CS 4328: Homework #2

Due on Feb, 14, 2019

Jamal Rasool

In Collaboration With:

Problem 1

Consider a disk subjected to I/O requests arriving at an average rate of λ requests/sec. It was observed that on average, there was about 5 requests waiting for service and that each request waits for 10 msec, on average. What is the effective arrival rate λ for this disk that led to those numbers? [4 pts]

Problem 1 :

I/o Request / sec = λ

5 request waiting for service

10 msec → .01 seconds

Determine effective Arrival

W = n • Lambda

$5 = .01 \cdot \lambda$

$\frac{5}{.01} = \lambda$

= 500 processes / sec

Problem 2

Assume processes arrive to the CPU based on a Poisson distribution with an arrival rate λ of 5 processes per second. Assume also that the service times are exponential. Given that the CPU is, on average, busy 70% of the time. What is the average turnaround time of these processes? How many processes you expect to see waiting, on average, in the Ready Queue? [4 pts]

Problem 2:

λ = 5 process/second , exponenticl service times

CPU Avg 70%.

What is turn orround rate / How many processes.

Q = Rho / 1 - Rho $\qquad$ $\lambda \cdot Tq = q$

Q = 2.333 $\qquad$ $5 \cdot Tq = 2.33$

Tq = 2.33 / 5 = .466 = Turn orround time.

Q = w + Rho

W = 2.33 - .70 = 1.63 is the amount of Avg processes waiting in the ready queue.

Problem 3

A system is composed of CPU, Disk and Network. The execution of a process proceeds as follows:

1. The process uses the CPU and then with probability 0.2 proceeds to step 2, with probability 0.3 proceeds to step 3, and with probability 0.5 proceeds to step 4.
2. The process performs Disk I/O and then with probability 0.3 proceeds to step 1, and with probability 0.7 proceeds to step 4.
3. The process performs Network I/O and then with probability 0.4 proceeds to Step 1 and with probability 0.6 proceeds to step 4.
4. The process is leaves the system (perhaps due to an I/O error).

The following information is known about this system:

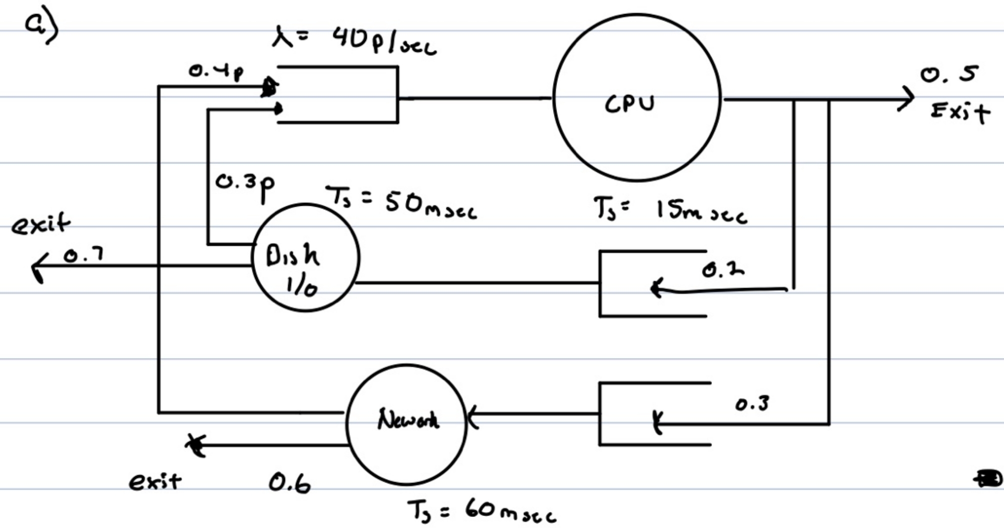
• Processes arrive according to a Poisson process with an average rate of 40 processes per second. • The service time of the CPU is exponentially distributed with an average of 15 msec.
• The service time of the Disk is exponentially distributed with an average of 50 msec.
• The service time of the Network is exponentially distributed with an average of 60 msec.

• All buffers are of infinite sizes.
Answer the following questions [3 points each]:

(a) Draw a queuing diagram depicting the system above

Problem 3:

a)



(b) Which resource is the bottleneck resource?

b) $\lambda_c = 40 + (\lambda_D \cdot 0.3) + (\lambda_N \cdot 0.4)$

$\lambda_D = 0.2\lambda_c$

$\lambda_N = 0.3 \cdot \lambda_c$

$\lambda = 40$ p/sec

$T_{s_{CPU}} = .015$ seconds

$T_{s_{Disk}} = 0.05$ seconds

$T_{s_{Network}} = 0.06$ seconds

$\lambda_c = 40 + 0.3(0.2\overset{=x}{\lambda_c}) + 0.3(0.4\overset{=x}{\lambda_c})$

$\lambda_c = 40 + .06x + .12x$

$\left.\begin{array}{l} x = 40 + .18x \\ .82x = 40 \end{array}\right\} -.18x$

$\lambda_c = 48.78$

- $\lambda_c = 48.78$
- $\lambda_D = 9.76$
- $\lambda_N = 14.63$

$\rho_{CPU} = 48.78 \cdot 0.15s = .7317$

$\rho_{Disk} = 9.76 \cdot 0.05s = .48$

$\rho_{Network} = 14.63 \cdot .06s = \boxed{.87 \quad \therefore \text{ Network is bottleneck}}$

(c) What is the average total turnaround time for a process submitted to the above system?

c) $q_{CPU} = \frac{.7317}{1-.7317}$ $q_{Disk} = \frac{.48}{1-.48}$ $q_{Network} = \frac{.87}{1-.87}$

$q_{CPU} = 2.7s$ $q_{Disk} = 0.92s$ $q_{Net} = 6.69s$ $q_{total} = 10.31$

for turn around

$T_{average} = \frac{q_{Total}}{\lambda_{total}} = \frac{10.31}{40} = .25$

(d) What is the probability that a process would not experience any waiting time once it gets submitted to the CPU?

d) $\rho_{CPU} = 0.7317$

$1 - 0.7317 = 0.268 \rightarrow 26.8\%$

(e) What arrival rate would render this system unstable?

e)

Since network is bottleneck

$\rho_{Network} = 1 \rightarrow \lambda_N = 16.66$

$\lambda_c = 16.66 / .3 = 55.5$

$55.5 = \lambda + 0.3(\lambda_n) + .4(\lambda_n)$

$55.5 = \lambda + 0.3(0.2 \cdot 55.5) + 0.4(0.3 \cdot 55.5)$

$\lambda = 45.5$

Problem 4

Using the rand() function (that returns a random number uniformly distributed between 0 and $RAND_M$ AX), write a simple program the generates the arrival times of 1000 processes (i.e., when each process arrives) that follow a Poisson distribution with an average arrival rate λ of 10 processes per second. Submit the arrival times of the first 10 processes and the actual average arrival rate over the 1000 processes. [Hint 1: A Poisson arrivals means Exponential inter-arrival times. Hint 2: Use the CDF of Exponential Distribution.] [10 pts]

Function to calculate the values: (sizeS = 10, sizeL = 1000)

```
def mathHandler(random_number, lambda_value):
    return -math.log(1-random_number)/lambda_value

# Run the 10 cases
for i in range(0,sizeS):
    rNumber = float(random.randint(0,sys.maxsize))
    rNumber = rNumber/sys.maxsize
    smallerValArr.append(mathHandler(rNumber,10))

# Running 1000 cases then taking the average.
for i in range(0,sizeL):
    rNumber = float(random.randint(0,sys.maxsize))
    rNumber = rNumber/sys.maxsize
    largerValArr.append(mathHandler(rNumber,10))

resultingavg = (sum(largerValArr)/sizeL) * 100
```

Arrival times of the first 10 processes (Rounded to the thousandths place):

1. 0.3747

2. 0.1267
3. 0.0906
4. 0.0994
5. 0.4481
6. 0.0582
7. 0.0831
8. 0.0367
9. 0.0187
10. 0.0759

Actual Average Arrival Rate over 1000 processes:

- 10.0799168558

This value is near the lambda of 10, therefore is an accurate representation of the average rate.

Problem 5

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

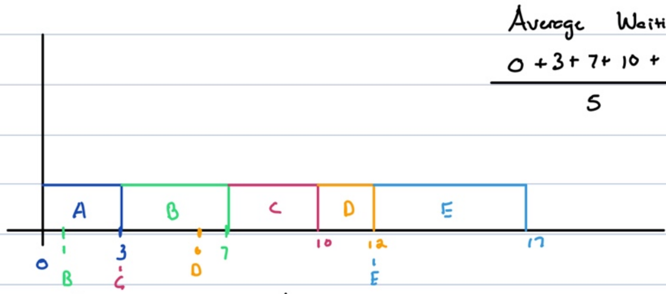
(a) FCFS. [5 pts]

Problem 5 :

a) FCFS

Average Waiting time

$$\frac{0+3+7+10+12}{5} = 6.4$$

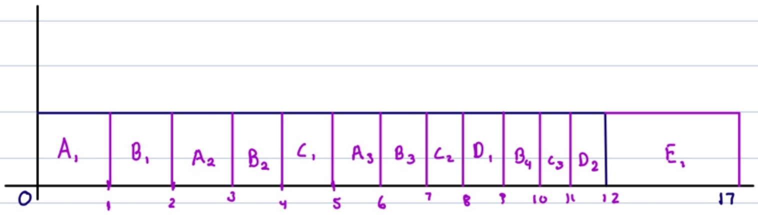


	Arrival time	CPU burst	Completion time	Turn around
A	0	3	3	3
B	1	4	7	6
C	3	3	10	7
D	6	2	12	6
E	12	5	17	5

$$\frac{3+6+7+6+5}{5} = 5.4 = \text{Avg turn around}$$

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

b) Round Robin q=1



	Arrival time	CPU burst	Completion time	Turn around
A	0	3	6	6
B	1	4	10	9
C	3	3	11	8
D	6	2	12	6
E	12	5	17	5

$$\frac{6+9+8+6+5}{5} = 6.8 = \text{Avg turn around}$$

(c) Shortest Job First. [5 pts]

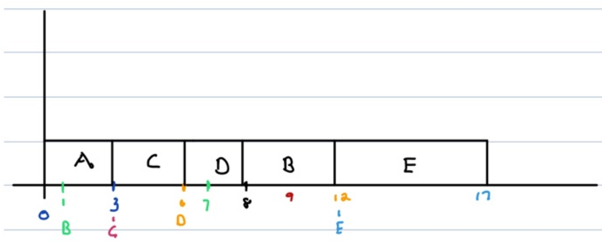
c) Shortest Job First

	Arrival time	CPU burst	Completion time	Turn around
A	0	3	3	3
B	1	4	7	6
C	3	3	12	9
D	6	2	9	3
E	12	5	17	5

$$\frac{3+6+9+3+5}{5} = 5.2 = \text{Avg turn around}$$

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

d) Shortest Remaing time first

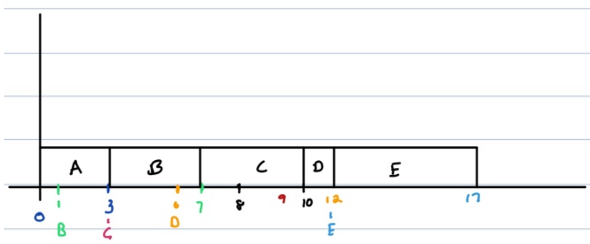


	Arrival time	CPU burst	Completion time	Turn around
A	0	3	3	3
B	1	4	12	11
C	3	3	6	3
D	6	2	8	2
E	12	5	17	5

$\frac{3+11+3+2+5}{5} = 4.8$ = Avg turn arround

(e) HRRN. [5 pts]

e) HRRN

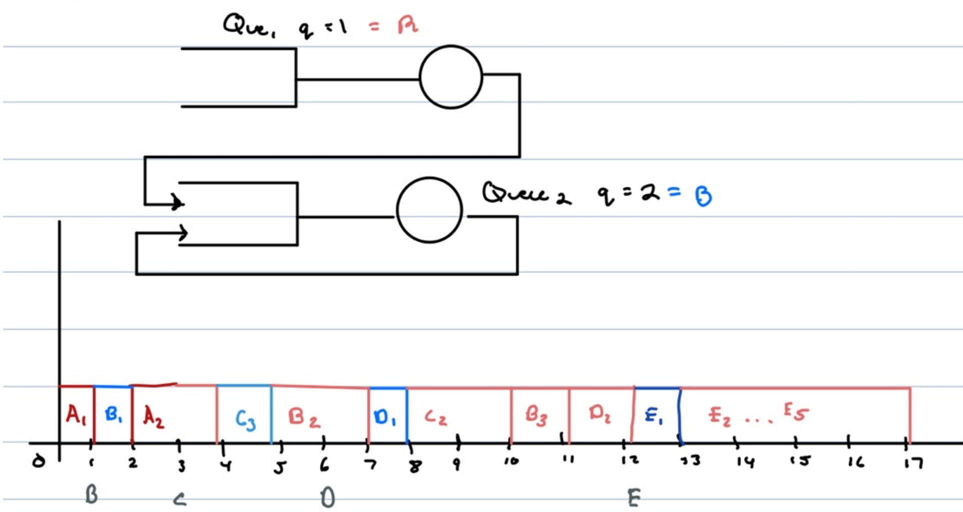


	Arrival time	CPU burst	Completion time	Turn around
A	0	3	3	3
B	1	4	7	6
C	3	3	10	7
D	6	2	12	6
E	12	5	17	5

$\frac{3+6+7+6+5}{5} = 5.4$ = Avg turn arround

(f) Feedback with 2 queues. Queue 1 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 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 quantums. [5 pts]

f)



	Arrival time	CPU burst	Completion time	Turn around
A	0	3	4	4
B	1	4	11	10
C	3	3	10	7
D	6	2	12	6
E	12	5	17	5

$$\frac{4 + 10 + 7 + 6 + 5}{5} = 6.4 = \text{Avg turn around}$$

Problem 6

Prove that the Shortest Process Next (SPN) scheduling algorithm achieves the minimum average waiting time for a bunch of processes that arrived at the same time.[5 pts]

Problem 6:

Proof for SPN :

Asume N number of Jobs , and that all jobs arrive at the same time. ∴ it follows the genral setup, x_1 , x_2 , x_3 ... x_n respectively.

To prove for minimum average waiting time, one must say $x_1 \leq x_2 \leq \ldots x_n$ in terms of increacing order. Since SNP processes shortest available process among all the processes, n jobs will have to wait till the 1st Job is executed. n-1 jobs will have to wait the 2nd job is done and so forth

$$A_{res} = \frac{nx_1 + (n-1)x_2 + \ldots x_n}{n}$$

from this let the jobs P and q where p is less than q are exchanged we find that simplified it becomes

$$I_{res} = \frac{(p-q) \cdot (x_p - x_q)}{n} \geq 0$$

∴ the average response time of new scheduler is $N_{res} = A_{res} + I_{res}$ for $N_{res} \geq A_{res}$ ∴

| the average response time increaces if the jobs exchanged and not in increacing order of service times. Hence why SPN provides the minimum average waiting time for a batch of jobs among non-preemptive scheduling Algorithms

Problem 7

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 = D−t−C (1)

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]

The task must be executed immediately otherwise the task will fail to meet its required deadline. Therefore that means that the process would be placed a the start of the queue in order to meet the required deadline.

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

This means that the way the queue is setup, the negative slack will point to the process not being able to meet the required deadline.

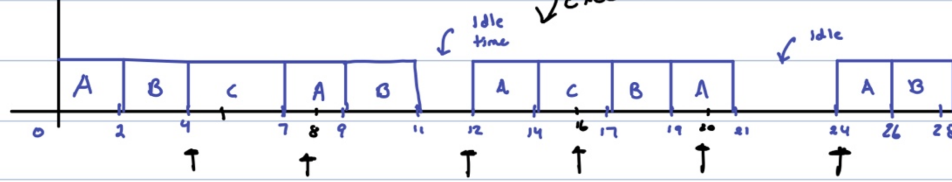
(c) How long may the scheduler delay starting a task (and still meet its deadline), if that task has a slack s? [2 pts]

In this case if the task has a slack of s, then the task or process can possibly be delayed up until an interval of S and still hit that required deadline time. Due to it being able to be scheduled for that specific time.

(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**. Assume that preemption *may* occur at 4-ms intervals for LSPN. [9 pts]

d) Earliest Deadline first (EDF)

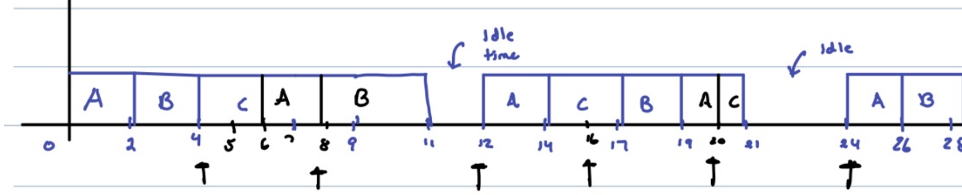
C period = 12
t=11, C has executed 4ms preemption
execute A



Repeats every 24ms

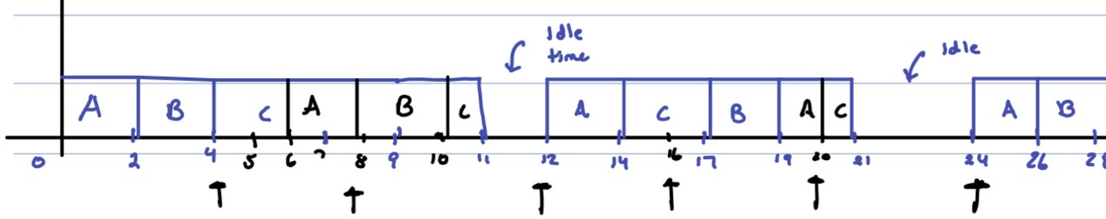
Least Laxity Scheduling

4ms preemption



Rate-monotonic scheduling

4ms preemption



Since A's periods is less than B the priority order becomes A > B > C

Problem 8

Please make a copy of your solutions since you will not receive the graded homework before the midterm. We will discuss the solutions in the review session before the midterm.

Done.