22/04/2021

CSPC43 – OS

106119100

Rajneesh.

Cycle Test #02

Question (1)

Process	Allocation R_1 R_2 R_3	Max. R_1 R_2 R_3	Available R_1 R_2 R_3	Need R_1 R_2 R_3
P_1	2 2 3	3 6 8	2 3 0	1 4 5
P_2	2 0 3	4 3 3	4 3 3	2 3 0
P_3	1 2 4	3 4 4	[5 5 7]	2 2 0
			[7 7 10]	

$P_2 = [230] + [203] = [4\ 3\ 3]$

$P_3 = [433] + [1\ 2\ 4] = [5\ 5\ 7]$

$P_1 = [557] + [2\ 2\ 3] = [7\ 7\ 10]$

Safe state

Safe sequence = $P_2\ P_3\ P_1$

Request : $P_1(1, 1, 0)$

New Allocation ,

$\Rightarrow P_1(2\ 2\ 3) + P_2(1\ 1\ 0) = (3\ 3\ 3)$

Available .

$\Rightarrow (2, 3\ 0) - (1, 1, 0) = [1, 2, 0]$

Process	Allocation			Max			Available			Need		
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3
P_1	3	3	3	3	6	8	1	2	0	0	3	5
P_2	2	0	3	4	3	3	3	2	3	2	3	0
P_3	1	2	4	3	4	4	4	4	7	2	2	0
							7	7	10			

$P_2 = [1\ 2\ 0] + [2\ 0\ 3] = [3\ 2\ 3]$

$P_3 = [3\ 2\ 3] + [1\ 2\ 4] = [4\ 4\ 7]$

$P_4 = [4\ 4\ 7] + [3\ 3\ 3] = [7\ 7\ 10]$

∴, safe state.

safe sequence = $P_2\ P_3\ P_1$

Question (2)

(i) Direct Communication

In Direct communication mode, each process must explicitly specify the sender or the reciver.

Ex: send (P, message)
recieve (Q, message)

This process sends a message to process P and recieves a message from process Q. It explicitly specified the sender and the reciver

The logical link created b/w these types is associated with exactly 2 process and the process must know the producer process and consumer process this is called symmetry in addressing

(ii) Blocking send

The sending process is blocked until the recieving process gets the message from the mail box

(iii) zero capacity buffer In this process queue, maximum capacity is 0.

Hence message can't exist in this case if, the process wants to make sure that the reciver gets the message.

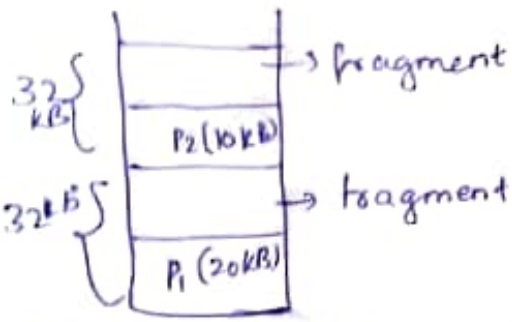
(3)

Fragmentation: Fragmentation is a scenario in which memory space is used inefficiently, reducing capacity and performance

Fragmentation is of two types:-

Internal fragmentation

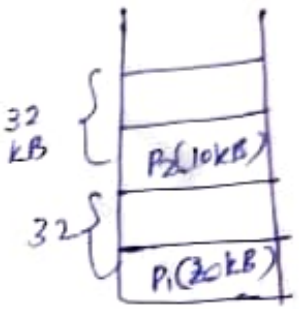
→ occurs when process is allocated to a memory block whose size is more than that of process



→ Paging results in internal fragmentation

external fragmentation

→ occurs when memory space is available but not-contiguous



but non-contiguous

→ Segmentation (cause)

Solutions to internal fragmentation:

→ Following partitioning algorithms knowing the incoming processes in prior like bestfit, firstfit

→ Dynamic partitioning (or) dynamic allocation can solve internal fragmentation.

→ Spanning can also solve internal fragmentation upto some extent.

Solutions to external fragmentation:

→ Compaction, paging can be used to solve the problem of external fragmentation.

→ Dynamic relocation instead of static relocation

→ Spanning

→ Segmentation using dynamic allocation.

→

Thrashing: thrashing occurs when CPU gets involved most of its time in swapping in and out of physical memory (main memory & disk)

Reasons

→ High degree of multiprogramming

→ Less main memory (RAM) ≈ Less number of frames

Solutions to thrashing:

→ Allocate small and high priority processes first

→ Instructing schedulers

→ Having a limit to number of swappings and after that long term scheduler should not bring processes into memory until inside processes are completed.

→ Suspending some processes if system gets into thrashing.

→ Increasing number of frames, decreasing degree of multiprogramming.

Question (4)

Functions of interrupt handler and device drivers.

→ **Device Drivers:**

In computing, device drivers is a computer program that operates or controls a particular type of device that is attached to a computer.

a. They are hardware dependent and operating system specific

→ **Interrupt handelers:**

An interrupt handlers is fusion that kernal run in response to a sepecifk interrupt:

A device that generates interrupts has an associated interrupt handler.

The interrupt handlers for a device is pat of device. Interrupt handlers are invoked in response to interrupts and they run space context called interrupt context. An interrupt handers job to. acknowledge the interrupt recept of hardware.

However, interrupt handlers are often have a large amount of work to per form.

Question 5

Consistency semantics is the concept which is used by users to check file systems which are supporting file sharing in their systems. Basically, its specification to check that how in a single system multiple users are getting access to same file at same time

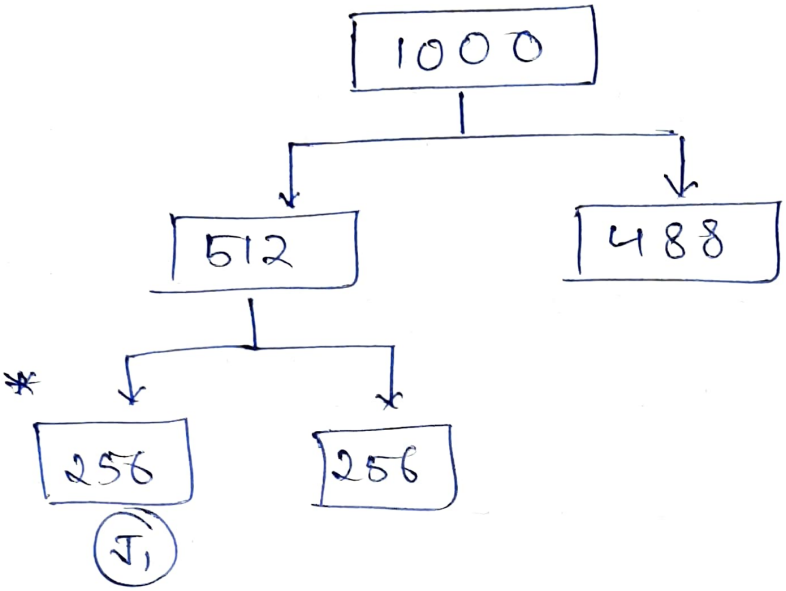
this concept which is in a direct relation with synchronisation algorithm.

Question 6

* Job1 ⟶ 200K

Avaible ⟶ 1000K

128 < 200 < 256



* Job2 ⟶ 500K

256 < 500 < 512

waits J1 is dellocated

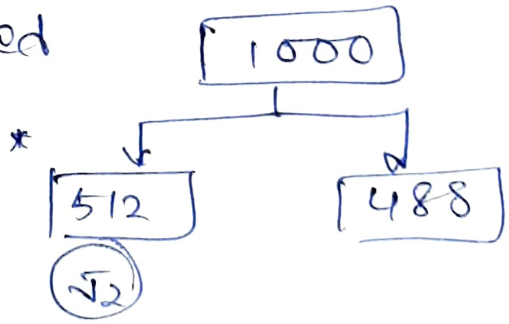
* J3 ⟶ 450K

256 < 450 < 512

waits until J1 is deallocated, waits with J2

* Job 1 finished

Naw, J2 is allocated

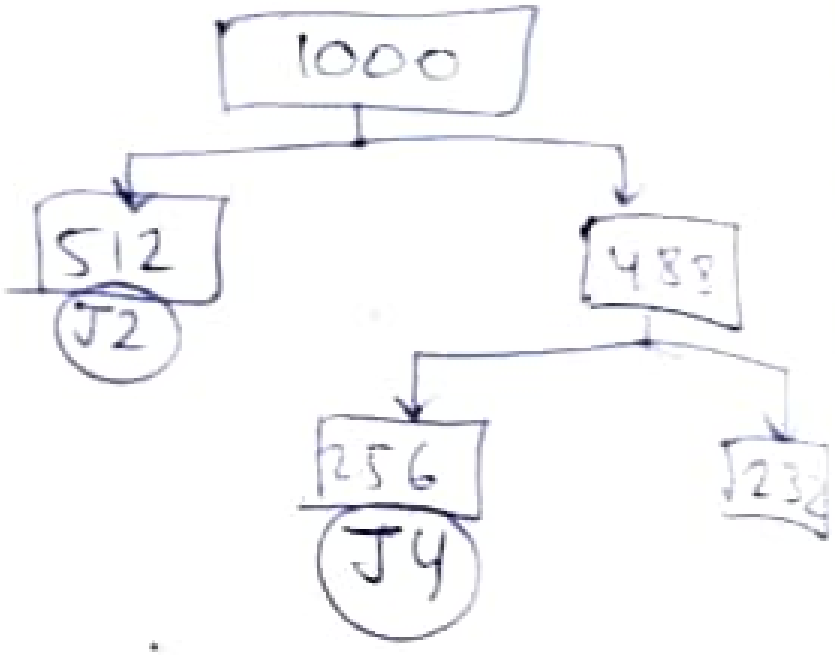


* Job3 waits until J2 finish

* Job4 → 150K

128 < 150 < 256

∴ J4 is allocated



* Job 5 → 300 K

256 < 300 < 512

J5 waits until J2 finishes.

∴ After J2 finishes, J3 is allocated and then after J3 finishes; J5 is allocated.