Q.1) Write a note on Semaphores.

Ans Semaphore is a synchronization tool. Semaphore is a variable 's' which can be accessed through only two standard atomic operations called wait() & signal().

The syntax of wait() is,

```
wait(s) {
    while s<=0
        ; // no operation
    s--;
}
```

The syntax of signal() is,

```
signal(s) {
    s++;
}
```

There are 2 types of semaphores.

i) Counting Semaphores.
ii) Binary Semaphores.

Counting semaphores has unrestricted domains whereas, binary semaphores ranges between 0 & 1. Binary semaphores is also called as mutex lock.

Q2) Write a note on critical section problem & a solution to solve it.

Ans Each process has a segment of code called "critical section" no two processes can execute in their critical section simultaneously. These is called critical section problem.

To solve this problem the processes should co-operate using a protocol i.e., every process request permission to enter into critical section this is called "entry section". & than comes "critical section", than at last come "exit section".

The following is the general syntax of a process,

```
do{
    entry section
        critical section
    exit section
        Remainder section;
}while(TRUE);
```

The solution for critical section problem is to satisfy 3 requirements;

i; **Mutual Exclusion:** If a process is executing in its critical section than no other process can enter in their critical section.

iii) **progress:** If no process is executing in critical section, than the other process should take decision on which will enter in critical section.

iii) **Bounded Wait:**

There exists a limit or bound on a number of ~~processes~~ times a process can enter in critical section.

~~Q3)~~ What is deadlock? What are necessary conditions for deadlock to occur.

Ans A situation where a process make requests for resource holding some other resources with it & the requested resources are held by some other processes.

There are 4 necessary condition for deadlock to occur:

1) **Mutual Exclusion:**

Atleast one resource should be in non-shareable mode i.e, only one process can use the resource at a time

2) **Hold & Wait:**

The process will hold atleast one resource & waiting for acquire additional resources which are currently being held by other process.

3) **No-premption:**

Resources cannot be prempted. Resources can be released by a process voluntarily only it completes its task.

4) **Circular Wait:**

A set $(P_0, P_1, P_2 - - - P_n)$ of waiting processes in such a way that p_0 is waiting for the resource which has been held by P_1 & P_1 is waiting for P_2 & so on

FO

Ajay
Page No: 06
Date / / 20

Q45 Explain how to recover from deadlock.

Ans There are 2 options to recover from deadlock:

i. process Termination.
ii. Resource preemption.

i. Process Termination:

Here, the process are aborted to eliminate the deadlock.

Again there are 2 ways to abort a process.

i. Abort all the deadlocked processes:

here, all the processes are aborted to eliminate the deadlock.

ii. Abort one process at a time.

A single process is aborted one by one until the deadlock is eliminated. Before aborting a

process first check process priority & number of time a process executed.

→ **Resource premption:**

It has 3 approaches,

i) Selecting a victim: which resource & process are to be prempted are selected here,

ii) Roll back: After premption, the process starts execution from the begining.

iii) Starvation: Same process & resource should not be preempted everytime.

Q5) Explain the Banker's algorithm?

Ans This algorithm avoids the deadlock. This algorithm make use of following data structures.

1. **Available:** It indicates the number of resources available. It is a vector variable.

2. **Max:** It indicates the maximum demand of each process.

3. **Allocation:** It is a matrix that defines the number of resource for each process allocated currently.

4. **Need:** It indicates the remaining need of each process.

This algorithm is called banker's algorithm because it is mainly used in banks to keep the bank in safe state.

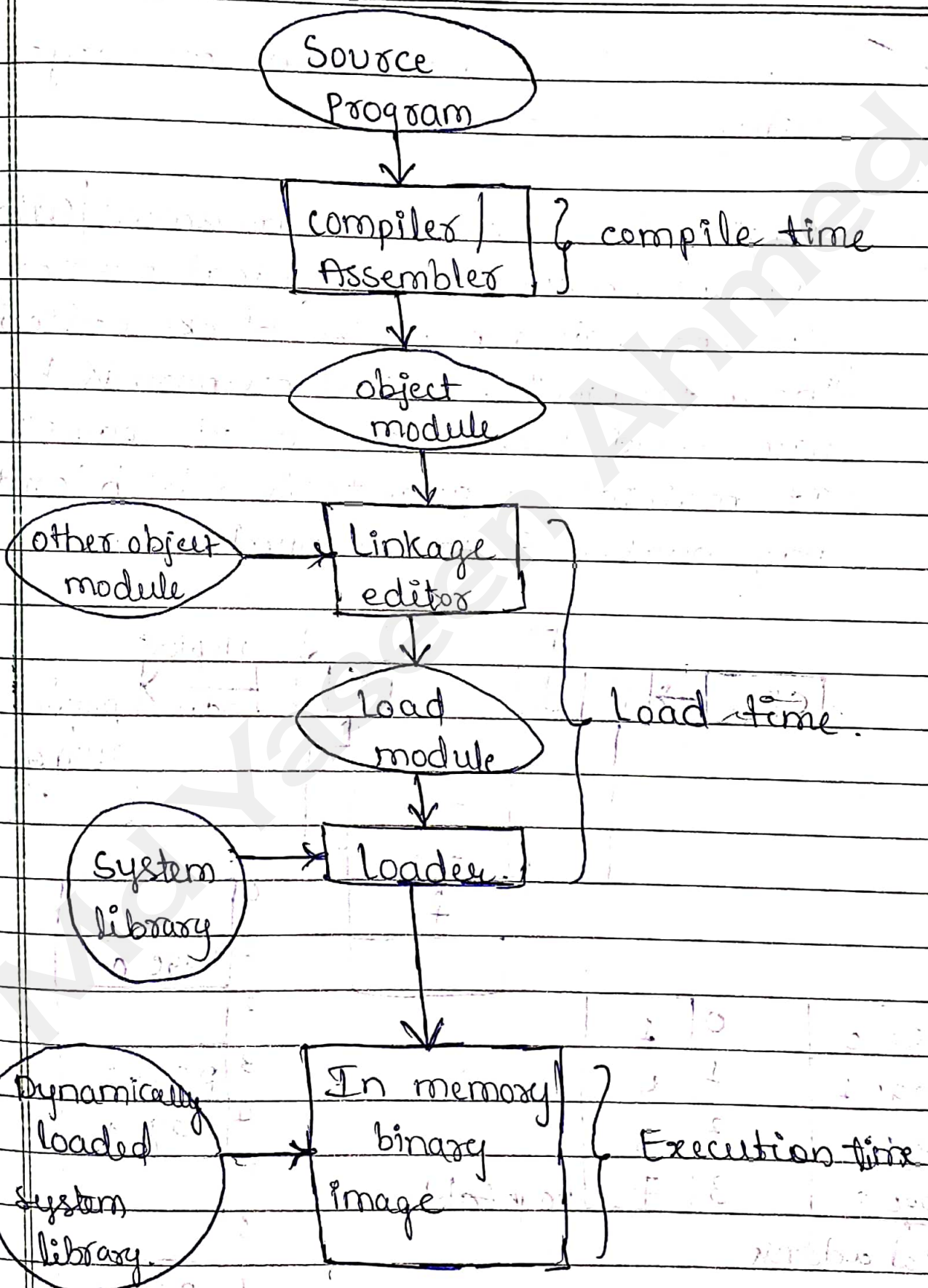
Q6) Explain address binding with a neat diagram.

Ans Assigning the address of physical main memory to the source code is termed as address binding. The address binding can happen at one of three stages.

i) **Compile time:**
If the compiler knowns where the source code resides in the physical main memory then generates absolute address otherwise (the binding is delayed till load time) x

ii) **Load time:** If the compiler does not known where the source code resides in the memory then generates a relocatable address & binding is delay till load time.

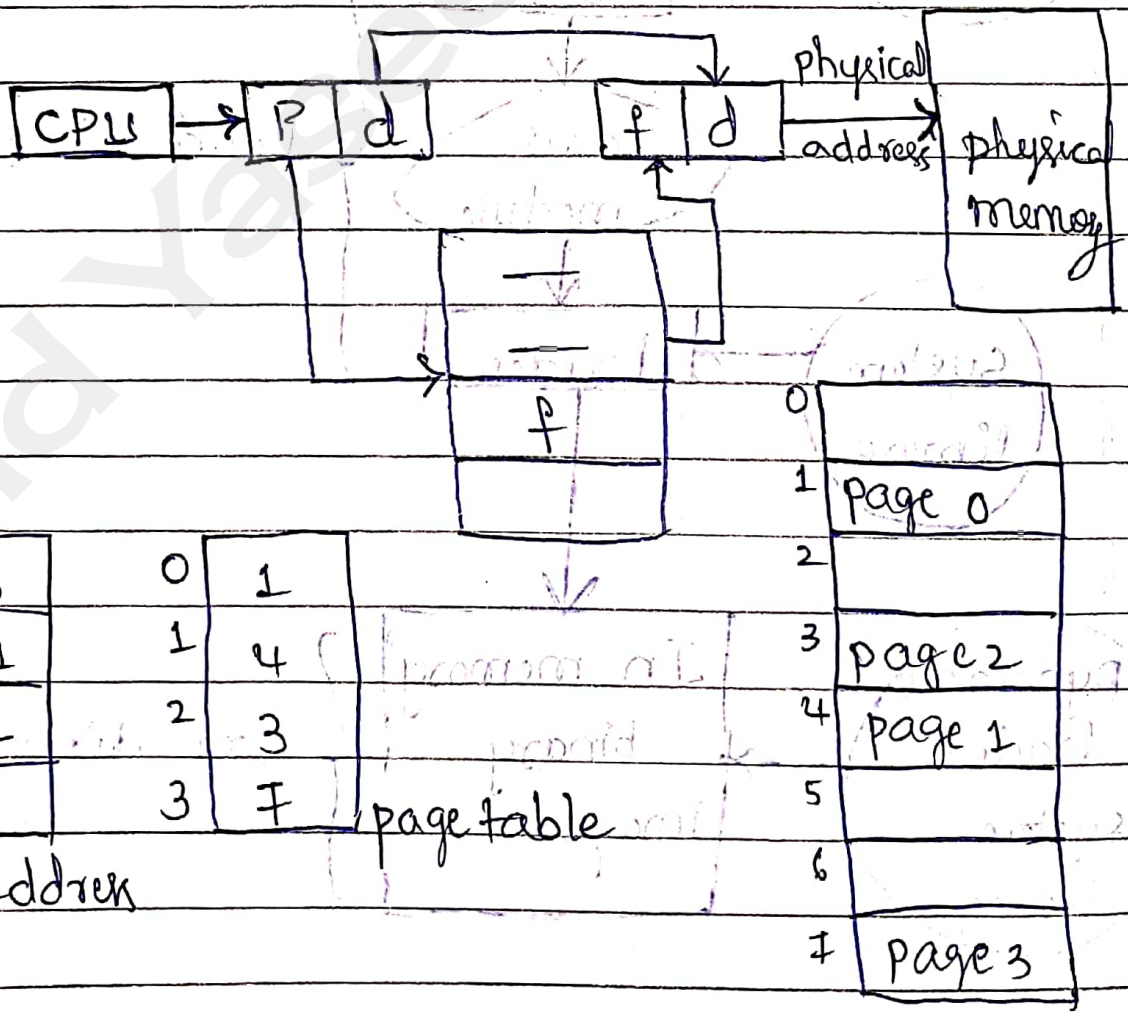
iii) **Execution time:** If a process can be moved during its execution from one memory segment to another, then the binding must be delayed till run time.



Ajay
Page No: 11
Date / / 20

Q.1 Explain basic concept of memory allocation using paging.

Ans paging is a memory management allocation scheme, where the physical memory is divided into fixed size blocks called "page frames" & logical memory is breaked into fixed size blocks called "pages". It maintain a page table which contain page number & frame number.



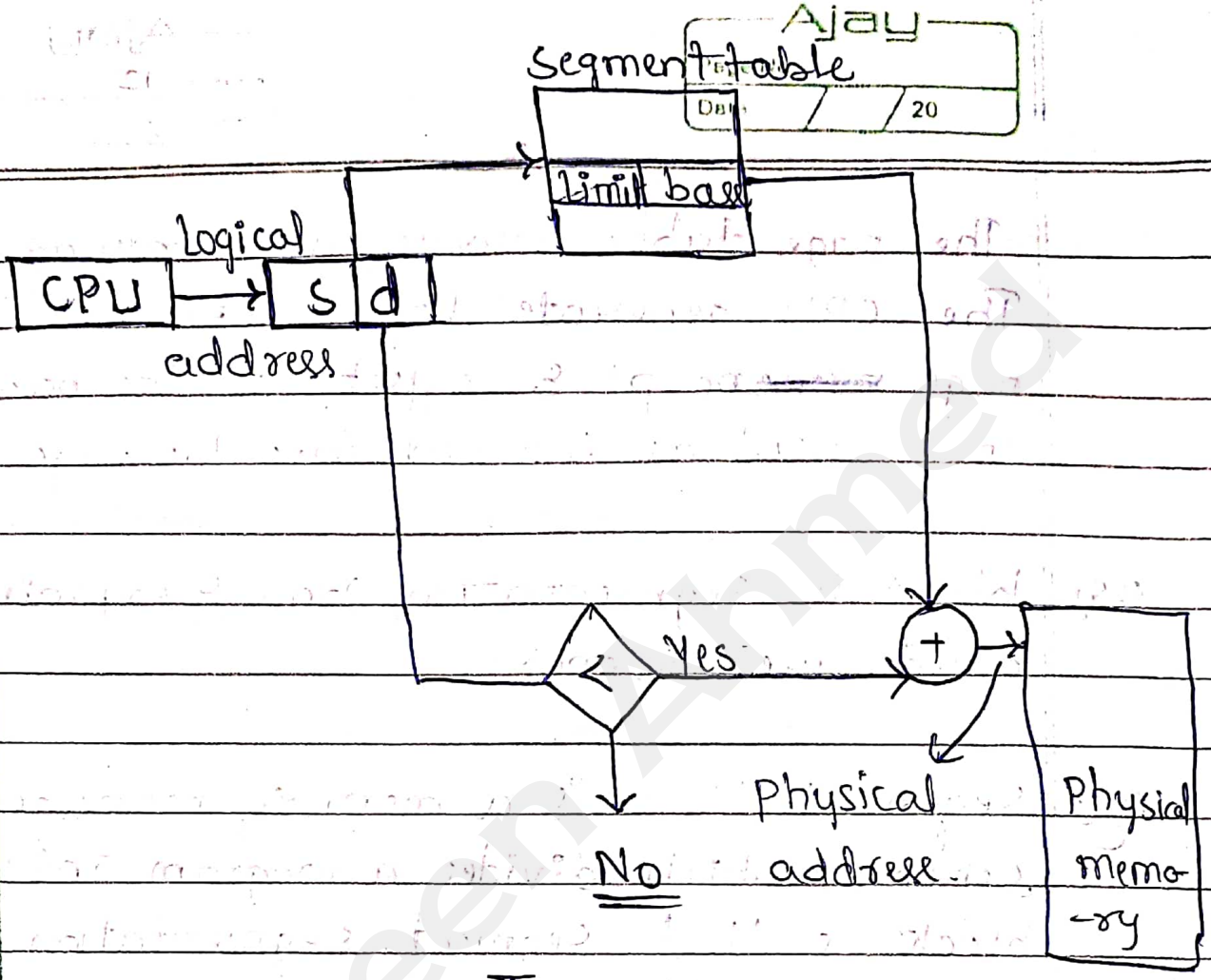
The page table serveres as hardware support. The CPU generates logical address which contain page ~~name~~ no 'p' & offset 'd'. The page number 'p' is used as indexed into the page table

Q8 What is segmentation. Draw & explain its hardware support.

Ans Segmentation is a memory management scheme, which divides a program into number of blocks called segment. segmentation is a variable size. It reduces (eliminates) internal fragmentation In segmentation the logical address space contain 2 parts:

i. segment number &
ii. offset

The figure shows the hardware support for segmentation. The segment table acts as hardware support. The segment table contain segment base & segment limit. The segment base contain starting physical address of the segment & the limit specifies the length of segment.



→ The logical address contain 2 parts : segment number 's' & offset of segment 'd'. The segment number is used as index for segment table & the offset must be between 0 & limit. Then it is added to segment base to produce the physical address.

Ajay
Page No:
Date / / 20

Q9. Explain First Fit, Best Fit & Worst Fit memory allocation strategies.

Ans **First Fit:** It allocates the first hole which is big enough. Searching is started either for the begining of set of holes, or from where the previous first fit search has ended. We can stop search as soon as we find the hole that is large enough.

Best Fit: It allocates the smallest hole which is big enough. We can search the entire list unless the list is kept in order by size. This strategy produces smallest leftover hole

Worst Fit: It allocates the largest hole which is big enough. We can search the entire list until the list is sorted by size. This strategy produces largest leftover hole which is more useful than Smallest leftover hole of Best fit strategy.