Department : Computer Science and Engineering

Program: B.S.c in CSE

Course no: CSE 3213

Course Title: Operating System

Examination: Semester Final

Session: Spring 2020

Student no: 1702041105

Signature & Date:
Md. Mushfiqur Rahman
24/05/2021

Scanned with CamScanner

CSE 3213
170204105
Mushfiq

Answer to the Question no: 5(a)

Page fault: Page fault occurs when a Program attempts to access a block of memory that is not stored in the phsical memory or RAM. The fault notifies the operating system that it must locate the data in virtual memory, then transfer it from the storage device such as HDD, or SSD to ~~the~~ the system RAM.

Scanned with CamScanner

170204105
CSE 3213
Mushfiq

Answer to the question no: 5 (a)

Given that,

Page frame size : 16 KB

RAM size : 16 MB

The address space size of the process : 32 MB

$$\text{No. of Frame} = \frac{\text{RAM size}}{\text{Page size}}$$

$$= \frac{16 \times 2^{20}}{16 \times 2^{10}}$$

$$= 1024$$

$$\text{No. of Pages in the process} = \frac{\text{Process size}}{\text{Page size}}$$

$$= \frac{32 \times 2^{20}}{16 \times 2^{10}}$$

$$= 2048$$

Scanned with CamScanner

170204105
CSE 3213
Mushfiq

Offset size = 16×1024

= 16384

No. of frame = 1024 = ~~10 bit~~ 2^{10} = 10 bit

No. of Page = 2048 = 2^{11} = 11 bit

Offset = 16384 = 2^{14} = 14 bit

logical address

11 bit	14 bit
Page no.	offset

Physical address

10 bit	14 bit
frame no.	offset

Scanned with CamScanner

170204105
CSE 3213
Mushfiq

Given that,

the requested data is available in the 28th memory frame

Process request data from 200th page with offset 15200

$(28)_{10} = (00000\ 111\ 00)_2$

$(200)_{10} = (00\ 01\ 10\ 01\ 000)_2$

$(15200)_{10} = (111011\ 0110\ 0000)_2$

logical address (25 bit)

00011001000	1110110110 0000

Physical address (24 bit)

000001 1100	1110110110 0000

Scanned with CamScanner

17 02 04 105
CSE 3213
Mushfiq

Answer to the question no: 5(b)

Using Least Recently Used (LRU)

0	15	4	2	12	0	1	7	9	7	15	0	4	7
0	0	0	2	2	2	1	1	1	1	15	15	15	7
	15	15	15	12	12	12	7	7	7	7	7	4	4
		4	4	4	0	0	0	9	9	9	0	0	0
P	P	P	P	P	P	P	P	P		P	P	P	P

10	5	21	3	11	5	1	21
7	7	21	21	21	5	5	5
4	5	5	5	11	11	11	21
10	10	10	3	3	3	1	1
P	P	P	P	P	P	P	P

Total Page fault : 21

Scanned with CamScanner

170204105
CSF 3213
Mushfiq

Using Optimal Page replacement algorithm.

0	15	4	2	12	0	1	7	9	7	15	0
0	0	0	0	0	0	0	0	9	9	9	0
	15	15	15	15	15	15	15	15	15	15	15
		4	2	12	12	1	7	7	7	7	7
P	P	P	P	P		P	P	P			P

4	7	10	5	21	3	11	5	1	21
4	4	10	5	5	5	5	5	1	1
15	15	15	15	21	21	21	21	21	21
7	7	7	7	7	3	11	11	11	11
P		P	P	P	P	P		P	

Total ~~PA~~ Number of Page fault = 16

Here, we can see that the number of Page fault of LRU is 21 and in optial is 16.

~~No~~ So, the number of optimal replacement Page fault is less then LRU. for these reason,

Scanned with CamScanner

170204105
CSE 3213
Mushfiq

Optimal cannot be implemented as OS will not have the future request in hand already. So, scaning the past request LRU will give comparative less page faults.

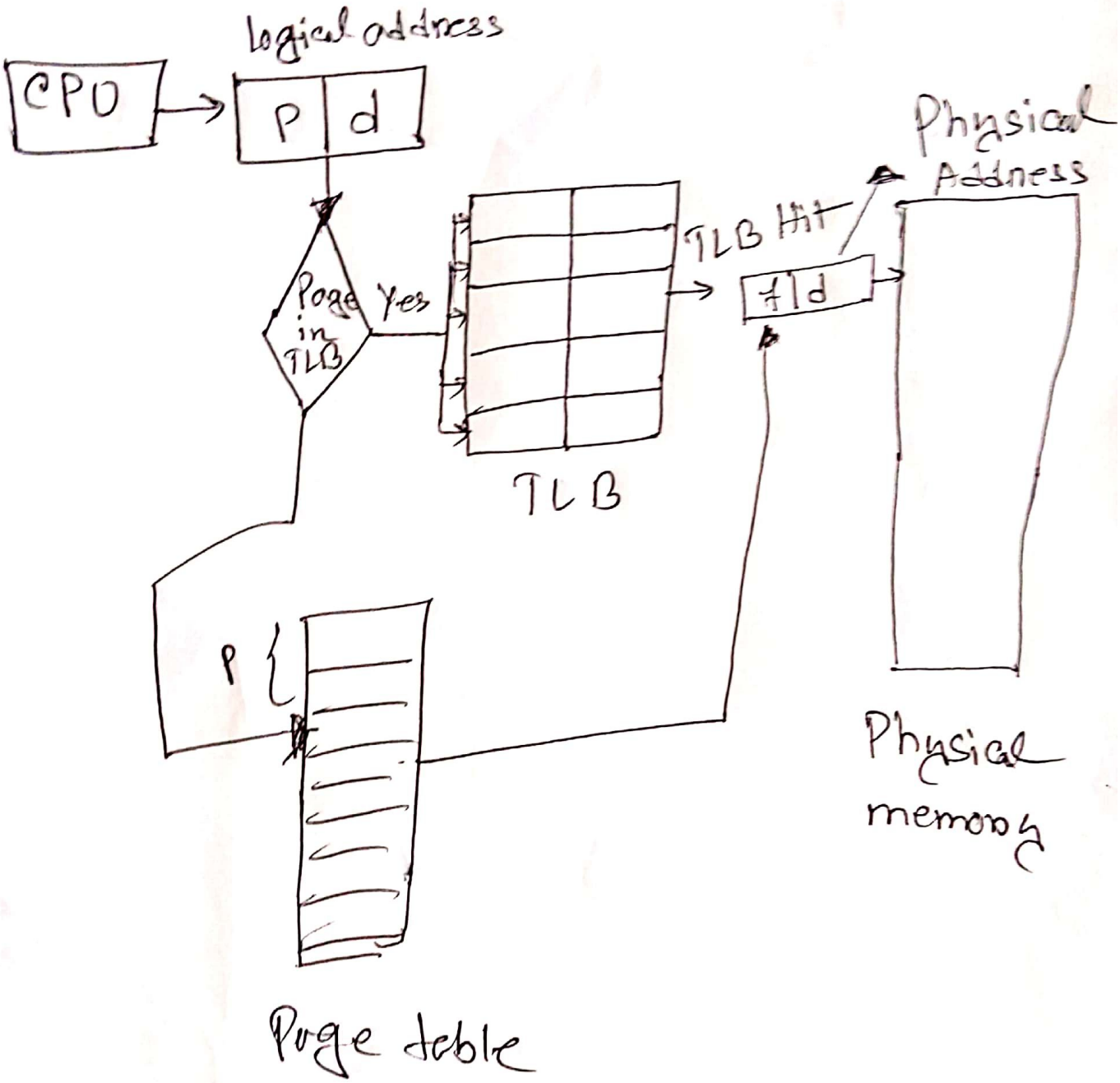
Answer to the question no: 5 (c)

Drawing the diagram for an address translation process in any virtual memory system with a translation look-aside Buffer.

Scanned with CamScanner

170204166
CSE 3213
Mushfiq

Answer to the question no-5(c)



Scanned with CamScanner

17020410b5
CSE 3212
Mushfiq

Answer to the question no; 7 (a)

Threads are used for small tasks, whereas process are used for more heavy weight task besically the ~~exe~~ execution of application. Another difference between a thread and a process is that threads with in the same process its share the same address space, whereas different process do not.

A thread creation of a process is preferable than replicating a process when the process has to deal with multiple client request. For example assume a process P1 is assigned with the objective of fetching result of exam for a group of students P1 has to

Scanned with CamScanner

1702041105
CSE 3213
Mushfiq

get required info from multiple students, fetch necessary data from the server and present it to students. M is a large amount replicating P1 n times is a memory and CPU intensive process. So, to ignore that hassle, creating threads of P1 is more Convenient.

When a new thread is created it shares its code section, data section and operating system resources like open files with other threads.

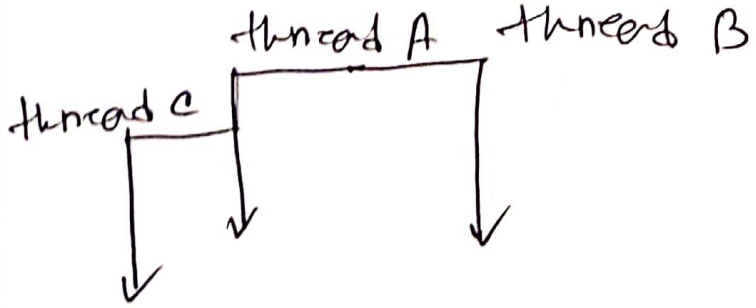


Fig: Thread Creation

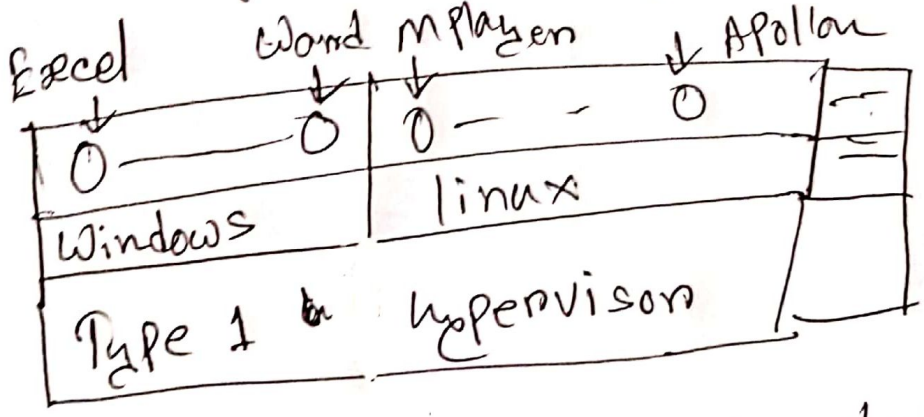
Scanned with CamScanner

170204105
CSE 3213
Mushfiq

Answer to the Question no: 07(b)

Virtual Machines:

Type-1 hypervisors run directly on the system hardware. They often referred to as a native or bare metal or embedded hypervisors in vendor just like an operating system Type-1 hypervisors provide higher performance, availability and security than type-2 hypervisors



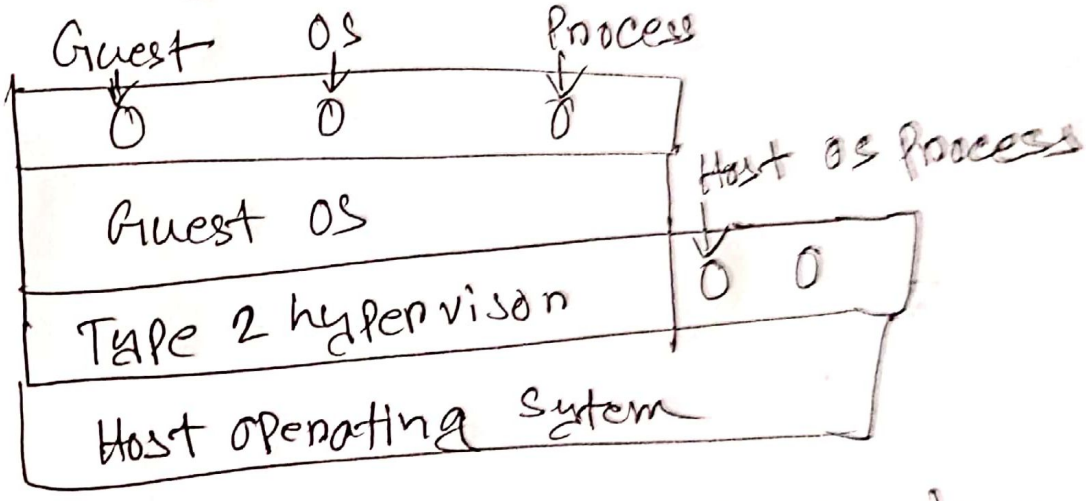
Type-1 hypervisors run on dedicated hardware. They required management console and are used in data centers. Example of this type of hypervisor include Oracle OVM for SPARC

Scanned with CamScanner

17020105
CSE 3213
Mushfiq

ESXi, Hyper-V and KVM.

Type-2 hypervisors run on a host operating system. When the virtualization movement first began to take off type-2 hypervisors were most popular. Administrators could buy the software and install it on a server they already had.



Examples of this type of hypervison include VMware Fusion, Oracal Virtual Box, oracle vm for x86, Solaris zones, Parallels and vmware workstation.

Virtual machine is called hypervison.

Scanned with CamScanner

1702041105
CSE 3213
Mushfik

Answer to the Question no: 7(c)

Deadlock Recovery:

A ~~tt~~traditional operating system such as windows doesn't deal with deadlock recovery as it is time and space consuming process. Real-time os used. Deadlock Recovery method:

1. Killing the process: Killing all process involved in the deadlock. Killing process one by one. After killing each proces check for deadlock again keep repeating the process till system recover from deadlock.

2. Resource Preemption:

To eliminate deadlocks using resource preemption, we preempt

Scanned with CamScanner

170204105
CSE 3213
Mushfiq

some resources from process and give those resources to other processes. This method will raise three issues.

(a) Selecting a victim:

We must determine which resources and which processes are to be preempted and also the order to minimize the cost.

(b) Rollback:

We must determine what should be done with the process from which resources are preempted. One simple idea is total Rollback. That means about the process and restart it.

Scanned with CamScanner

170204105
CSE 3213
Mushfiq

(c) Starvation:

In a system, it may happen that same process is always picked as a victim. As a result, that process will never complete its designated task. this situation is called starvation and must be avoided. One solution is that a process must be picked as a victim only a finite number of times.

Scanned with CamScanner

170204105
CSE 3213
Mushfiq

Answer to the Question no: 3(a)

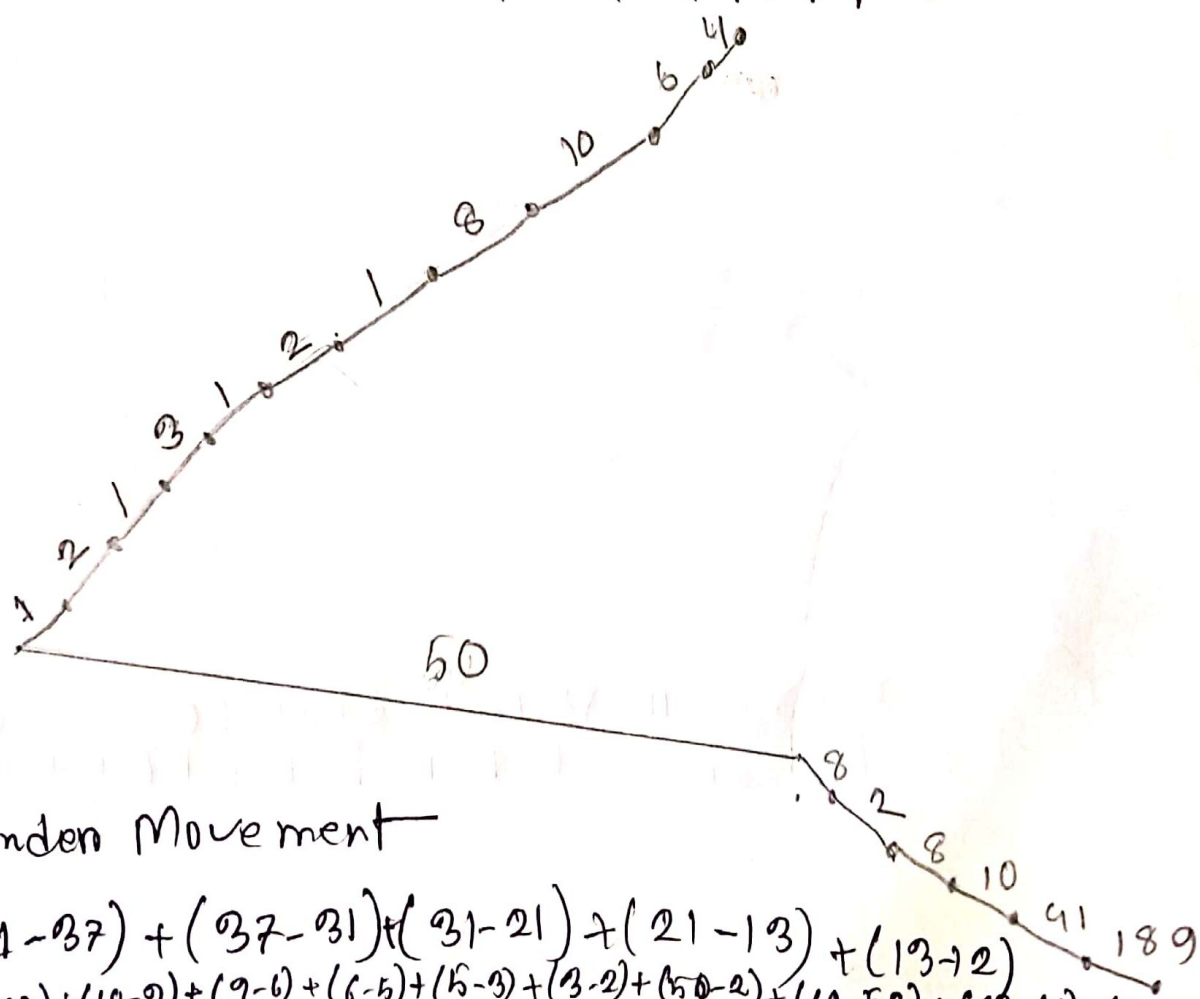
Given that,

Queue = 12, 6, 3, 121, 52, 9, 80, 70, 37, 62, 2, 60, 10, 310, 9, 31, 21, 5, 13

head start at 41.

SSTF

0 2 3 5 6 9 10 12 13 21 31 37 41 52 60 62 70 80 121 310 50



Cylinder Movement

$= (41-37) + (37-31) + (31-21) + (21-13) + (13-12) + (12-10) + (10-9) + (9-6) + (6-5) + (5-3) + (3-2) + (52-2) + (60-52) + (62-60) + (70-62) + (70-62) + (80-70) + (121-80) + (310-121) = 397$

16

170204105
CSE 3213
Mushfiq

Answer to the question no - 3(a)

C-look

0 2 3 5 6 9 10 12 13 21 31 37 41 52 60 62 70 80 121 310 500

Cylinder movement = 11 + 8 + 2 + 8 + 10 + 41 + 189 + 1 + 2 + 1 + 3 + 1 + 2 + 1 + 8 + 10 + 6 = 304

Hence, cylinder movement of C-look is lower than SSTF. So, C-look is Better.

Scanned with CamScanner

CSE 3213
170204105
Mushfiqun

Answer to the question no: 3 (b)

block size = 4KB

Pointer size = 4 byte

For 1st block = (4 × 12) KB = 48 KB

The total no of Pointers = $\frac{2^{10} \times 4}{4}$

= 1024

For 13th block = 1024 × 4 KB = 4 MB

For 14th block = 1024 × 1024 × 4 KB

= 4 GB

For 15th block = 1024 × 1024 × 1024 × 4 KB

= 4 TB

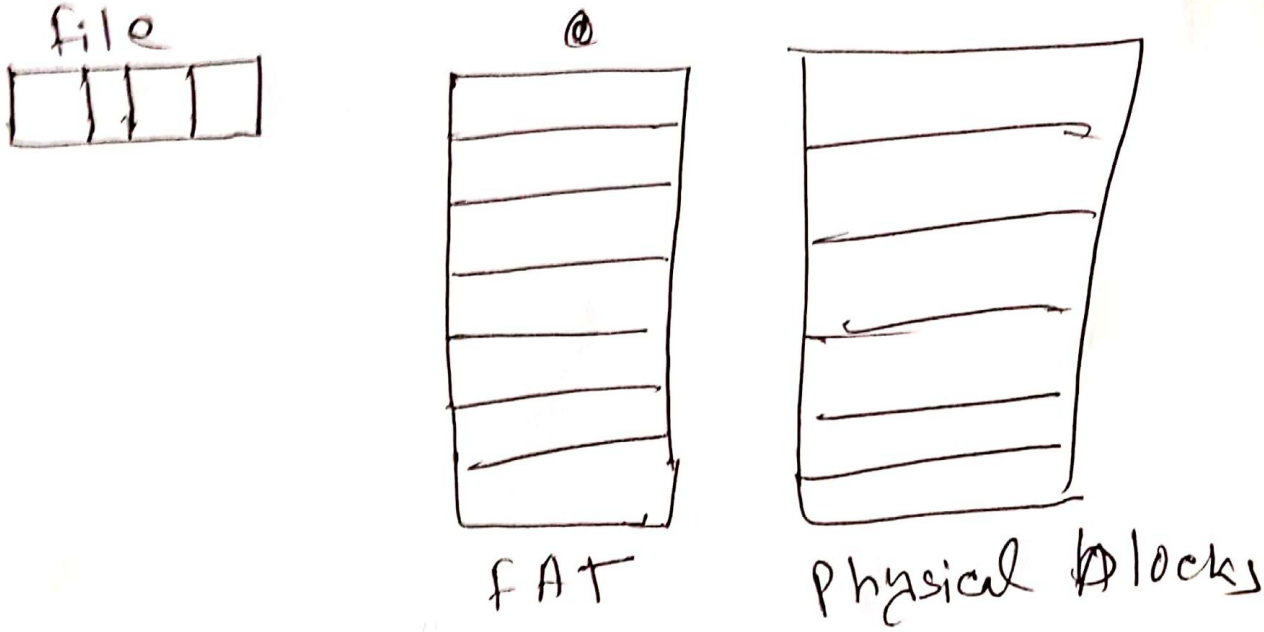
Maximum file size = 48 KB + 4 MB + 4 GB + 4 TB

Scanned with CamScanner

170204105
CSE 3213
Mushfiqun

~~Here, cylinder movement of SSC-Look is lower than SSTF. So, C-Look is Better.~~

Answer to the question no: 6c)



Directory Table

File name	Starting block number

At first, the file is separated into block and they are saved into the empty spaces of the physical block.

Scanned with CamScanner

170204105
CSE 3213
Mushfiqun

But it is not Contiguous. So, the next block's address is also needed to be saved. There file Allocation table is used. Where the address of the next block is saved and it is Continued.

Scanned with CamScanner

1702091055
CSE 3213
Mushfiq

Answer to the Question no: 4(A)

The formula to estimate CPU time is,

$$E_{n+1} = \alpha t_n + (1-\alpha) E_n$$

where,

E_{n+1} = Estimated time at n+1

α = Constant

t_n = Actual time

E_n = Estimated time at n

To, Simplify,

$$E_{n+1} = \alpha t_n + (1-\alpha) E_n$$

$$= \alpha t_n + (1-\alpha) \alpha t_{n-1} + (1-\alpha)^2 E_{n-1}$$

$$= \alpha t_n + (1-\alpha)\alpha t_{n-1} + (1-\alpha)^2 \alpha t_{n-2} + (1-\alpha)^3 E_{n-2}$$

$$= \alpha t_n + (1-\alpha)\alpha t_{n-1} + (1-\alpha)^2 \alpha t_{n-2} + (1-\alpha)^3 E_{n-2} + \cdots (1-\alpha)^{n+1} E_0$$

Scanned with CamScanner

CSE 3213
170204105
Mushfiq

We can see that, the equation is exponentially weighted average the value falls exponentially. Depends on E_0 (avg system time) that varies computer to computer.

Ideal value for α is 0.5.

If we use $\alpha = 1$, then

$E_{n+1} = t_n$

$\alpha = 0 \Rightarrow E_{n+1} = E_n$

$E_2 = E_1 = E_0$

meaning estimating same value forever.

<u>Race Condition:</u> Race condition is a computer operating senario where two or more process have share

Scanned with CamScanner

17020410n
CSE 323

resource and try to access that at the same time and eventually data gets over-writer one or more processes fail to do this job.

In Printer daemon programe there's a queue where many process puts their data for printing and since the ~~proce~~ resources are shared multiple process can put data in same slot if context switches occurs before accessing the in, out countens this can also happen in Airline or Bus/Train ticket reservation.

Scanned with CamScanner

17020410 5

Answer to the question no-4(b)

To prevent race condition some solutions are work good they provide mutual exclusion, no process outside critical region can block other processes. Some solution, that can do this are peterson, TSL, XCHG. But they have busy waiting problems meaning some process inside critical zone finished its task. Checking if the zone is unlocked or not. This busy waiting problem not only wastes CPU time

(24)

Scanned with CamScanner

but also its resources too.

That's why busy waiting based solution towards the conditional problem not preference.

Semaphore:

Semaphore is a data structure used to solve race condition and busy waiting prg problem at the same time.

There are two semophore

① Binary semaphore

② Conting semaphore

Scanned with CamScanner

12020405

Answer to the question no: 4(c)

Process: Process is a programe in execution. A process generally includes stack, which contents temponary data a data section a text section, also it may have heap section which is dynamically allocated during process run time

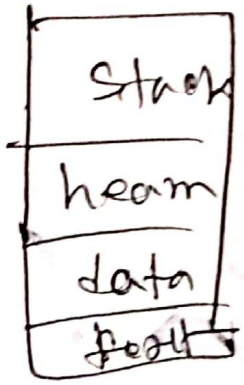


fig: Process in memory

Scanned with CamScanner

12020105

Program: Program is a set of instruction that should be executed sequentially to perform a task.

Programs are written by developers and process is handled by OS.

Scanned with CamScanner

120250105

Answer to the Question no 2(A)

In this Condition, n number of Reader and m number of writer can be in the system.

A race condition free algorithm of the system.

```
Reader() {
  while (true){
    down(mutex);
    read_Count ++;
    if (read_Count == 1)
        down(db);
    up(mutex)
    Read_DB()
    down(mutex);
```


Scanned with CamScanner

17020410

```
        read_Count --;
        if (read_Count == 0)
            up(db);
            up(mutex);
    }
}

writer(i) {
    while (true) {
        Prepare_Data(i);
        down(writer mutex);
        writer_Count ++;
        if (writer_Count == 1)
            down(mutex);
        up(writer mutex);
        down (Exclusive);
        write(data);
        up(Exclusive);
```


Scanned with CamScanner

```
    down (writer mutex);
      writer_Count--;
      up (writermutex);
      if(writer_Count ==0)
        up(dp);
    }
}
```

Answer to the question no - 2(b)

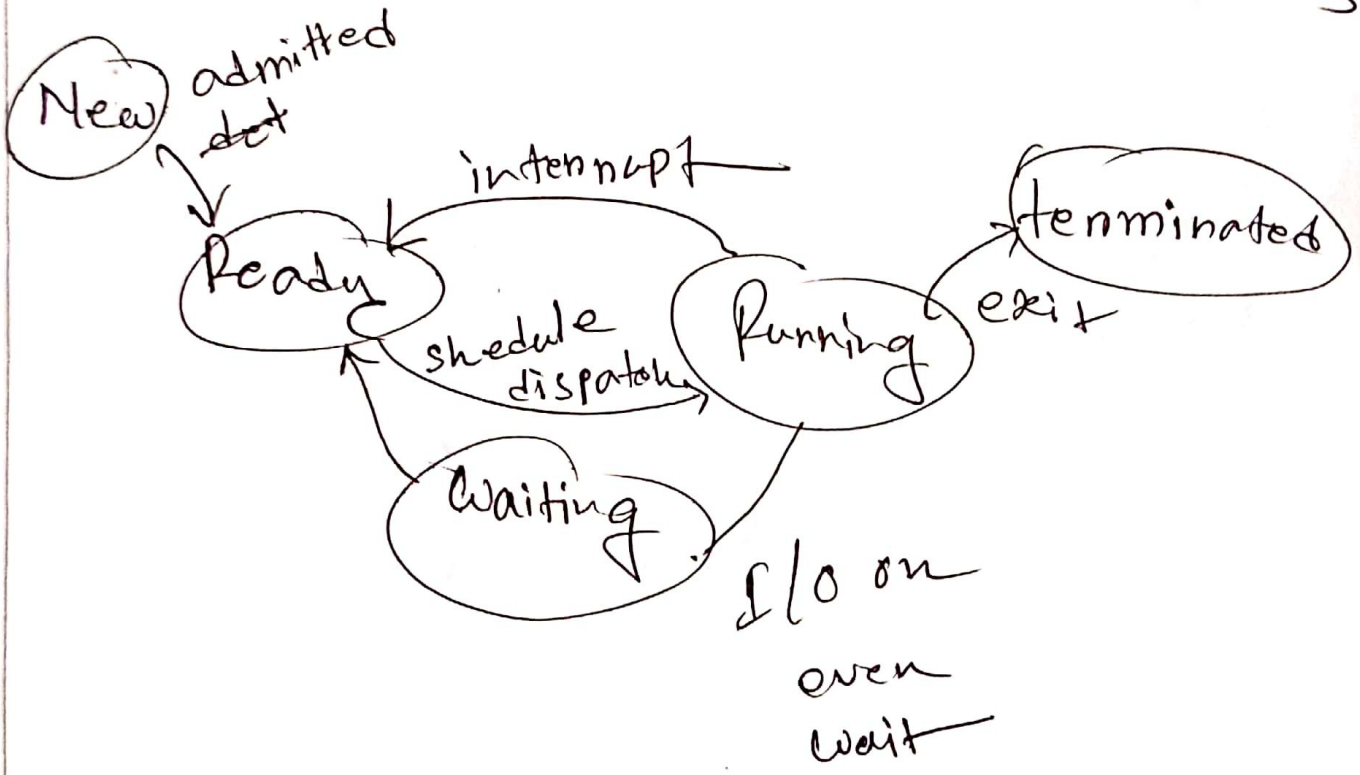
Dispatcher runs in kernal mode.
The main responsibility of dispatcher is context switching. It switching the kernel mode to user mode.
The another task of dispatcher is

Scanned with CamScanner

17/02/2021

to transfer Process from ready queue to CPU. It keeps track of Program counter of a Process.

State transition diagram of a process



When a Process is created, it goes to ready queue. Then it goes to running state. If the process completes the

Scanned with CamScanner

execution. then it goes to terminate state. If, the process request I/O it goes to waiting state. After Completing I/O request, the process back to ready state. If the time quantum of the process is over and the next process from queue goes to running state.

Scanned with CamScanner

17020405

Answer to the Question no -2(c)

Round Robin : q = 4

Pno	A.T	C.T
P1	5	12/8/4/0
P2	0	11/7/3/0
P3	4	9/5/1/0
P4	12	17/13/9/5/1/0
P6	35	13/9/5/1/0

~~P2~~	~~P3~~	~~P2~~	P1	~~P3~~	~~P4~~	~~P2~~	~~P1~~	~~P3~~	~~P4~~	~~P1~~	~~P6~~	~~P4~~	~~P6~~	~~P4~~	~~P6~~

~~P4~~	~~P6~~

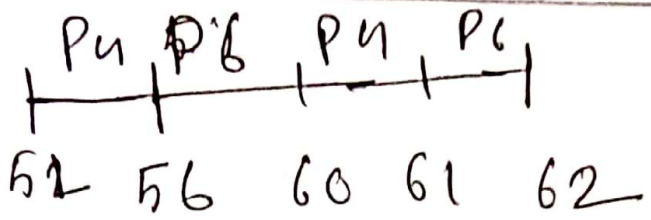
P2 P3 P2 P1 P3 P4 P2 P1 P3 P4 P1 P6 P4 P6 P4 P6

0 4 8 12 16 20 24 27 31 32 36 40 44 48 52 56 60 61 62

(a,b)

Scanned with CamScanner

17020410 5



waiting time

P1 = 23

P2 = 16

P3 = 19

P4 = 32

P5 = 14

$$A.W.T = \frac{23+16+19+32+14}{5} = 20.8$$

$$A.R.T = \frac{7+8+5}{5}$$

$$= \frac{20}{5} = 4$$

34

Scanned with CamScanner

1202410015

SRTN:

P	A.T	C.T
P1	5	12/0
P2	0	11/0
P3	4	9/0
P4	12	17/14
P6	35	13/0

G.C

P2	P3	P1	P4	P6	P4

0 11 20 32 35 48 62

W.T = P1 = 15

P2 = 0

P3 = 7

P4 = 33

P6 = 0

35

Scanned with CamScanner

$$A.W.T = \frac{15+7+33}{5} = 11$$

$$A.R.T = \frac{15+7+20}{5} = 8.4$$

We see that for SRTN, avg wait ing time is lower and Avg response time is higher that RR. So, SRTN is better than RR in this scenario.

Scanned with CamScanner