CSE325 (Section-2)

Mid-1

Md. Adnan Monshed

Id: 2020-1-60-155

Roll: 28

Ans: to the q.n. 1

a batch process

P0			P1			P2			P3		
I0 4			I1 8			I2 11			I3 9		
	C0 11			C1 17			C2 18			C3 6	
		O0 3			O1 6			O2 3			O3 7
4	11	3	8	17	6	11	18	3	9	6	7

Total = 103

I = Input

CPU = CPU burst time

O = output

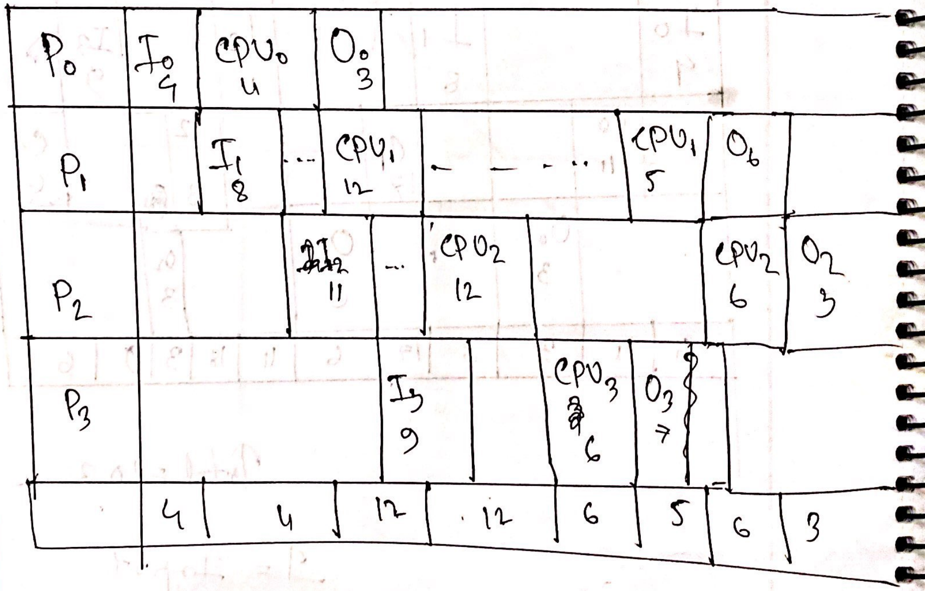
The execution time of the process

= 103

Ans. to the q. n. 1(b)

time sharing process

time ~~slice~~ quantum =12



total = 59

I = Input

CPU = CPU burst time

O = output

total execution time = 59

Ans.to the q n:2

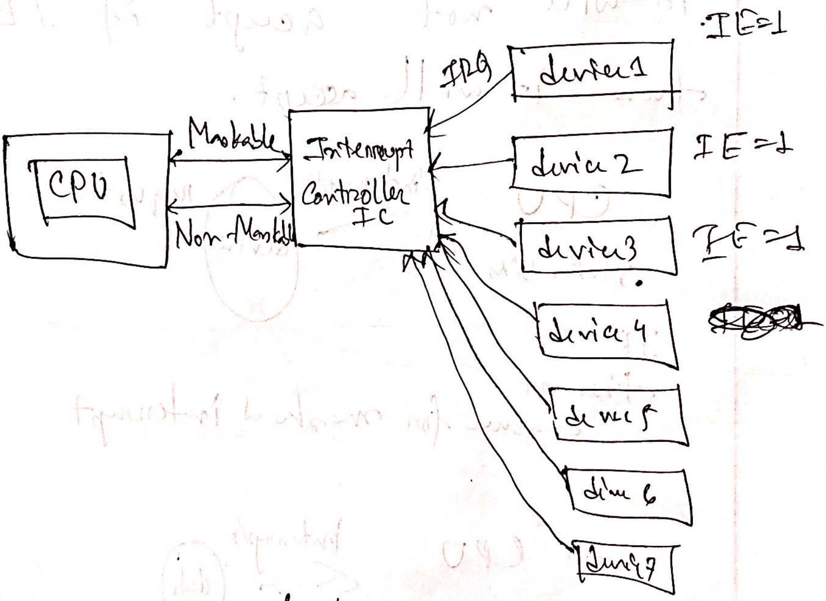
Parent pid will create child1 in pid = fork(); child1 pid will be equals to 0 so it will enter the if block. and create child2 in pid1= fork(); and child 2 will print "East West University and complete ~~its~~ its execution. for the mean time child 1 will also print" East West University. The parent pid will be greater than 0 so it will go to else statement and will print "EWU reopens on February 27, 2022". Output will be:

Parent prints EWU reopens on February 27, 2022

Child1 prints East West University

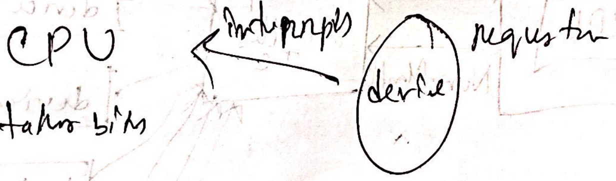
Child2 prints East West University

Ans. to the q.n. 3



for non-maskabe CPU IE = 1 . so no need to check_it will always acept the internupt . Interrupt hander will always send the interruption to CPU for those 3 non-muskable intorrupts.

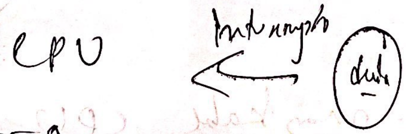
and for the 4 maskable interrupts if the IE=~~8~~0 then it will not acupt if IE=1 then it will accept.



status bits

IE=0

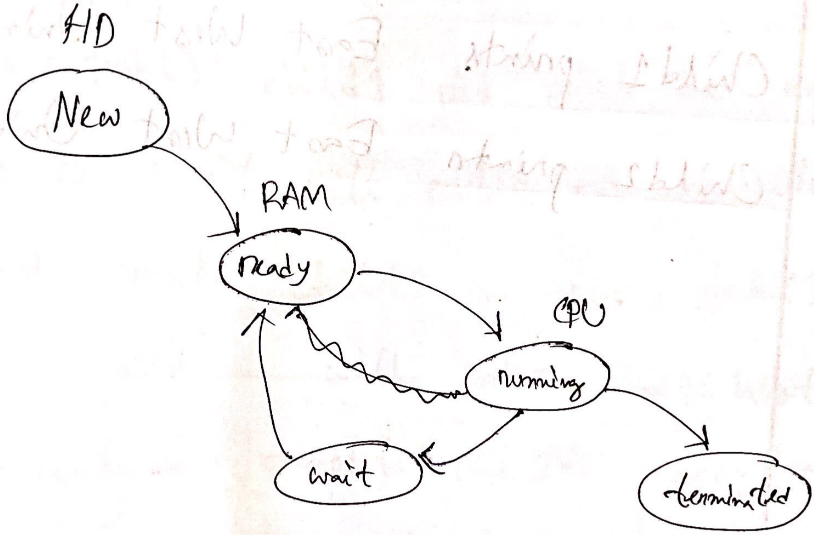
either or
erasure for masked interrupt



IE=1
enable

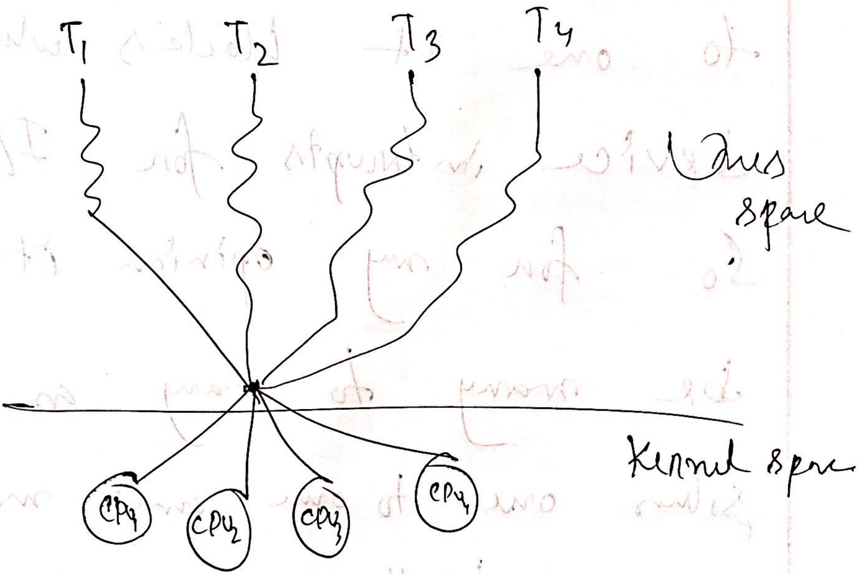
for non maskable it will not check if IE=1 always flag on

Ans. to the q.n. 4



In the new state the process is being created. Then it ~~will be~~ is in the job queue. After that it will go into the ready state which is the ready queue. In this case process is ready to be executed to CPU. When the software is being calculated it is executing in the running State. If any input is needed the device driver code send to process to I/O device queue. from there process take input according to the priority. After taking input the process go. to the ready queue. Then it again goes to CPU and after the computation it will terminate.

Ans. to the q.n.5



many to many level. Here one or more I/O operations can be done at a time. in one-to one model 1 thread can have I/O device at a time but it cannot have more than one model at a time; For many to one it blocks when a device interrupts for I/O operation So for my opinion it will be many to many as to solve one to one and many to one restrictions and user can creat as many thread as possible and I/O interruption. So many to many will be more efficient in this regard.