Final Exam

Name: Naitik Malav
Roll No: CS19BTECH11026

Ans6. *Server programs* such as *database and web servers* repeatedly execute requests from multiple clients and these are oriented around processing a large number of short tasks. So we can use thread pooling concept for building a server application because such applications would create a new thread each time a request arrives and service this new request into newly created thread.
As we know that, thread pooling is basically a software design pattern for achieving concurrency in computer program in which we create a number of threads at start-up and place them into a pool, where they sit and wait for work, so this concept would be mostly helpful in server programs.
Also by supervising program, it maintains multiple threads waiting for tasks to be allocated for concurrent execution. And due to frequent creation and destruction of threads, thread pools also helps in maintaining latency in execution.

Ans5. According to Amdahl's Law:

Speedup <= 1/(S + (1-S)/N)

a) 4 processors, i.e. N = 4, S=0.20

Max value of speedup is = max(speedup) = 2.5

b) 8 processors, i.e. N = 8, S=0.20

Max(speedup) = 10/3 = 3.333(nearly)

Ans3. Given that it has 64bit instructstions composed of two bit fields. The first 4 bytes contain the opcode, and the remainder an immediate operand or an operand address, so the remaining bits in 64-bit instructions are:

64-(4*8) = 32

So the maximum directly addressable memory capacity is 2^(32) bits.

Ans1. JOB1: CPU time=3ms, execution time=23, other=20ms

JOB2: CPU time=5ms, execution time=29, other=24ms

JOB3: CPU time=4ms, execution time=14, other=10ms

So in case of *uniprogramming:*

CPU utilization = (23+29+14)/(3+5+4) = 66/12 = 5.5

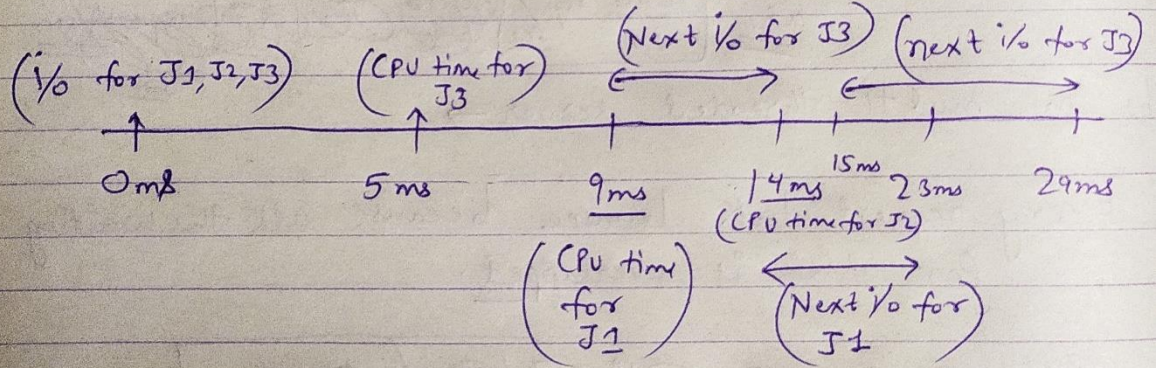
In case of multiprogramming:

Refer below image.

	CPU Time	i/o time
Job 1 (J_1)	3	20
Job 2 (J_2)	5	24
Job 3 (J_3)	4	10

In this case CPU time is → 12 ms

and Execution time can be calculated as →



∴ total execution time is 29 ms

∴ CPU Utilization = $\frac{29}{12} = 2.4166$ as

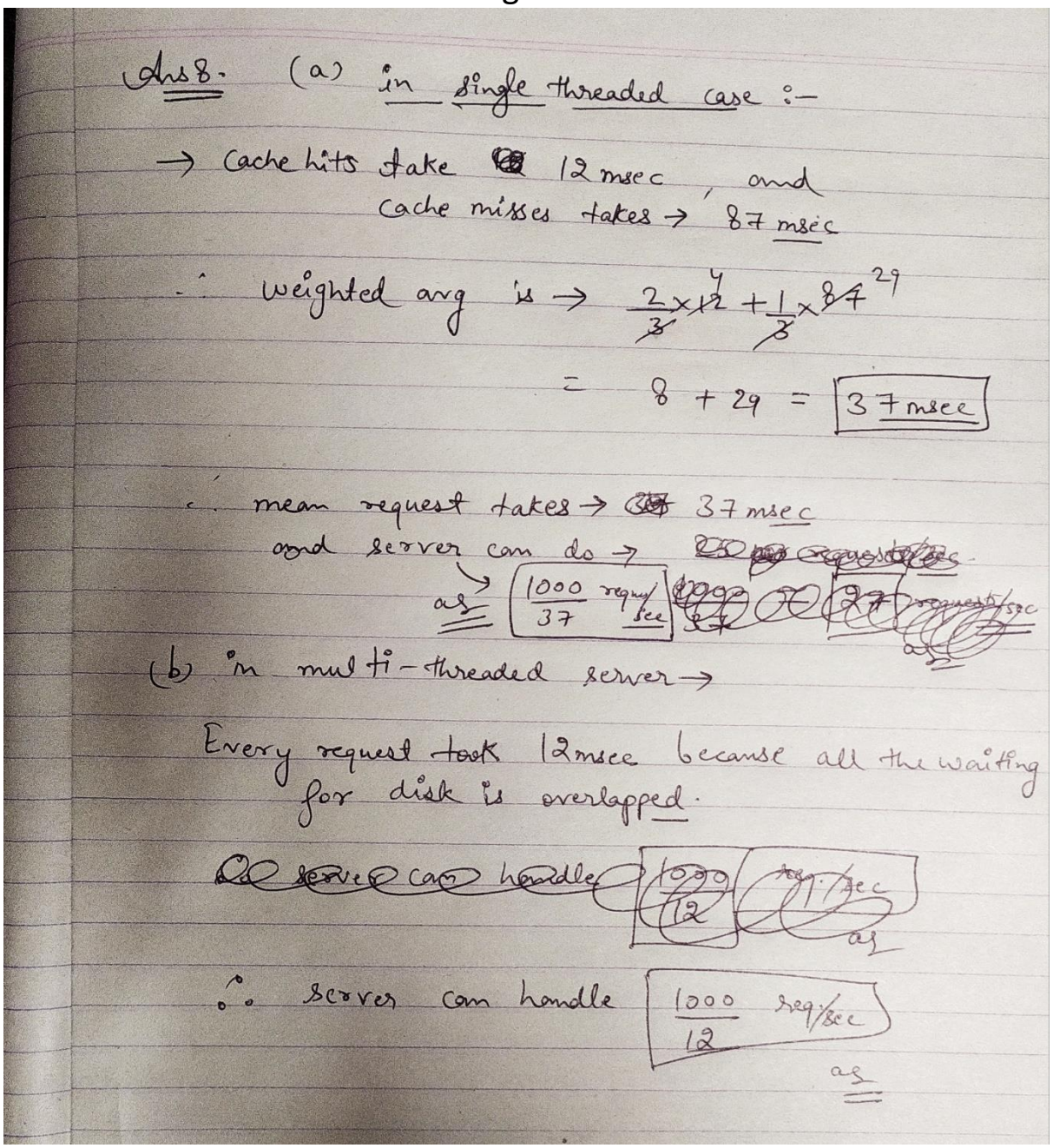
Ans4: a) Yes, it is possible that you would want to allow a process to wait on more than one event at the same time. For example, if a process may need to transfer data from one device to another, then in this case it may request both of the devices at same time and can wait until both are available for use.

Also you can take network applications as another example, like it may wait on multiple sockets until data packets arrive at any of them.

b) As we already know that *process, event* and *process-event pair* is needed in event queue model, which is defined for event queue nodes. A chain may be constructed to track all the events that a process is expecting, to enable a process to wait on multiple events at the same time. Also the nodes in the event queues need additional fields for the event chains they belonged to.

Ans7. Yes it is possible that algorithm that performs several independent calculations concurrently (e.g., matrix multiplication) be more efficient if it it did not use threads. This problem can be divided so if we use multiple threads we can utilize our CPU more and more efficient.

Ans8. Please refer the below image.



Ans2. A) The teletype is able to encode an alphanumerical symbol to an 8-bit word and decode into an alphanumerical symbol. A simple keyboard and printer teletype can be controlled by i/o module. The CPU contains 4 registers that are connected to system bus.
Input is stored into input register INTR. It will only accept data from teletype when input flag FGI is zero. Then data stored in INTR, and FGI is setted to ONE, and later CPU transfers the contents of accumulator(AC) and set back FGI to ZERO.

If CPU wants to send data to teletype it checks o/p flag FGO. If FGO=0, the CPU must wait. If FGO=1 then CPU transfers contents from AC to o/p register OUTR and sets FGO=0. The teletype set FGI=1 after word is printed.

B) The CPU is much faster than teletype and it must repeatedly check FGI and FGO. If interrupts are used, teletype can issue an interrupt to CPU whenever it wants to accept or send data. The IEN register can be set by the CPU under program control.