University of Asia Pacific

Admit Card

Final-Term Examination of Spring, 2020

Financial Clearance	PAID

Registration No : 17101007

Student Name : Mahnaz Rafia Islam

Program : Bachelor of Science in Computer Science and Engineering

SI.NO.	COURSE CODE	COURSE TITLE	CR.HR.	EXAM. SCHEDULE
1	CSE 400	Project / Thesis	3.00	
2	CSE 401	Mathematics for computer Science	3.00	
3	CSE 403	Artificial Intelligence and Expert Systems	3.00	
4	CSE 404	Artificial Intelligence and Expert Systems Lab	1.50	
5	CSE 405	Operating Systems	3.00	
6	CSE 406	Operating Systems Lab	1.50	
7	CSE 407	ICTLaw, Policy and Ethics	2.00	
8	CSE 410	Software Development	1.50	
9	CSE 427	Topics of Current Interest	3.00	

Total Credit: 21.50

1. Examinees are not allowed to enter the examination hall after 30 minutes of commencement of examination for mid semester examinations and 60 minutes for semester final examinations.

2. No examinees shall be allowed to submit their answer scripts before 50% of the allocated time of examination has elapsed.

3. No examinees would be allowed to go to washroom within the first 60 minutes of final examinations.

4. No student will be allowed to carry any books, bags, extra paper or cellular phone or objectionable items/incriminating paper in the examination hall.
Violators will be subjects to disciplinary action.

This is a system generated Admit Card. No signature is required.

Admit Card Generation Time: 26-Oct-2020 05:55 PM

University of Asia Pacific
Department of Computer Science and Engineering

Final Term Examination: Spring-2020

Name: Mahnaz Rafia Isam			Registration No: 17101007
Roll No: 07	Year: 4th	Semester: 1st	Course Code: CSE 405
Course Title: Operating Systems		Date: 03.11.2020	

Answer to the question no: 1(a)

My student ID is - 17101007

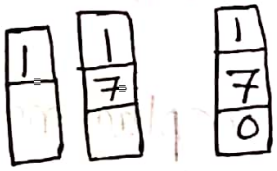
Reverse string of my ID - 70010171

So, Page Request - 17101007 70010171

Given, Window size = 3

Fifo:

1 7 1 0 1 0 0 7 7 0 0 1 0 1 7 1



Page fault : 3

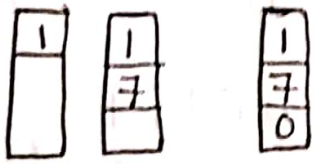
LRU:

1 7 1 0 1 0 0 7 7 0 0 1 0 1 7 1

Page fault : 3

Optimal:

1 7 1 0 1 0 0 7 7 0 0 1 0 1 7 1



Page fault: 3

In this scenario, page fault is same for all the algorithm applied in this scenario. so no comparison can be made for this scenario.

But in general, among these 3 page replacement algorithm Optimal and LRU is better than Fifo and it depends on the given scenario (ie page requests for a system). FIFO is very easy to implement. but not always good at performance. It may replace an active page to bring a new one and thus may cause a page fault of that page immediately, so the page fault rate may increase as the number of allocated

page frame increases. LRU page replacement algorithm is quiet efficient. An optimal page replacement algorithm has the lowest page fault rate of all the three algorithms. It replace the page that will not be used for the longest period of time. So because of lowest page fault occurs in the Optimal page replacement algorithm, it is considered to be the best, but according to page requests for a system LRU can work better also.

Scanned with CamScanner

Answer to the question no: 2(a)

We know, Need = Max − Allocation

	Allocation			Max			Need			Available		
	P	Q	R	P	Q	R	P	Q	R	P	Q	R
P_0	1	2	1	8	6	4	7	4	3	0	0	1
P_1	3	1	1	4	3	3	1	2	2			
P_2	4	1	3	9	1	3	5	0	0			
P_3	3	2	2	3	3	3	0	1	1			
P_4	1	1	3	5	4	4	4	3	1			
	12	7	10									

3 resource type → P (11 instances), Q (6 instances)

R (8 instances).

Last 3 digit of my id → 007

Available = (0 % 3, 10 % 3, 7 % 3)

= 0 0 1

To determine if there is any deadlock present in this scenario, we will perform Banker's Algorithm → Safety Algorithm.

Step-1:

Here,

Work = [0, 0, 1]

	0	1	2	3	4
Finish =	false	false	false	false	false

Step 2: For i = 0, $Need_0 = 7, 4, 3$

Finish[0] is false and $Need_0 > Work$.

$7, 4, 3 > 0, 0, 1$

So, P_0 must wait as it doesn't satisfy the condition $Need_0 \leq Work$.

Step-2: For i = 1, $Need_1 = 1, 2, 2$

Finish[1] is false but $Need_1 > Work$

$1, 2, 2 > 0, 0, 1$

So, P_1 must wait as it doesn't satisfy the condition $Need_1 \leq Work$.

Step-2 For i = 2, $Need_2 = 5,0,0$

Finish[2] is false but $Need_2 > Work$.

$5,0,0 > 0,0,1$.

So, P_2 must wait as it doesn't satisfy the condition $Need_2 \leq Work$.

Step 2: For i = 3, $Need_3 = 0,1,1$

Finish[3] is false but $Need_3 > Work$

$0,1,1 > 0,0,1$

So, P_3 must wait as it doesn't satisfy the condition $Need_3 \leq Work$.

Step-2: For i = 4, $Need_4 = 4,3,1$

Finish[4] is false but $Need_4 > Work$

$4,3,1 > 0,0,1$

So, P_4 must wait as it doesn't satisfy the condition $Need_4 \leq Work$.

Step-4: As Finish[i] = false for $0 \leq i \leq n$

Hence, the system is not in safe state.

Hence, the system has deadlock in it.

So the set of processes is deadlocked as every process P_0, P_1, P_2, P_3, P_4 is in the waiting state.

So, there is deadlock present in this scenario.

Answer to the question no: 3(a)

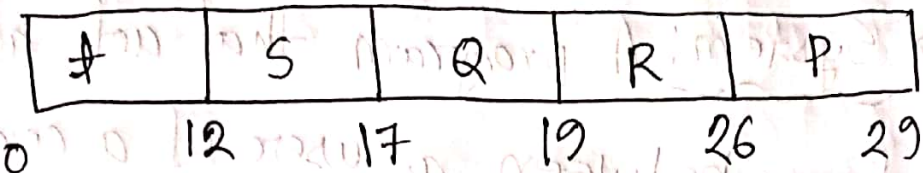
and shortest job first scheduling

Non-preemptive priority scheduling is acceptable for the given scenario.

Non pre-emptive priority scheduling:

Process	Burst time	Priority
P	3	8
Q	2	5
R	7	5
S	5	2
t	12	1

Grant Chart:



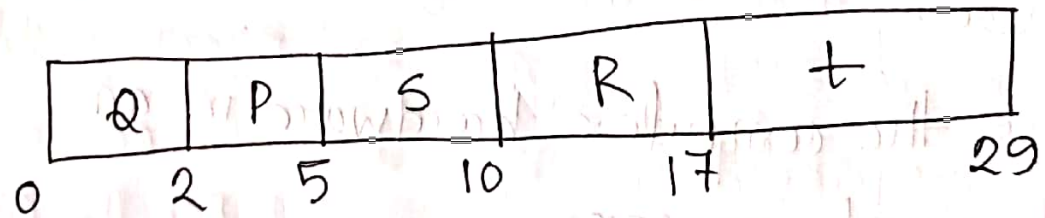
Average waiting time = (0+12+17+19+26)/5

= 74/5

= 14.8 ms

Non-premptive shortest job first:

Grant Chart:



Average waiting time = (0+2+5+10+17)/5

= 6.8 ms.

Answer to the question no: 3(b)

Operating System: A program that acts as an intermediary between a user of a computer and the computer hardware.

Operating system goals-

- Execute user programs and make solving user problems easier.
- Make the computer system convenient to use.
- Use the computer hardware in an efficient manner.

Design goals of an operating system-

- Convenience, abstraction of hardware resources for user program.
- Efficiency of usage of CPU, memory etc.

- Isolation between multiple processes.

Besides all of these goals, operating system does many things such as—

- It manages program memory.
  - Loads program executable (code, data) from disk to memory.
- OS manages CPU
  - Initializes program counter (PC) and other registers to begin execution.
- OS manages external devices:
  - Read/write files from disk.

Answer to the question no: 4(OR) a

Service provided by Operating System is described below-

One set of operating-system services provides functions that are helpful to the user is given below-

- Communications- Processes may exchange information, on the same computer or between computers over a network.
  - Communication may be via shared memory or throug message passing which are two models of IPC-Inter Process Communication (packets moved by the OS).
- Error detection: OS needs to be constantly aware of possible errors
  - May occur in the CPU and memory hardware in I/O devices, in user program.

- for each type of error, OS should take the appropriate action to ensure correct and consistent computing.
- Debugging facilities can greatly enhance the users and programmers abilities to efficiently use the system.

Another set of OS functions exists for ensuring the efficient operation of the system itself via resource sharing.

☐ Resource allocation: When multiple users or multiple jobs running concurrently, resources must be allocated to each of them.

Many types of resources - Some (such as CPU cycles, main memory and file storage) may have special allocation code, other (I/O devices) may have general request and release code.

☐ Accounting: To keep track of ~~each~~ which users use how much and what kinds of computer resources.

Protection and security: The owners of information stored in a multiuser or network computer system may want to control use of that information, concurrent processes should not interfere with each other.

* Protection: involves ensuring that all access to system resource is controlled.
* Security of the system from outsiders requires user authentication, extends to defending external I/O devices from invalid access attempts.

If a system is to be protected and secure, precautions must be instituted throughout it. A chain is only as strong as its weakest link.

These are the services provided by operating system.

Answer to the question no: 4(OR)b

There are two types of schedulers—

(i) Long-term scheduler: (or job scheduler)
- selects which processes should be brought into the ready queue.

(ii) Short-term scheduler: (or CPU scheduler)
- selects which process should be executed next and allocates CPU.

The long-term scheduler controls the degree of multiprogramming.

Different types of process scheduling queues are described below—

(i) Job queue: set of all processes in the system.

(ii) Ready queue: set of all processes residing in main memory, ready and waiting to execute.

(iii) Device queue: set of processes waiting for an I/O device.

Processes migrate among the various queues.

Representation of Process Scheduling Queue: