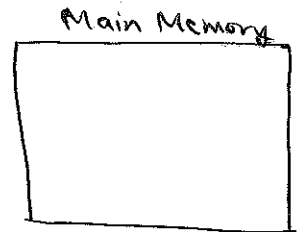


# Memory Management

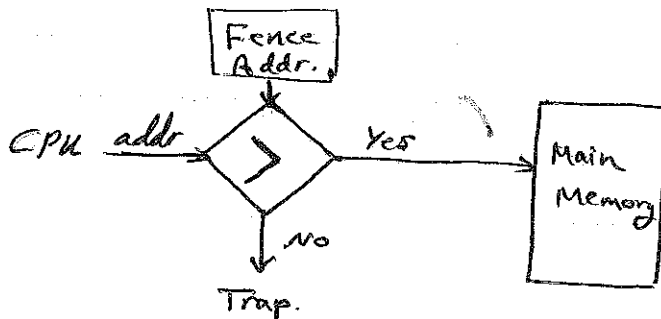
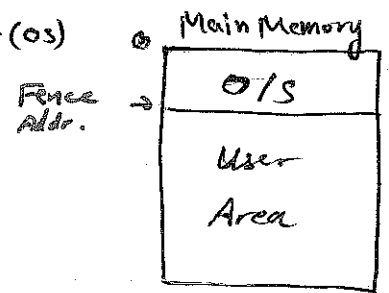
- Bare Machine: - No preloaded operating system.
  - The designer has to decide how to manage memory



Bare Machine  
(dedicated system)

## Resident Monitor:

- Part of the main memory is reserved for OS.
- We need a protection of the monitor part (OS) against any user program.
- Fence address is the boundary between the user area and the monitor area.



The process will be terminated because it is trying to access a memory which is not legal.

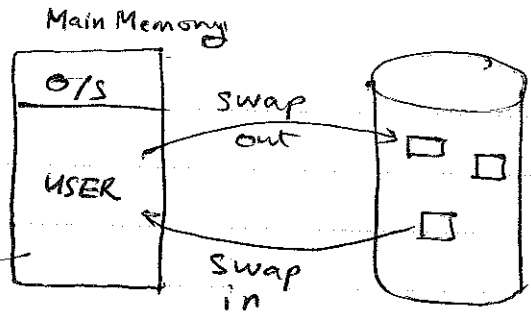
- Two options for implementing the above protection technique
  - Software: flexible but slow
  - Hardware: fast but fixed (cannot upgrade OS)
- Special purpose register called fence register. So whatever is the content of the fence register that is the fence address.
  - The fence register must be changed by some privileged instruction.

## Swapping:

### (1) Single User partition

- There is one program in the main memory in the user area.

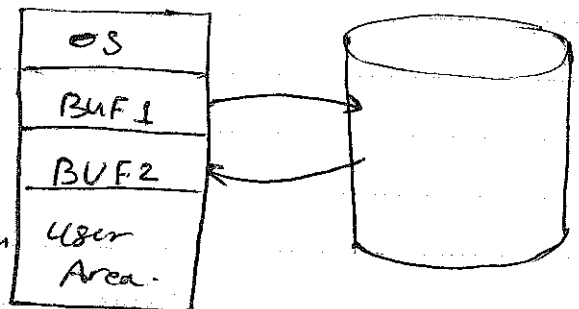
Single user partition.



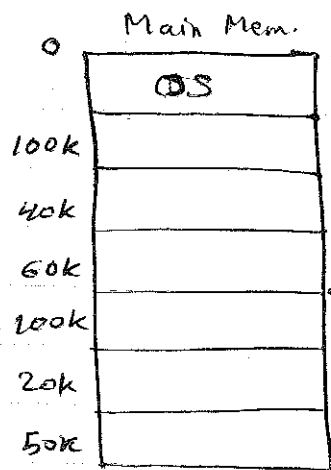
- the swap operation is very slow compared to main memory memory operation.

### (2) Multiple User partitions.

- Speed up the processing by allowing the swaps overlap with CPU execution.

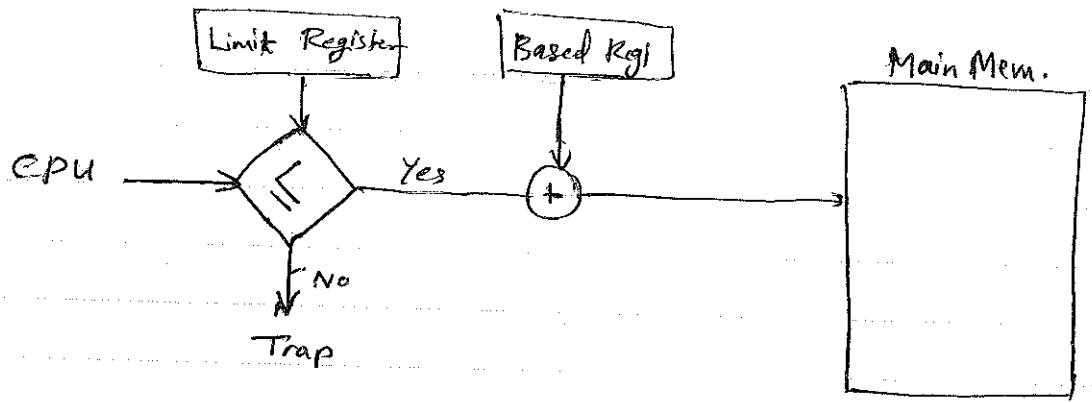


## MFT: Multiprogramming with Fixed number of Tasks



To protect the processes from each other  
 Based Register } Task  
 Limit Register }

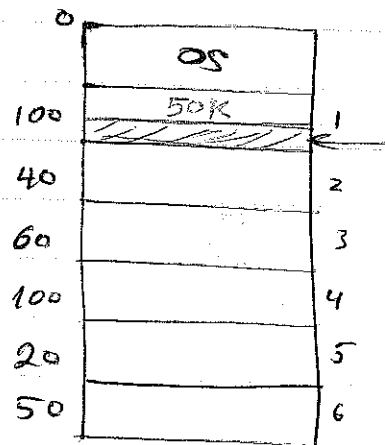
- The main memory is divided to many partitions (with different or same size)
- The size of every partition is fixed
- The number of multiple processes depends on the number of partitions.
- The system have to maintain information about partitions.
  - (1) - The size of the partitions
  - (2) - The status of the partitions
  - (3) - The starting location of the partitions



- To put a new process into a free partition (Hole)

- (1) First-fit
- (2) Best-fit
- (3) Worst-fit

- Assume that we have only partition 1 is free and we have a process which needs 50K byte of memory.



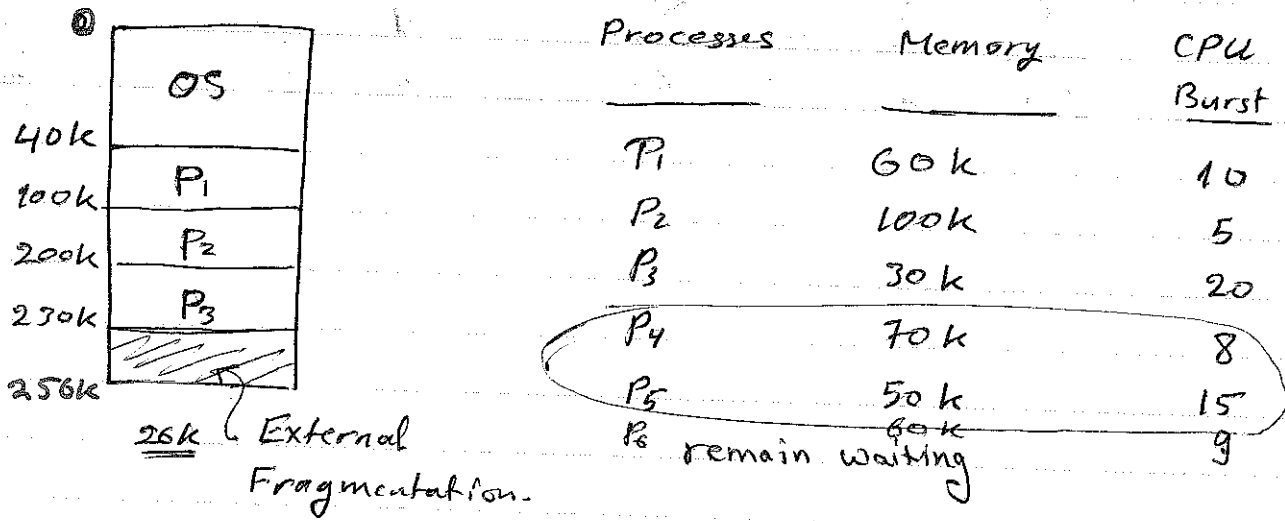
- 50K byte is wasted

memory because partition 1

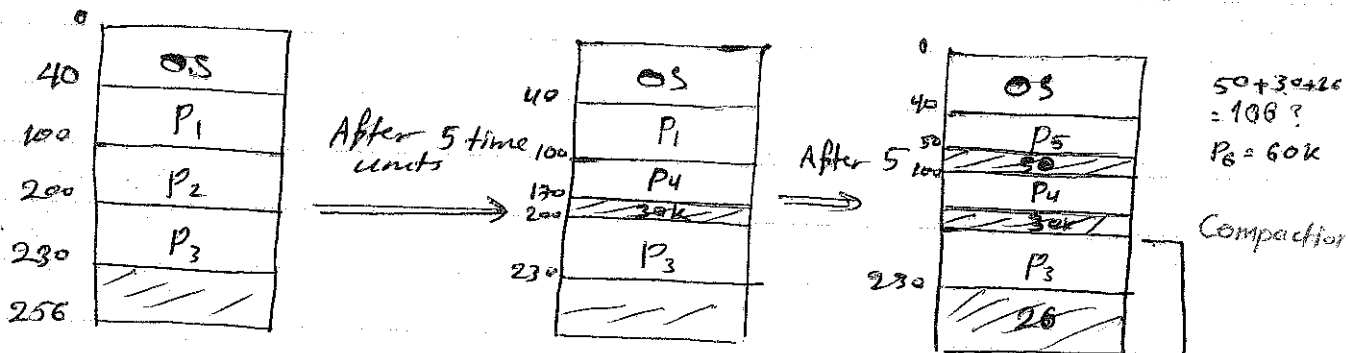
cannot be used by other process  $\Rightarrow$  Internal Fragmentation.

- Assume we have a process which needs 120K bytes, but no partition in the main memory with size 120KB or greater to accommodate this process  $\rightarrow$  External Fragmentation.

# MVT = Multiprogramming with Variable number of Tasks



- Assume the memory is 256 kbytes and 40 kbytes is occupied by the operating system.



- The External fragmentation can be minimized if we could use the compaction technique.

- Memory compaction can be used with MVT but not MFT.

- The CPU cannot execute any process during the compaction.

- The information need to be maintain

- (1) Allocation Table
- (2) Free Area table.