CS4328 & CS5305: Homework #2

Due on Feb 23 11:55 PM)

Aaron Luna

PLEASE READ: You may discuss this problem set with other students. However, you must *write up your answers on your own*. You must also write the names of other students you discussed any problem with. Each problem has a different weight. Please state any assumptions you are making in solving a given problem. *Show your work*. Late assignments will not be accepted with prior arrangements. By submitting this assignment, you acknowledge that you have read the course syllabus.

Problem 1

Consider the following processes:

Process	Arrival Time	Processing Time
A	0	5
B	1	4
C	3	3
D	4	2
E	8	3

Show how the above processes execute over time on a single CPU system. Compute the completion time for each process, the average turnaround time and the average normalized turnaround time for all processes under each of the following schedulers:

(a) FCFS. **[5 pts]**

Non-preemptive, very simple, process execution is first come first serve, processes run until completion, long average waiting times, and starvation.

Turnaround = completion time − arrival time

avg turnaround = total turnaround time / # of processes

Process	completion time	turnaround time
A	5	5
B	9	8
C	12	9
D	14	10
E	17	9

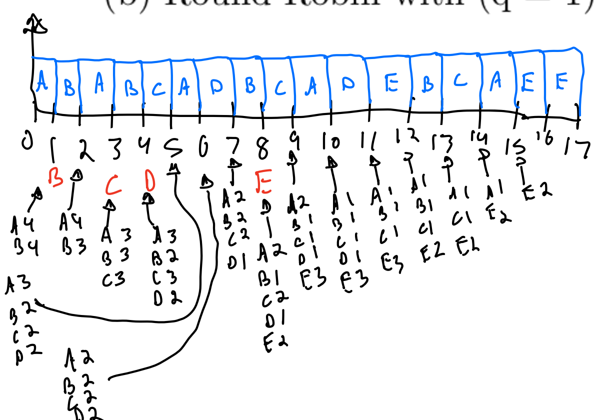
avg $= (5+8+9+10+9)/5$

$= 8.2$

avg norm turnaround =

$\frac{\frac{5}{5}+\frac{8}{4}+\frac{9}{8}+\frac{10}{2}+\frac{9}{3}}{5} = 2.8$

(b) Round Robin with (q = 1). **[5 pts]**



Process	comp time	turn around
A	15	15
B	13	12
C	14	11
D	11	7
E	17	9

avg $= \frac{15+12+11+7+9}{5}$

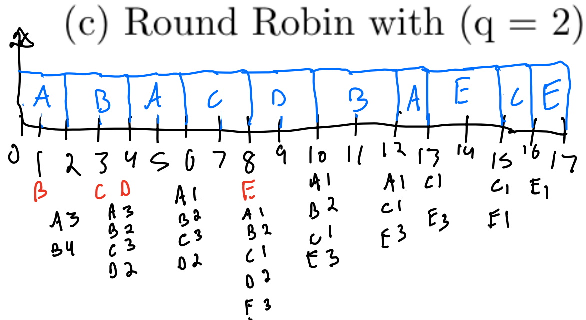
$= 8.8$

avg norm turnaround =

$\frac{\frac{15}{5}+\frac{12}{4}+\frac{11}{3}+\frac{7}{2}+\frac{4}{3}}{5} = 3.2$

For both B & C: Round robing is time quantum, preemptive based on clock, processes get equal execution time, based on time slice given as a.

(c) Round Robin with (q = 2). **[5 pts]**



Process	completion time	turn around
A	13	13
B	12	11
C	16	13
D	10	6
E	17	9

avg $= \frac{13+11+13+6+9}{5}$

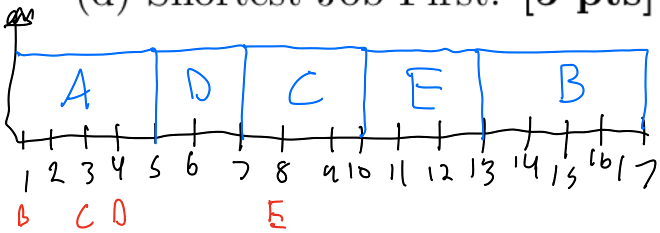
$= 10.4$

avg norm turnaround =

$\frac{\frac{13}{3}+\frac{11}{4}+\frac{13}{3}+\frac{6}{2}+\frac{9}{3}}{5} = 3.14$

(d) Shortest Job First. [5 pts]

Non-preemptive, assigns CPU to process that is shortest, shortest job among available jobs, and starvation can occur.



Turn around time = completion time − arrival time

avg turn around time = total turn around time / # of processes

Process	completion time	turn around
A	5	5
B	17	16
C	10	7
D	7	3
E	13	5

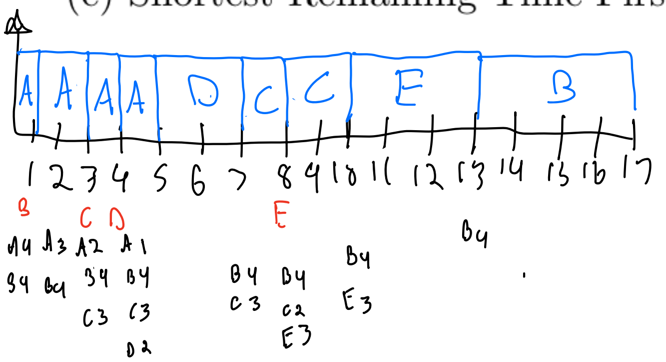
$$\text{avg} = \frac{5+16+7+3+5}{5} = 7.2$$

avg norm turnaround =

$$\frac{\frac{5}{5}+\frac{16}{4}+\frac{7}{3}+\frac{3}{2}+\frac{5}{3}}{5} = \underline{2.1}$$

(e) Shortest Remaining Time First. [5 pts]

Preemptive, scheduler checks every tick to choose shortest job available, and better turnaround time than SJF.



Process	completion time	turn around
A	5	5
B	17	16
C	10	7
D	7	3
E	13	5

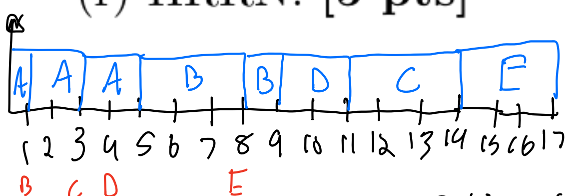
$$\text{avg} = \frac{5+16+7+3+5}{5} = 7.2$$

avg norm turnaround =

$$\frac{\frac{5}{5}+\frac{16}{4}+\frac{7}{3}+\frac{3}{2}+\frac{5}{3}}{5} = \underline{2.1}$$

(f) HRRN. [5 pts]

Highest Response Ratio Next, non-preemptive, no starvation, favors process with short service times, and limits waiting time for longer jobs.



@ time 1, A & B here RR?

A: (1+5)/5 = 1.2 ✓

B: (0+4)/4 = 1

@ time 3, B & C here, RR?

B: (2+4)/4 = 1.5

C: (0+3)/3 = 1

A: (3+4)/4 = 1.75 ✓

@ time 5, B & C & D here RR?

B: (4+4)/4 = 2 ✓

C: (2+3)/3 = 1.67

D: (1+2)/2 = 1.5

@ time 8, B & C & D & E RR?

B: (7+1)/1 = 8 ✓

C: (5+3)/3 = 2.67

D: (4+2)/2 = 3

E: (0+3)/3 = 1

@ time 9, D & C & E

C: (6+3)/3 = 3

D: (5+2)/2 = 3.5 ✓

E: (1+3)/3 = 1.33

@ time 11, C & E

C: (8+3)/3 = 3.67 ✓

E: (3+3)/3 = 2

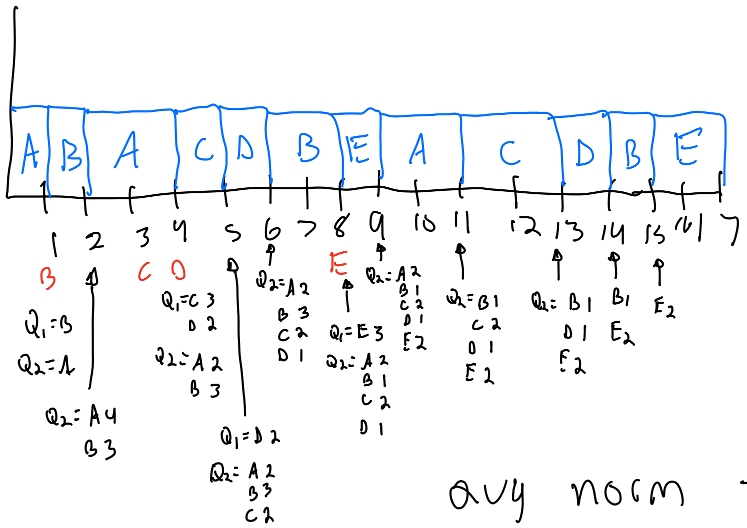
Process	Comp time	turn around
A	5	5
B	9	8
C	14	11
D	11	7
E	17	9

$$\text{avg} = \frac{5+8+11+7+9}{5} = 8$$

avg norm turn around =

$$\frac{\frac{5}{5}+\frac{8}{4}+\frac{11}{3}+\frac{7}{2}+\frac{9}{3}}{5} = \underline{2.63}$$

(g) Multi-level Feedback with 2 queues. Queue serves 1 quantum (unit of time) at a time while queue 2 serves 2 quantums at a time. All processes get serviced from queue 1 initially in FCFS fashion and if they do not complete, they move to queue 2. Queue 2 runs a round robin scheduler. Assume that Queue 1 has a higher priority than Queue 2, and assume that a process arriving to queue 1 cannot preempt an already running process from queue 2 *within* its 2 quantum. **[5 pts]**



Process	Comp time	Turn around
A	11	11
B	15	14
C	13	10
D	14	10
E	17	9

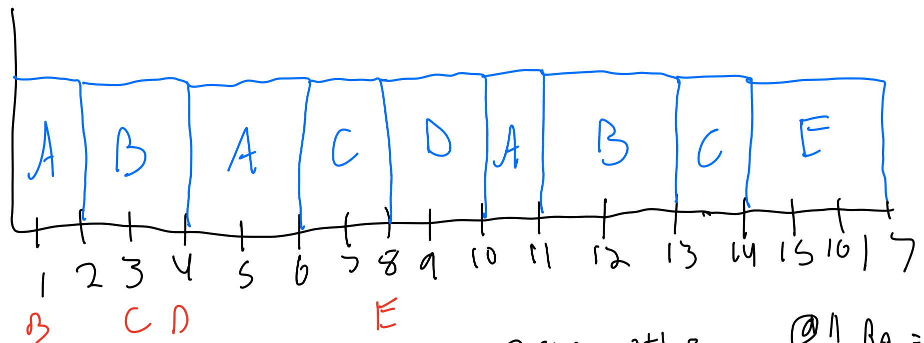
avg = $\frac{11+14+10+10+9}{5}$ = 14

avg norm turnaround = $\frac{11}{5}+\frac{14}{4}+\frac{10}{3}+\frac{10}{2}+\frac{9}{3}$ /5

= 3.41

(h) PHRRN is a new scheduler proposed by some students based on their discussions of the regular HRRN. PHRRN stands for preemptive highest response ratio next. In PHRRN, the scheduler runs periodically (in addition to when a process completes) and the ratios are computed at that time to make scheduling decisions. Show the execution of the above processes when the scheduler is invoked every 2 time units. **[5 pts]**

Response ration = w+s / s w = wait time s = service time



Process	Comp time	turn around
A	11	11
B	13	12
C	14	11
D	16	6
E	17	9

avg = 9.8

avg norm turnaround = 2.97

@1 $R_a = \frac{0+4}{4} = 1$

@2 $R_a = \frac{0+3}{3} = 1$
$R_B = \frac{1+4}{4} = \frac{5}{4}$

@3 $R_a = \frac{1+4}{4} = \frac{5}{4}$
$R_B = \frac{0+1}{1} = 1$
$R_C = \frac{1+3}{3} = \frac{4}{3}$

@4 $R_a = \frac{2+3}{3} = 5/3$
$R_b = 0+2/2 = 1$
$R_c = 1+3/3 = 4/3$
$R_D = 0+2/2 = 1$

@5 $R_a = \frac{1+1}{1} = 2$
$R_B = \frac{1+2}{2} = 3/2$
$R_C = \frac{2+3}{3} = 5/3$
$R_D = \frac{1+2}{2} = 3/2$

@6 $R_a = \frac{0+1}{1} = 1$
$R_B = \frac{2+2}{2} = 2$
$R_C = \frac{3+3}{3} = 2$
$R_D = \frac{2+2}{2} = 2$

@7 $R_a = \frac{1+1}{1} = 2$
$R_b = \frac{3+2}{2} = 5/2$
$R_c = \frac{1+1}{1} = 2$
$R_D = \frac{3+2}{2} = 5/2$

@8 $R_a = \frac{2+1}{1} = 3$
$R_b = \frac{4+2}{2} = 3$
$R_c = \frac{0+1}{1} = 1$
$R_D = \frac{4+2}{2} = 3$
$R_E = \frac{0+3}{3} = 1$

@10 $R_a = \frac{4+1}{1} = 5$
$R_b = \frac{6+2}{2} = 3$
$R_c = \frac{2+1}{1} = 3$
$R_E = \frac{2+3}{3} = 5/3$

@11 $R_B = \frac{7+2}{2} = 9/2$
$R_C = \frac{3+1}{1} = 4$
$R_E = \frac{3+3}{3} = 2$

@12 $R_B = \frac{8+2}{2} = 5$
$R_C = \frac{4+1}{1} = 5$
$R_E = \frac{4+3}{3} = 7/3$

@13 $R_C = \frac{5+1}{1} = 6$
$R_E = \frac{5+3}{3} = 8/3$

@14 $R_E = \frac{6+3}{3} = 3$

Problem 2

Prove that the Shortest Job First scheduling algorithm achieves the minimum average waiting time for a group of processes that arrived at the same time.**[5 pts]**

Shortest job First = minimum avg wait time for group of processes is achieved and executed. Assume N number of jobs arrive at same time, $x_1, x_2, x_3, x_4, \dots, x_n$; Proving for minimum avg wait time would be saying $x_1 \leq x_2 \leq x_3 \leq x_4 \leq \dots \leq x_n$, in increasing order. From this we get $A_{total} = n x_1 + (n-1)x_2 + (n-2)x_3 + (n-3)x_4 + \dots + x_n \;/\; n$

from here let y & z be jobs where y is before z.
this becomes $A_{new} = (y-z)(x_y - x_z) / n$. Combining both equations we get, $AA = A_{tot} + A_{new}$ where $AA \geq A_{tot}$.

In contradiction proof, if there is one algorithm that is better than SJF it would select longer job first. This algorithm would mean there exists at least one process, with a shorter execution time that experiences more waiting time than SJF

Now lets say there is one that selects a shorter job first, its the same as SJF above.

Conclusion, a contradiction is reached in both scenarios so therefore, there can't be a scheduling algorithm that achieves a lower waiting time than SJF if processes arrive at same time.

Problem 3

Least Slack Process Next (LSPN) is a real-time scheduler for periodic tasks. Slack is the amount of time between when a task would complete if it started now and its next deadline. Thus it can be expressed as:

$$Slack \quad = \quad D - t - C$$

where D is the deadline time, t is the current time and C is the processor time needed. LSPN selects the task with the minimum slack time to execute next. If two tasks have the same slack, they are serviced based on FCFS. Answer the following questions:

(a) What does it mean for a task to have a slack of 0? **[2 pts]**

Slack is amount of time a process can be delayed past its earliest start or earliest finish without delay. Can be calculated by difference in latest start time from earliest start time. $(D-t)-C$

Slack=0 then CPU will start immediately.

(b) What does it mean for a task to have a negative slack? **[2 pts]**

It means a task cannot be completed by wanted deadline. It means it is already behind schedule. Example- if a task has negative slack of 2 hours then it must have been started 2 hours before scheduled start of project. Making it impossible for CPU to have enough time to finish.

(c) How long may the scheduler delay starting a task (and still meet its deadline), if that task has a slack s? **[2 pts]**

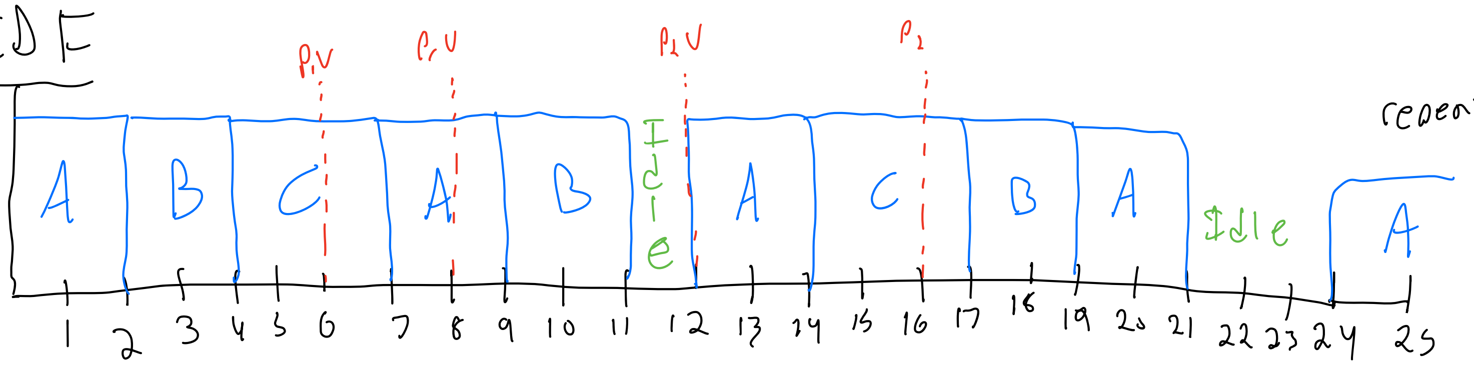
Delay S time units

Scheduler can delay a task up to the slack #.

(d) Consider 3 period tasks: A, B and C. Task A has a period 6 and execution time 2, task B has a period of 8 and execution time of 2 and task C has a period of 12 and execution time of 3. Illustrate (by drawing the executions of A, B and C over time) how LSPN would schedule these tasks in comparison to Earliest Deadline First and Rate Monotonic Scheduling. **[9 pts]**

Task	P	exc
A	6	2
B	8	2
C	12	3

EDF



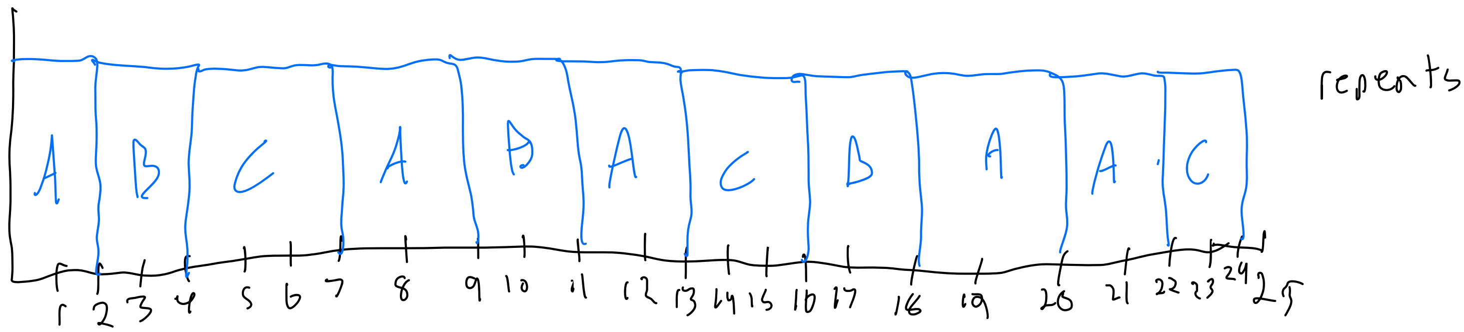
period: 12

t = 11, C executed

4 ms preemption total

Repeats after 24 ms

LSPN



@0 $S_A = 6-0-2=4$
$S_B = 8-0-2=6$
$S_C = 12-0-3=9$

@2 $S_A = 12-2-2=8$
$S_B = 8-2-2=4$
$S_C = 12-2-3=7$

@4 $S_A = 12-4-2=6$
$S_B = 16-4-2=10$
$S_C = 12-4-3=5$

@7 $S_A = 12-5-2=7$
$S_B = 16-7-2=7$
$S_C = 24-7-3=4$

@9 $S_A = 13-9-2=4$
$S_B = 16-9-2=5$
$S_C = 24-10-2=12$

@11 $S_A = 18-9-2=7$
$S_B = 16-9-2=5$
$S_C = 24-8-4=10$

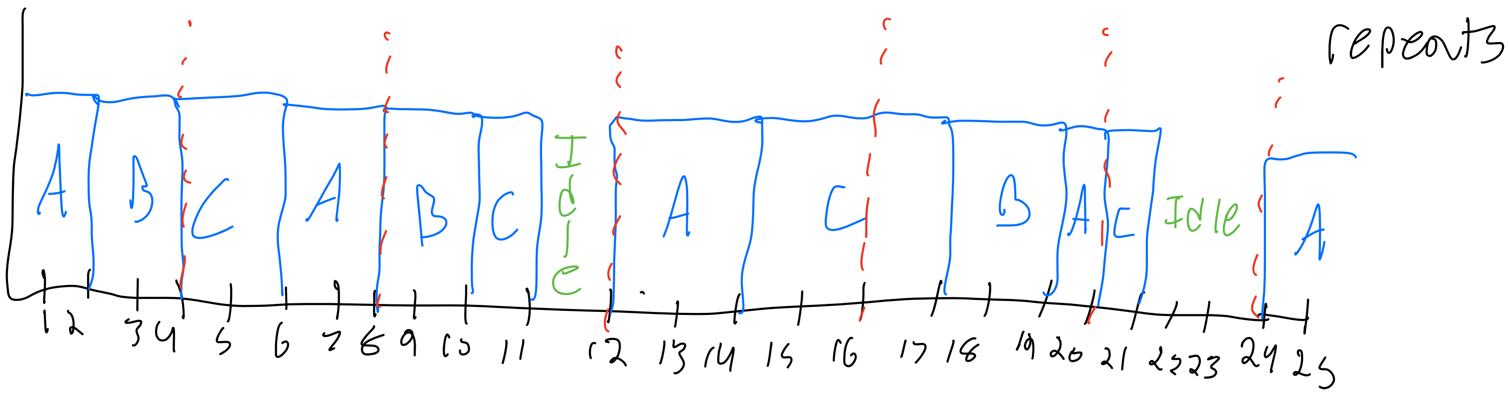
@13 $S_A = 24-16-2=4$
$S_B = 24-13-2=9$
$S_C = 24-13-3=8$

@16 $S_A = 24-16-2=6$
$S_B = 24-8-8=8$
$S_C = 36-16-3=17$

@18 $S_A = 24-18-2=4$
$S_B = 16-8-2=6$
$S_C = 36-14-8=12$

@20 $S_A = 30-20-2=8$
$S_B = 56-20-2=14$
$S_C = 56-32-9=13$

RMS



A periods are less than B so priority becomes

$A > B > C$