Mid 1

Name : Munem Shahriar

ID : 2020-1-60-156

course title : operating system

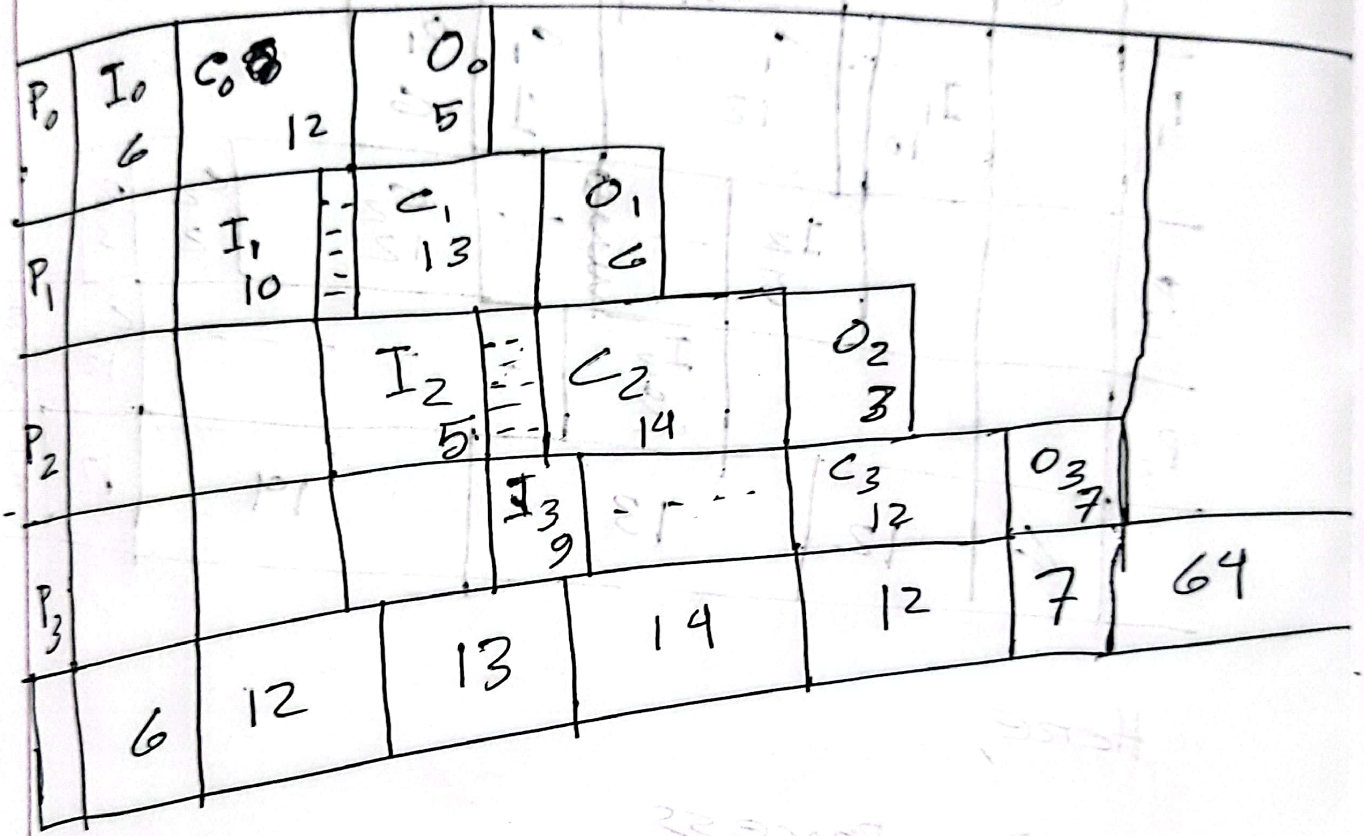
course code : CSE 325

section : 01

Roll no : 27

Ans to the Q no 2

(a)



Here,

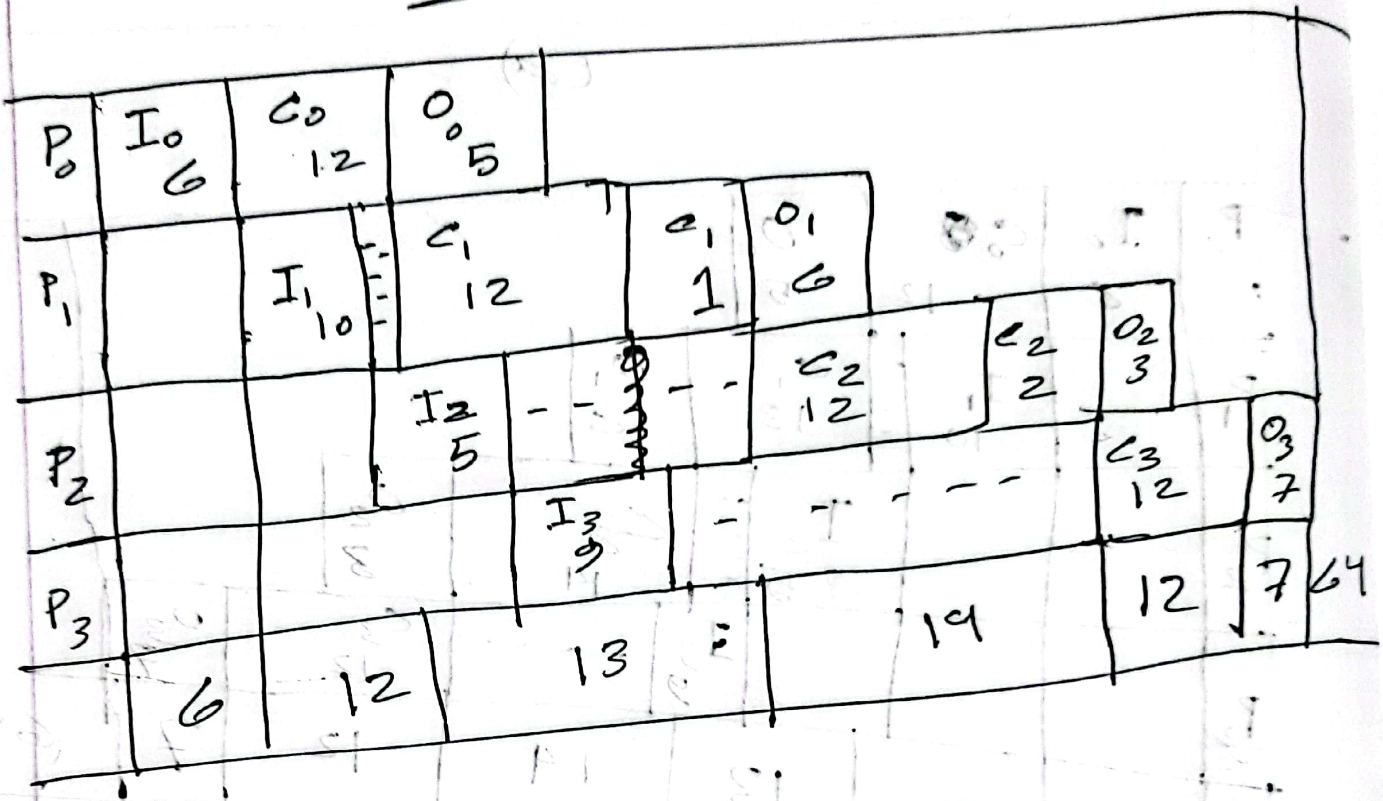
P = process

I = Input

C = computation

O = output

(b)



Here,

P = Process

I = Input

C = Computation

O = Output

Ans to the Q no 3

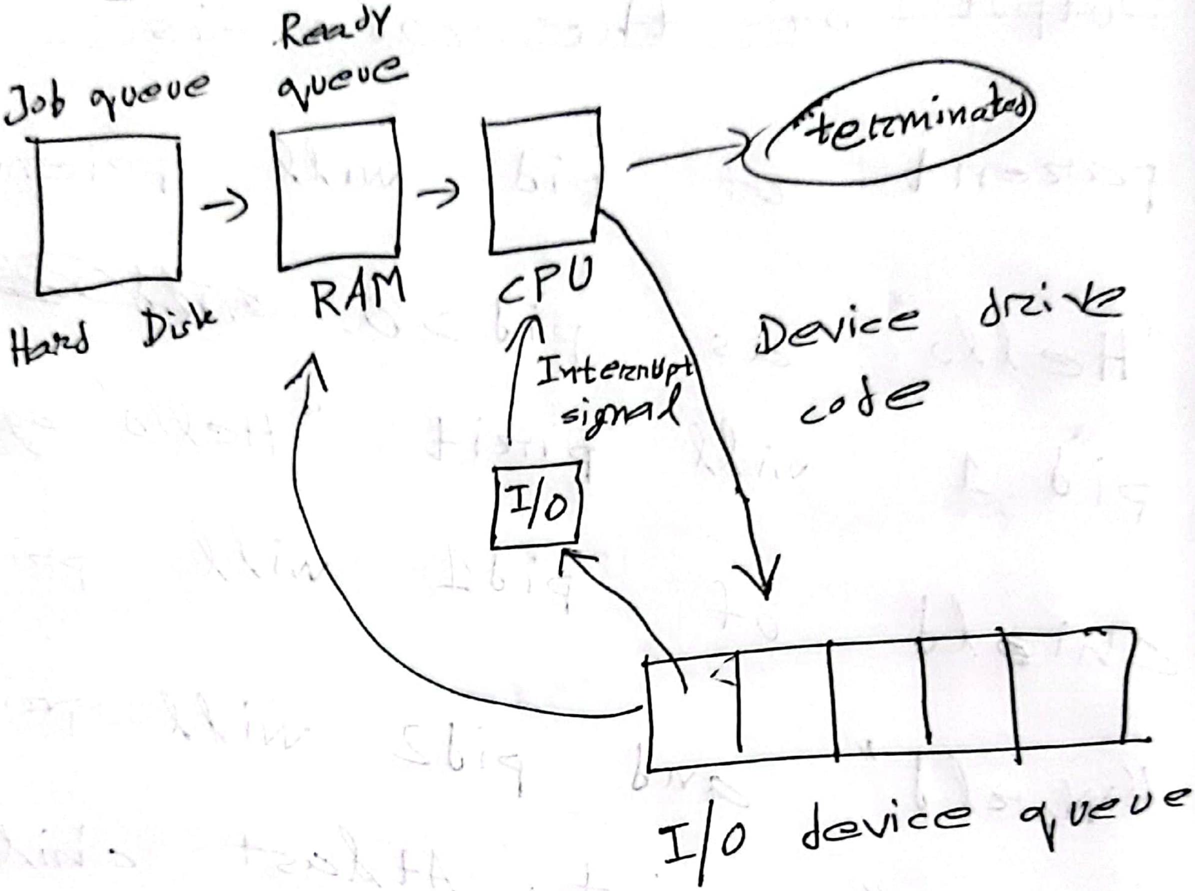
~~output of the code is:~~

parent of pid will priend "Hello" as pid > 0. pid 1 will pitrit "Hello" again. child of pid1 will print "World" and pid2 will print "World" againt. Atlast child of pid will print "EWU".

So the output of the code is:

```
Hello
Hello
World
World
EWU
```

Ans to the Q no 1



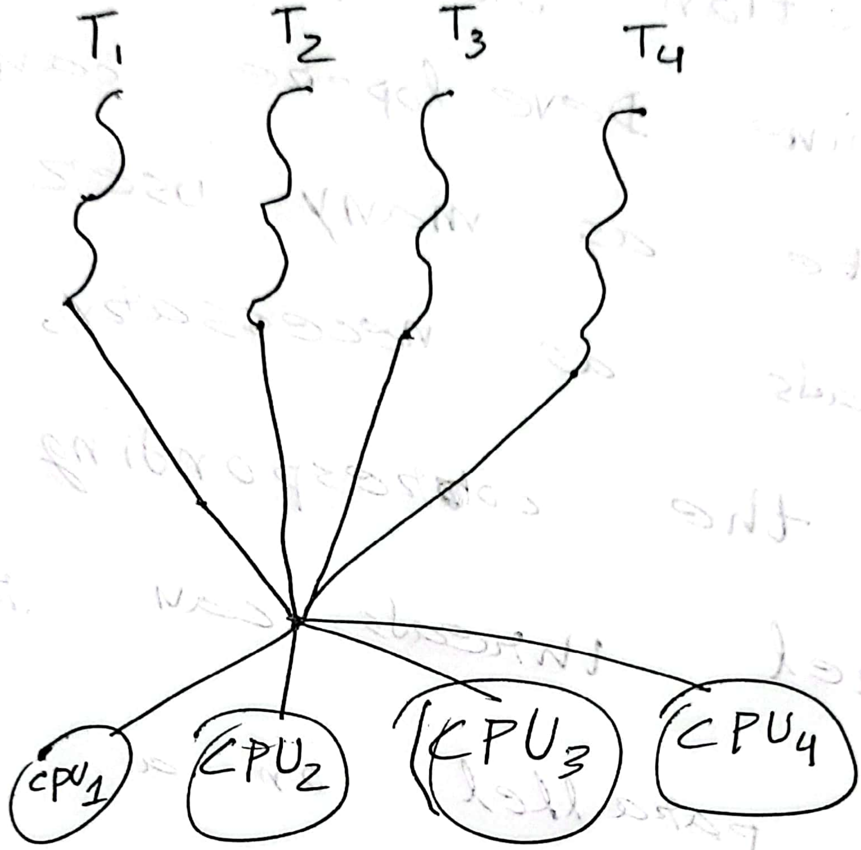
At first all the process are in Hard disk. Job ~~sche~~ scheduler will send some selected files to RAM. Then CPU ~~sh~~ scheduler will send one by one process to CPU.

if any process face interrupt for I/O or any other reason device ~~dive~~ drive code will send it to I/O device queue, and the next process will go to CPU, Otherwise the process will terminated by exit. According to queue priority process will take input. After taking input interrupt signal to CPU. Then CPU will save state to Process control Block. After that Process 1 will go to ready queue from I/O device queue. ~~All~~ It will continue like avobe untli all the process are terminated

Ans to the Q no 4

It is a input bound process. Here the process needs to take input meny times. For taking input the device driver code sends to process to I/O device queue. From there process take input according to ~~pri~~ piority. After taking ~~int~~ input the process go to ready queue. Then it againg goes to CPU and after computation it terminated by exit.

Ans to the Q no 5



Multiplex many user-level threads to a smaller or equal number of kenel threads.

The ~~num~~ number of kenex threads may be specific to either a prticqular application or particular machine. Developer can create as many user threads as necessary, and the corresponding kernel threads can run in parallel on a multi processor.

~~b~~Also when a thread prfome a blocking system call, the kerel can schedule anther thread for execution.