University of Asia Pacific

Department of CSE

Name: Rashik Rahman

Reg ID: 17201012

Year: 4th

Semester: 1st

Course Code: CSE 405

Course Title: Operating System

Date: 29.04.2021

"During Examination and upload time I will not take any help from anyone. I will give my exam all by myself."

University of Asia Pacific

Admin Card

Final-Term Examination of Fall, 2020

Financial Clearance	PAID

Registration No : 17201012

Student Name : Rashik Rahman

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 330	Industrial Training	1.50	
3	CSE 401	Mathematics for computer Science	3.00	
4	CSE 403	Artificial Intelligence and Expert Systems	3.00	
5	CSE 404	Artificial Intelligence and Expert Systems Lab	1.50	
6	CSE 405	Operating Systems	3.00	
7	CSE 406	Operating Systems Lab	1.50	
8	CSE 407	ICTLaw, Policy and Ethics	2.00	
9	CSE 410	Software Development	1.50	
10	CSE 427	Topics of Current Interest	3.00	
		Total Credit:	23.00	

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: 25-Apr-2021 02:11 AM

Answer to the Q.NO.1

~~Ava~~

Available = (0%4, 1%4, 2%4)

= 012

~~Need~~ Here, Need = Max - Allocation.

	Allocation	Max	Need	Available
P_0	1 2 1	864	743	012
P_1	3 1 1	433	122	
P_2	4 1 3	9 1 3	500	
P_3	3 2 2	3 3 3	0 1 1	
P_4	1 1 3	5 4 4	4 3 1	
	P Q R	P Q R	P Q R	P Q R

Here, work = Available = 012

~~Safety Algorithm~~,

~~For P_0~~,

Finish = | False | False | False | False | False |

②

17201012

∴ Safety Alogonithm,

Fon P_0,

finish[0] is false but Need > wonk

so P_0 must wait

Fon P_1,

finish[1] is false but Need > woonk

so P_1 must wait

Fon P_2,

finish[2] is false but Need > wonk

so P_2 must wait

Fon P_3,

finish[3] is false and Need ≤ wonk

so, wonk = wonk + Allocation

= 012 + 322

= 334

And, finish[3] = Tnue.

Fo,

for P_4,

finish[4] = false and Need > work

so P_4 must wait.

Again,

for P_0,

Finish[0] = False, Need > work

so P_0 must wait,

for P_1,

Finish[1] = false and Need ≤ work

so, work = work + allocation

= 334 + 311

= 645

finish[1] = True

for P_2,

Finish[2] is false and Need ≤ work

so, work = work + allocation

= 645 + 413 = 10 5 8

Finish[2] = True.

④
17201012

For P_4,

finish[4] = false and Need ≤ work

∴ work = work + allocation

$= 10\ 5\ 8 + 1\ 1\ 3$

$= 11\ 6\ 11$

finish[4] = True

Again,

For P_0,

finish[0] = false and Need ≤ work.

∴ work = work + allocation

$= 11\ 6\ 11 + 1\ 2\ 1 = 12\ 8\ 12$

finish[0] = True.

So,

finish = | True | True | True | True | True |

And for the sequence: $P_3\ P_1\ P_2\ P_4\ P_0$

The system is in safe state and deadlock will not occur.

Ans.

Answer to the Q. NO. 2

~~String = 1720101217720.10.12~~

Final string = '2101027117201012'

Three page replacement algorithm are
- i) FIFO
- ii) Optimal
- iii) Least recently used

~~Sol~~

i) FIFO:

2 1 0 1 0 2 7 1 1 7 2 0 1 0 1 2

Total miss = 04

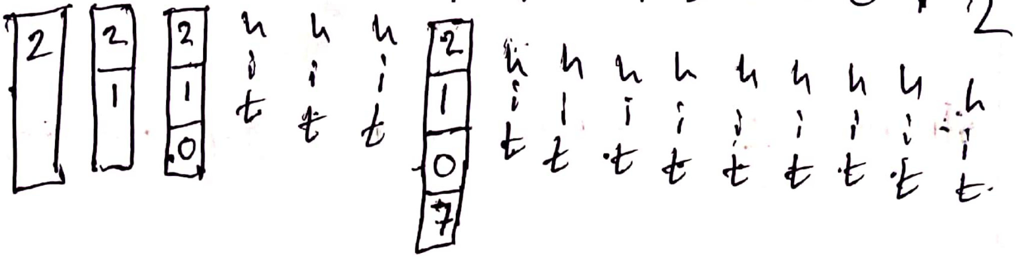
" hit = 12

6

17201012

ii) Optimal Algorithm;

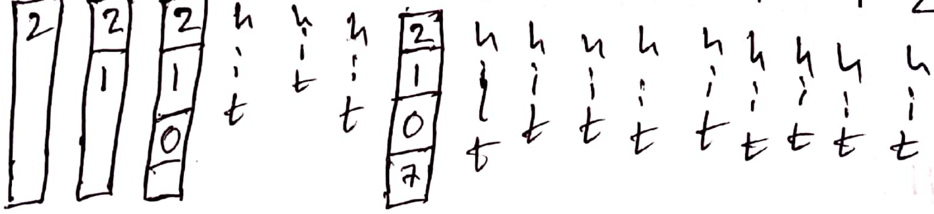
2 1 0 1 0 2 7 1 1 7 2 0 1 0 1 2



Total miss = 4
" hit = 12

iii) Least Recently used algorithm:

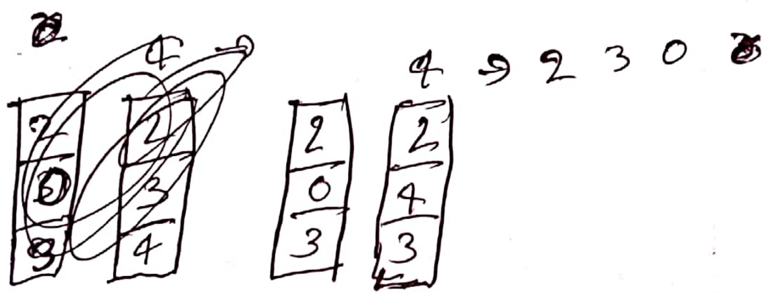
2 1 0 1 0 2 7 1 1 7 2 0 1 0 1 2



∴ Total miss = 4
" hit = 12

For the given scenarion all three algorithm has same perofonmance. But for another string i.e. 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1 with window = 3 the algorithms performs differently. For this example page miss/fault in FIFO is 15, optimal is 9 and LRU is 12.

According to this example and my ~~thought~~ allover observation Optimal Algorithm is best among all three algorithms. Because it considers the number that will appear next in the string close to the current number. So it keeps that number and replaces (in case of miss) the current number with the number (in the buffer) that will appear latter in the string. Like



Here in the string after 4, 0 appears at the last so ~~to~~ 4 is replaced with 0 (in the buffer).

For this reason Optimal page replacement algorithm is the best.

(8)

17201012

Answer to the Q. NO. 3(a)

Process	Burst time	Priority	Arrival Time
A	5	8	0
B	2	5	2
C	8	5	3
D	7	2	1
E	15	1	2

i) Shortes Job First (Pre-emptive) :

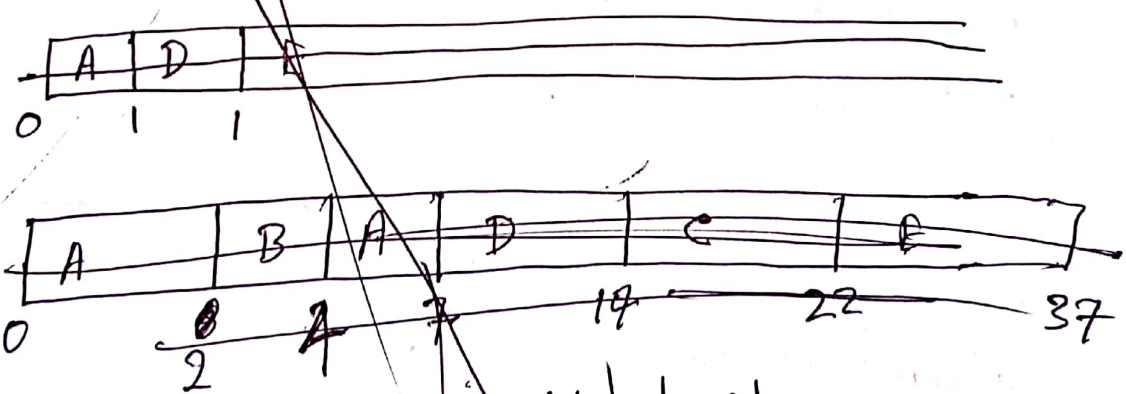


fig: ~~gantt~~ grant chart

Average wait time $= \{(22-2) + (14-3) + (7-1) + (4-0) + (2-2)\}/5$

$= 41/5 = 8.2$ ms

ii) Priority scheduling: (Pre-emptive)

A	D	E	D	B	C	A

0 1 2 17 23 25 33 37

Average wait time = $\{(33-1-0)+(25-0-3)$
$\cancel{+(23-0-2)}+(23-0-2)+(17-1-1)$
$+(2-0-2)\}/5$

$=(32+22+21+15)/5$

$= 18$ ms.

iii) Round Robin scheduling: (Pre-emptive)

quantum = 3

A	D	B	E	C	A	D	E	C	D	E	C

0 3 6 8 11 14 16 19 22 25 26 29 31

E	E

34 37

Average wait time = $\{(34-4\times3-2)+(25-2\times3-1)$
$+(29-2\times3-3)+(14-1\times3-0)$
$+(6-0\times3-2)\}/5$

(10)

17201012

$= (20+18+20+11+4) / 5$

$= 14.6$ ms.

ii) Shortest job first :

Gaant chart :

A	B	D	C	E

0 5 7 14 22 37

Average wait time $= \{(0-0)+(5-2)+(7-1) + (14-3) + (22-2)\} / 5$

$= (3+6+11+20) / 5$

$= 8$ ms.

Answer to the Q. No. 3(b)

Major differences between system call and function call:

System call	Function call
A function provided by the kernal to enter kernal mode to access a ~~resour~~ recourse	A request made by a program or script that execute a predetermined function
Context switching occurs in system call	There's no context switch occurance in function call
Allows the programs to access memory or a hardware resource from the kernal	Helps to pass the control to a specific function and to execute the defined ~~taks~~ task

Explaination of mechanism:

Function call is executed in user mode where task of that function is either predefined or given by programmer/user. But for a system call the scenario is different.

(12)
17201012

When a system function is called then the cpu switches to high pravilage and run in kernal mode. To do this a trap function is called and after execution of system call return from trap happens. But the cpu may return to another process in the user mode instead of ~~of~~ the process it left in the user mod: OS does this by the means of context switch and cpu time shaping.

13
17201012

Answer to the Q.No. 4(a)

Whenever we run a program a process is created by OS. A process goes through 5 ~~steps~~ states in total from when it starts to when it ends. The ~~proces~~ states of a process ~~is~~ is given below:

i) **Running**: In this ~~p~~ state the process is currently being executed on CPU

ii) **Ready**: Waiting to be scheduled for execution on CPU

iii) **Blocked**: Here a process may be suspended while it is running due to interruption.

iv) **New**: Process is created

~~v) Terminated~~

v) **Dead**: Process is terminated.

Scanned with CamScanner

(14)

17201012

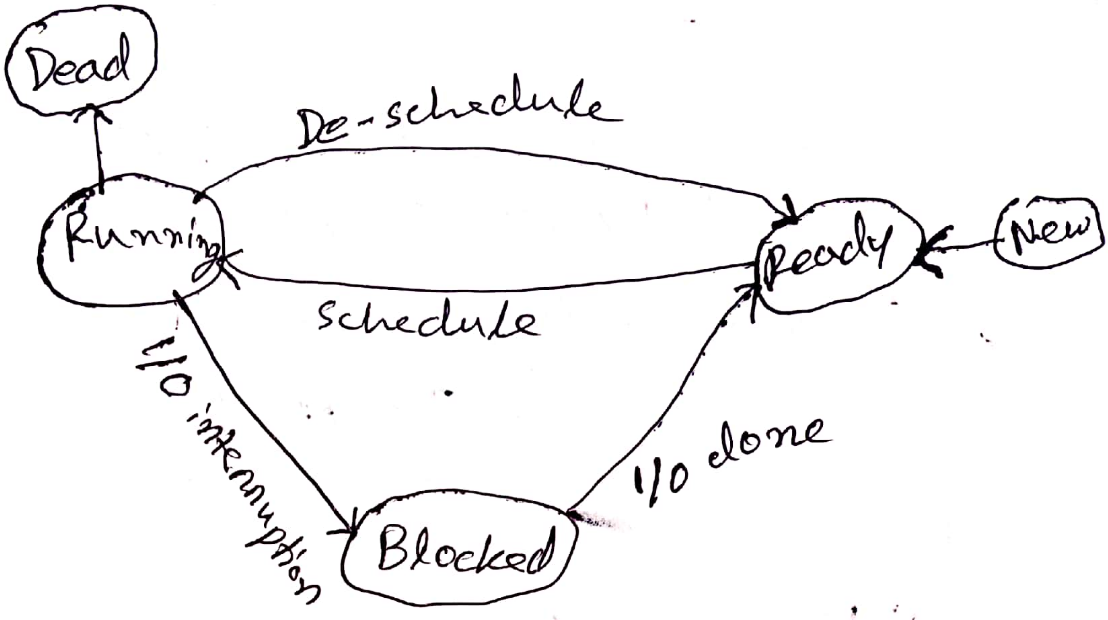


Fig: Diagram of process states.

Answer to the Q.No.4(b)

Pipe is communication medium between two or more related or interrelated process. It can be either within one process or a communication between child and ~~prane~~ parent process. Pipe system call return two file descriptors; read & write handle. Pip is a half-duplex communication. In regular pipe both file descriptors are in same process but in name pipe both file descriptors can conned different endpoints of two different process.

Sockets allow communication between two different processes on the same or different machine. It basically uses FTP. There are two types of socket.

16
17201012

Share memory vs messege passing.

Shared memory	Message passing
Shared memory region is used for ~~commun~~ communication	Message passing facility through kernal is used for communication
Used for communi-cation in between processes	Used for distributed environment
Relitavely fast	Relitavely slow
Data consistency needs to be ensured by process	Data inconsistency or conflicts don't need to be resolved

Scanned with CamScanner