

At time 1 process 1 started arr 1 total 6 remain 6 wait 0

At time 2 process 1 stopped arr 1 total 6 remain 5 wait 0

At time 2 process 2 started arr 2 total 10 remain 10 wait 0

At time 12 process 2 finished arr 2 total 10 remain 0 wait 0 TA 10 WTA 1.00

At time 12 process 1 resumed arr 1 total 6 remain 5 wait 10

At time 17 process 1 finished arr 1 total 6 remain 0 wait 10 TA 16 WTA 2.67

At time 19 process 3 started arr 19 total 14 remain 14 wait 0

At time 20 process 3 stopped arr 19 total 14 remain 13 wait 0

At time 20 process 4 started arr 20 total 7 remain 7 wait 0

At time 27 process 4 finished arr 20 total 7 remain 0 wait 0 TA 7 WTA 1.00

At time 27 process 3 resumed arr 19 total 14 remain 13 wait 7

At time 40 process 3 finished arr 19 total 14 remain 0 wait 7 TA 21 WTA 1.50

CPU utilization= 92.50 % Avg

WTA =1.54

Avg Waiting =4.25

RR: (for a quantum of 3)

#At time x process y state arr w total z remain y wait k

At time 1 process 1 started arr 1 total 6 remain 6 wait 0

At time 4 process 1 stopped arr 1 total 6 remain 3 wait 0

At time 4 process 2 started arr 2 total 10 remain 10 wait 2

At time 7 process 2 stopped arr 2 total 10 remain 7 wait 2

At time 7 process 1 resumed arr 1 total 6 remain 3 wait 3

At time 10 process 1 finished arr 1 total 6 remain 0 wait 3 TA 9 WTA 1.50

At time 10 process 2 resumed arr 2 total 10 remain 7 wait 5

At time 13 process 2 stopped arr 2 total 10 remain 4 wait 5

At time 13 process 2 resumed arr 2 total 10 remain 4 wait 5

At time 16 process 2 stopped arr 2 total 10 remain 1 wait 5

At time 16 process 2 resumed arr 2 total 10 remain 1 wait 5

At time 17 process 2 finished arr 2 total 10 remain 0 wait 5 TA 15 WTA 1.50

At time 19 process 3 started arr 19 total 14 remain 14 wait 0

At time 22 process 3 stopped arr 19 total 14 remain 11 wait 0

At time 22 process 4 started arr 20 total 7 remain 7 wait 2

At time 25 process 4 stopped arr 20 total 7 remain 4 wait 2

At time 25 process 3 resumed arr 19 total 14 remain 11 wait 3

At time 28 process 3 stopped arr 19 total 14 remain 8 wait 3

At time 28 process 4 resumed arr 20 total 7 remain 4 wait 5

At time 31 process 4 stopped arr 20 total 7 remain 1 wait 5

At time 31 process 3 resumed arr 19 total 14 remain 8 wait 6

At time 34 process 3 stopped arr 19 total 14 remain 5 wait 6

At time 34 process 4 resumed arr 20 total 7 remain 1 wait 8

At time 35 process 4 finished arr 20 total 7 remain 0 wait 8 TA 15 WTA 2.14

At time 35 process 3 resumed arr 19 total 14 remain 5 wait 7

At time 38 process 3 stopped arr 19 total 14 remain 2 wait 7

At time 38 process 3 resumed arr 19 total 14 remain 2 wait 7

At time 40 process 3 finished arr 19 total 14 remain 0 wait 7 TA 21 WTA 1.50

CPU utilization= 92.50 % Avg

WTA =1.66

Avg Waiting =5.75