Operating Systems

Quiz

Q. Critical Section Problem can be resolved by using

A Binary Semaphore

B Mutex Object

C Classic Semaphore

D. Both A & B

E. None of the above

Q. Which of the following signal an OS send to a process for forcefull termination?

A. SIGTERM

B. SIGEND

C. SIGSTOP

D. SIGKILL

Quiz

Q Processes which shares data with another processes referred as

A related processes

B cooperative processes

C independent processes

D. all of the above

E. none of the above

Q. Which of the following ipc mechanism is used for communication across the systems?

A Pipe

B message queue

C. chatting application

D. socket

E. shared memory model

Memory

- Memory holds (digital) data or information.
  - Bit = Binary Digit (0 or 1) --> Internally it is an electronic circuit i.e. FlipFlop.
    1 Byte = 8 Bits
    B, KB (2^10), MB (2^20), GB (2^30), TB (2^40), PB (2^50), XB (2^60), ZB (2^70)
- Primary memory – Memory
  - Directly accessible by the CPU.

    E.g. CPU registers, Cache, RAM.
- Secondary memory -- Storage
  Memory accessible via Primary memory. E.g. Disk, CD/DVD, Tape, ROM.
- Volatile vs Non-volatile memory
  - Volatile memory: The contents of memory are lost when power is OFF.
    Non-volatile memory: The contents of memory are retained even after power is OFF.

Memory Hierarchy

- Level 0: CPU Registers
- Level 1: L1 Cache
- Level 2: L2 Cache
- Level 3: RAM
- Level 4: Disk (Local)
- Level 5: Remote storage (NFS)
- Comparison
  - Capacity: Level 0 is smallest (B) to Level 5 is largest (TB)
  - Speed: Level 0 is fastest and Level 5 is slowest
  - Cost: Level 0 is costlier and Level 5 is cheaper

Memory Access

- CPU <----> Memory
- Address bus
  - Unidirectional from CPU to the memory
  - Address represent location of the memory to read/write
  - Number of lines = number of locations
- Data bus
  - Bi-directional from/to CPU to/from the memory
  - Carries the data
  - Number of lines = width of data unit
- Control bus
  - Read/Write operation
- CPU <---> Cache <---> RAM <---> Disk
- Sequential access: Read/write sequentially from start to end. e.g. Magnetic tapes

Direct access: Read/write to the block address e.g. Hard disk

RAM (Random Access Memory)

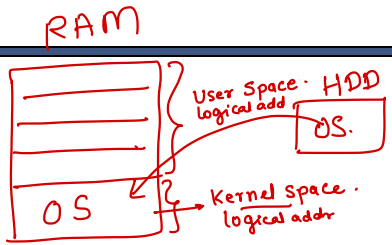
- RAM is packaged as a chip, Basic storage unit is a cell (one bit per cell)
- Internal memory of the CPU for storing data, program, and program result
- Used for Read/ Write
- Volatile (Temporary Storage)

Static RAM (SRAM)

- Retains its contents as long as power is being supplied.
- Made up of transistors.
- SRAM is more often used for system cache.
- SRAM is faster than DRAM.

Dynamic RAM (DRAM)

- Must be constantly refreshed or it will lose its contents.
- This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second.
- Made up of memory cells composed of capacitors and one transistor.
- DRAM is typically used as the main memory in computers.



ROM (Read Only Memory)

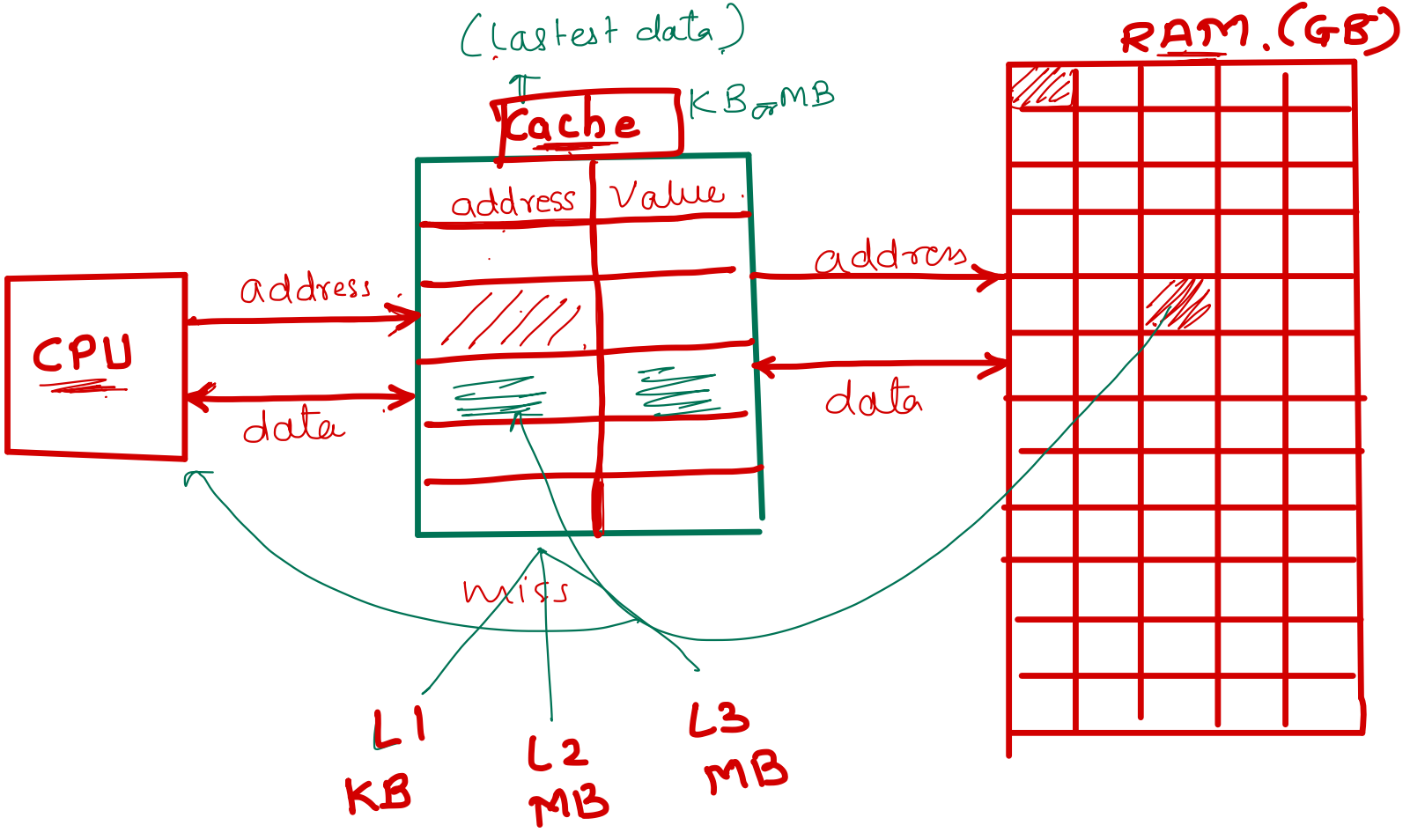
- Read-only memory (Not writable).
- This type of memory is non-volatile.
- The information is stored permanently.
- A ROM stores such instructions that are required to start (bootstrap) a computer.
- BIOS – Basic Input Output System
  - Set of programs stored in PC Base ROM (on Motherboard).
  - POST(Power On Self Test) – To test the peripherals.
  - Bootstrap Loader – To find OS in disk/usb/cd.com.
  - Basic/minimal device drivers for basic device functionality.
  - BIOS setup utility (F1 or ESC).
- Programs (executable instructions) stored in ROM are called -- Firmware. e.g. BIOS, Bootstrap loader, POST, ...

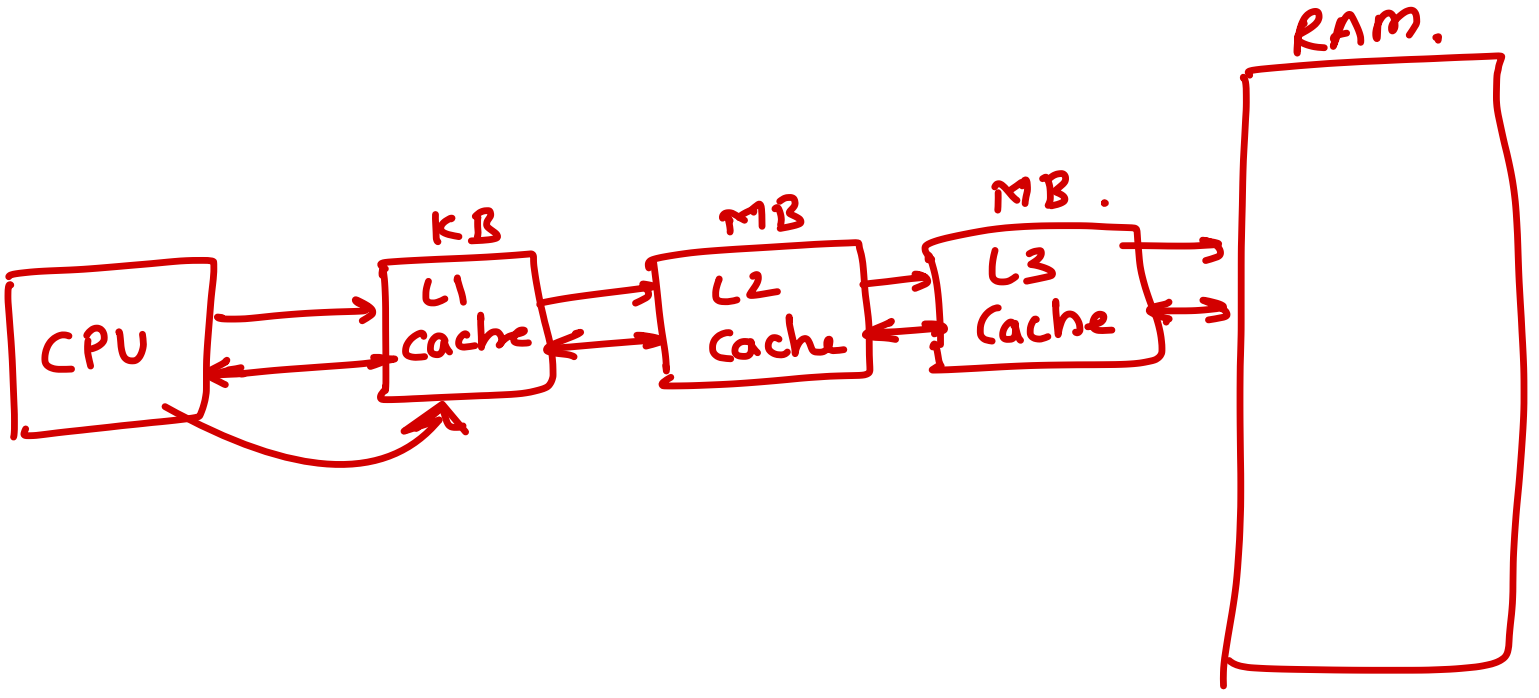
Types of ROM

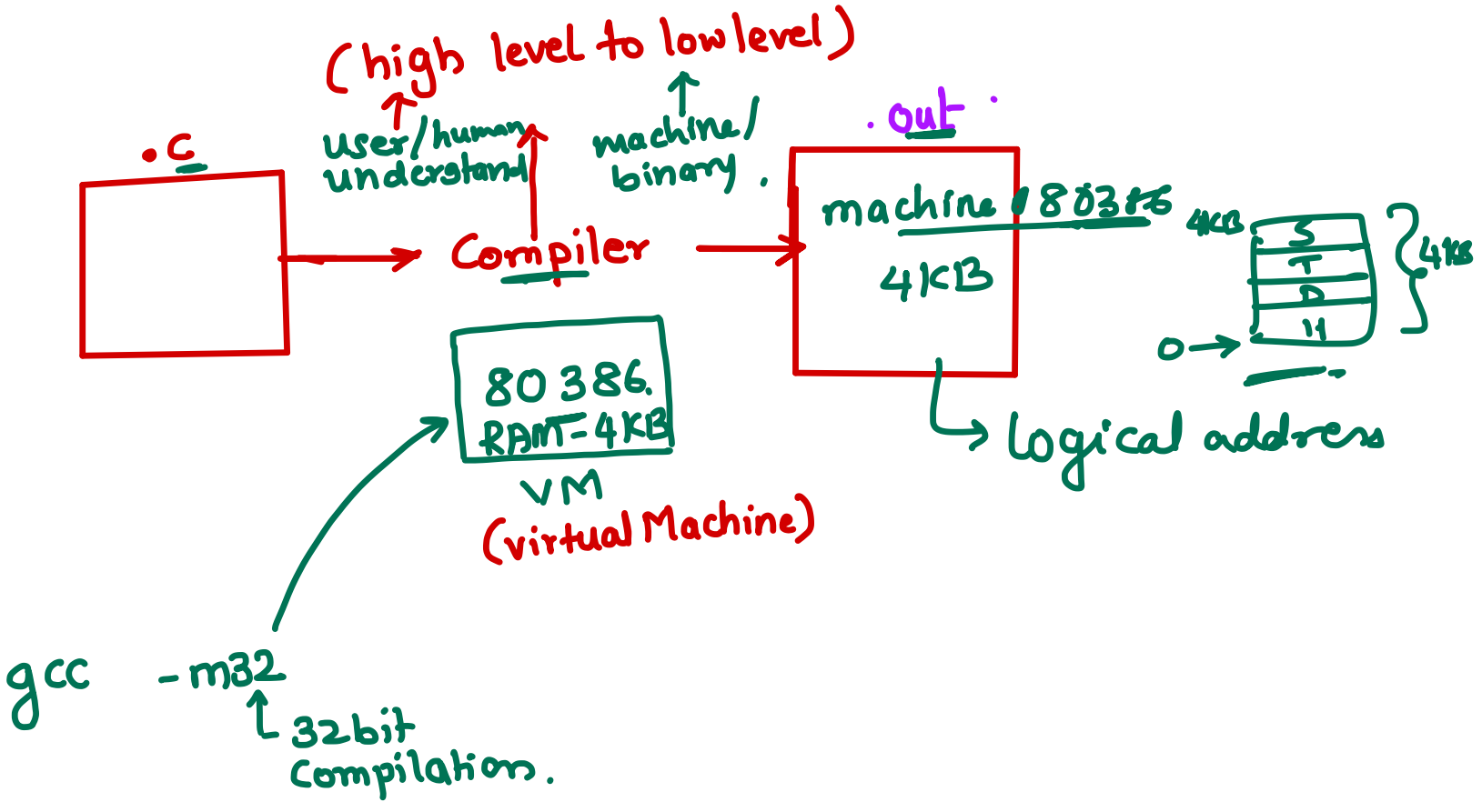
- Masked ROM (MROM) -- contents are fixed while manufacturing
- Programmable ROM (PROM) -- one time writable
- Erasable Programmable ROM (EPROM) -- written multiple times with special circuit
  - Ultra-Violet EPROM (UV-EPROM) -- all contents erased using UV rays
  - Electrical EPROM (E-EPROM) -- erase selected bytes using high electric current
- Flash (like E-EPROM) -- erase selected blocks - high speed

Cache memory

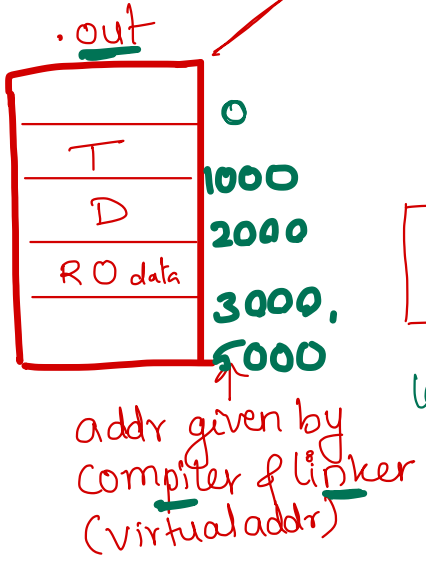
- Associative memory. Key = memory address, Value = memory contents.
- Made up of cache lines tagged by tag index.
- In CPU chip or outside CPU chip.
- If requested data is found in cache (cache hit), contents are sent from cache itself to CPU (faster access).
- If requested data is not found in cache (cache miss), contents are accessed from main memory, copied in cache and then sent to CPU (slower access).
- Cache memory size is limited (KB or MB).
- When cache is full, oldest data is overwritten (Least Recently Used).
- Cache always stores recently accessed data.

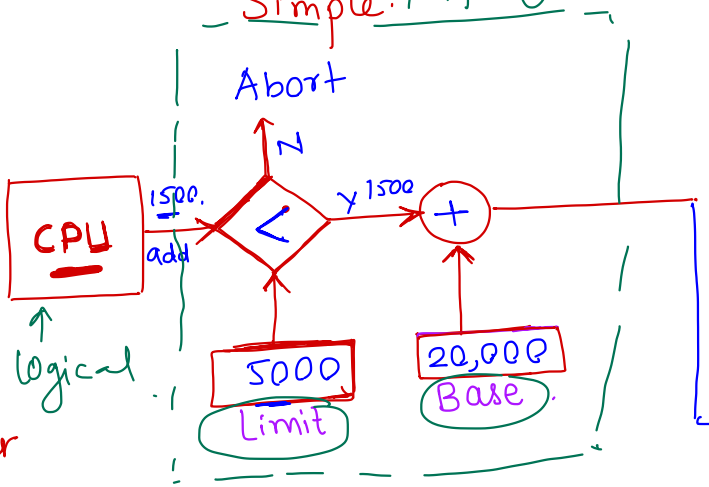

(Lastest data)
Cache
KB or MB
address
Value
CPU
address
data
address
data
RAM (GB)
miss
L1
KB
L2
MB
L3
MB

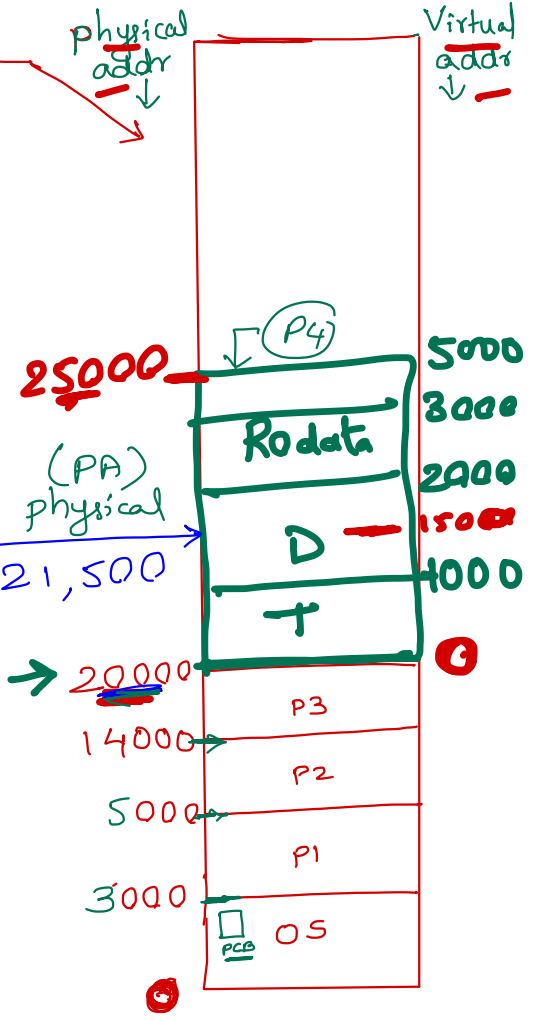

RAM.
KB
MB
MB.
CPU
L1
Cache
L2
Cache
L3
Cache


(high level to low level)
user/human understand
machine/ binary.
.c
Compiler
.out
machine 80386
4KB
4KB
S
T
D
11
0
80386.
RAM-4KB
VM
(virtual Machine)
logical address
gcc -m32
32bit
Compilation.

Loader





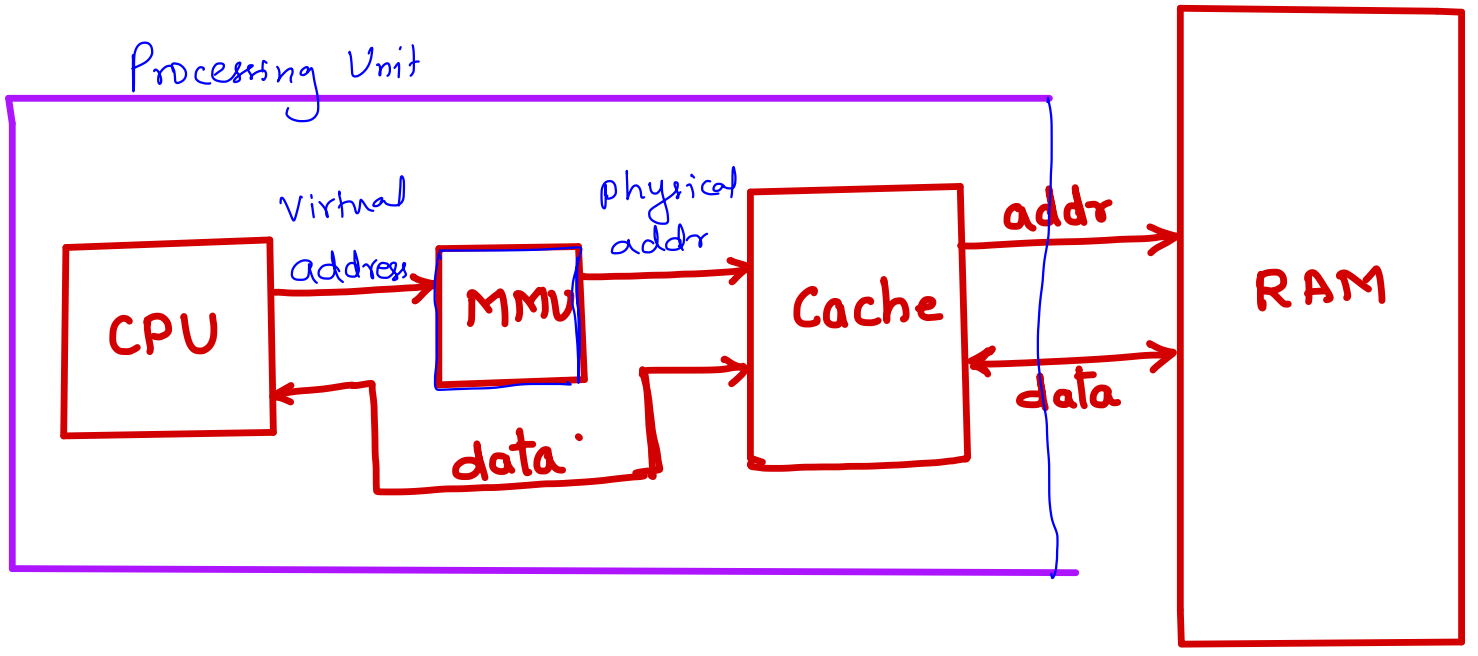


MMU ⇒ logic ⇒ physical.

MMU Hardware Unit.

1. **Simple MMU** → Continuous alloc.
2. Segmentation → seg alloc.
3. Paging → Page alloc

	Base	Limit
P3	14000	6000
P2	5000	9000
P1	3000	2000
P4	20000	5000


Processing Unit
CPU
Virtual address
MMU
physical addr
Cache
addr
data
data
RAM

Memory Management

- Compiler convert code from high level language to low level language.
- Compiler assumes a low config machine, while converting high level code to low level code called

as "Virtual Machine".

- Compiler and Linker assign addresses to each variable/instruction assuming that program will execute in VM RAM. These addressed are called as "virtual addr" or "logical addr". The set of virtual addresses used by the process is referred "Virtual address space".
- However while execution these addresses might be occupied by other processes. Loader relocates all instructions/variables to the address available in RAM. The actual addresses given to the process at runtime are called as "physical add" or "real add". The set of physical addresses used by the process is referred "Physical address space".
- CPU always executes a process in its virtual addr space i.e. CPU always request virtual addressed (on addr bus).
- These virtual addresses are verified and then converted into corresponding physical addresses by a special hardware unit called as "Memory Management Unit (MMU)".

Simple MMU holds physical base address and limit (length) of the process. The base & limit of each process is stored in its PCB and then loaded into MMU during context switch.

Memory Management

- RAM memory should be divided for multiple processes running concurrently.
- Memory Mgmt. scheme used by any OS depends on the MMU hardware used in the machine.
- There are three memory management schemes are available (as per MMU hardware).
  1. Contiguous Allocation
  2. Segmentation
  3. Paging

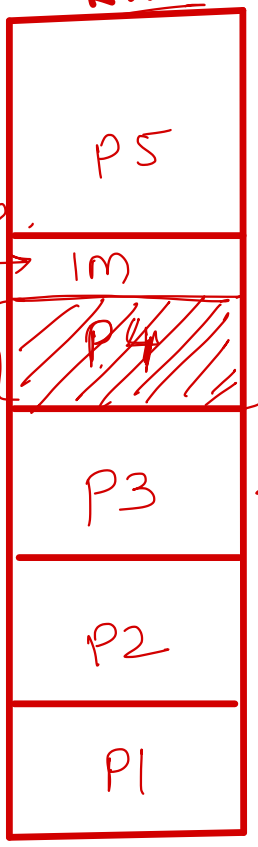
Contiguous Allocation

Fixed Partition

- RAM is divided into fixed sized partitions.
- This method is easy to implement.
- Number of processes are limited to number of partitions.
- Size of process is limited to size of partition.
- If process is not utilizing entire partition allocated to it, the remaining memory is wasted. This is called as "internal fragmentation".

Contiguous Allocation

① Fixed Partition



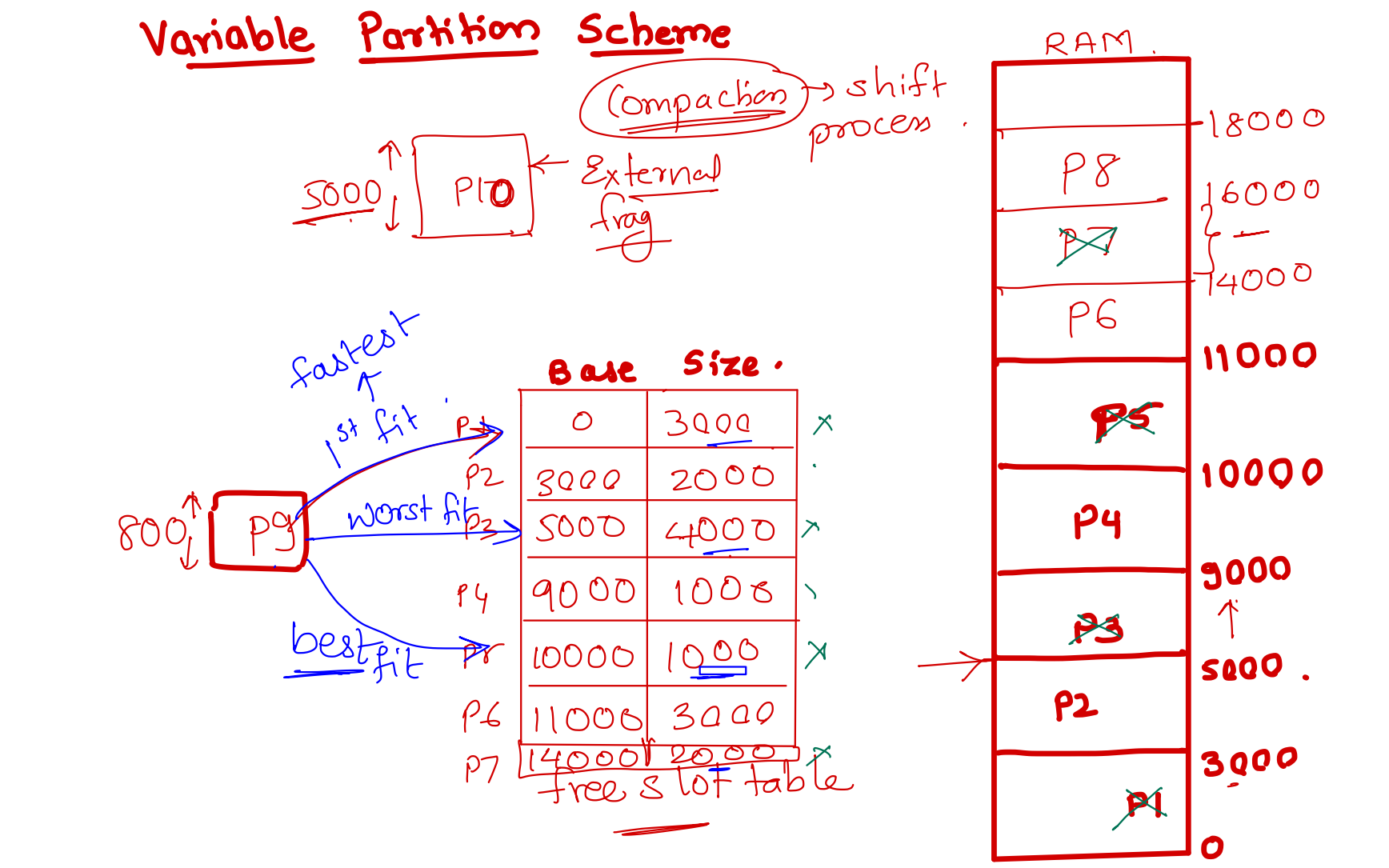
Internal fragmentation.
Wasted → 1M

2M

P4 → 2M

Limit.
→ partition – process run.
→ Internal fragmentation

Variable Partition Scheme

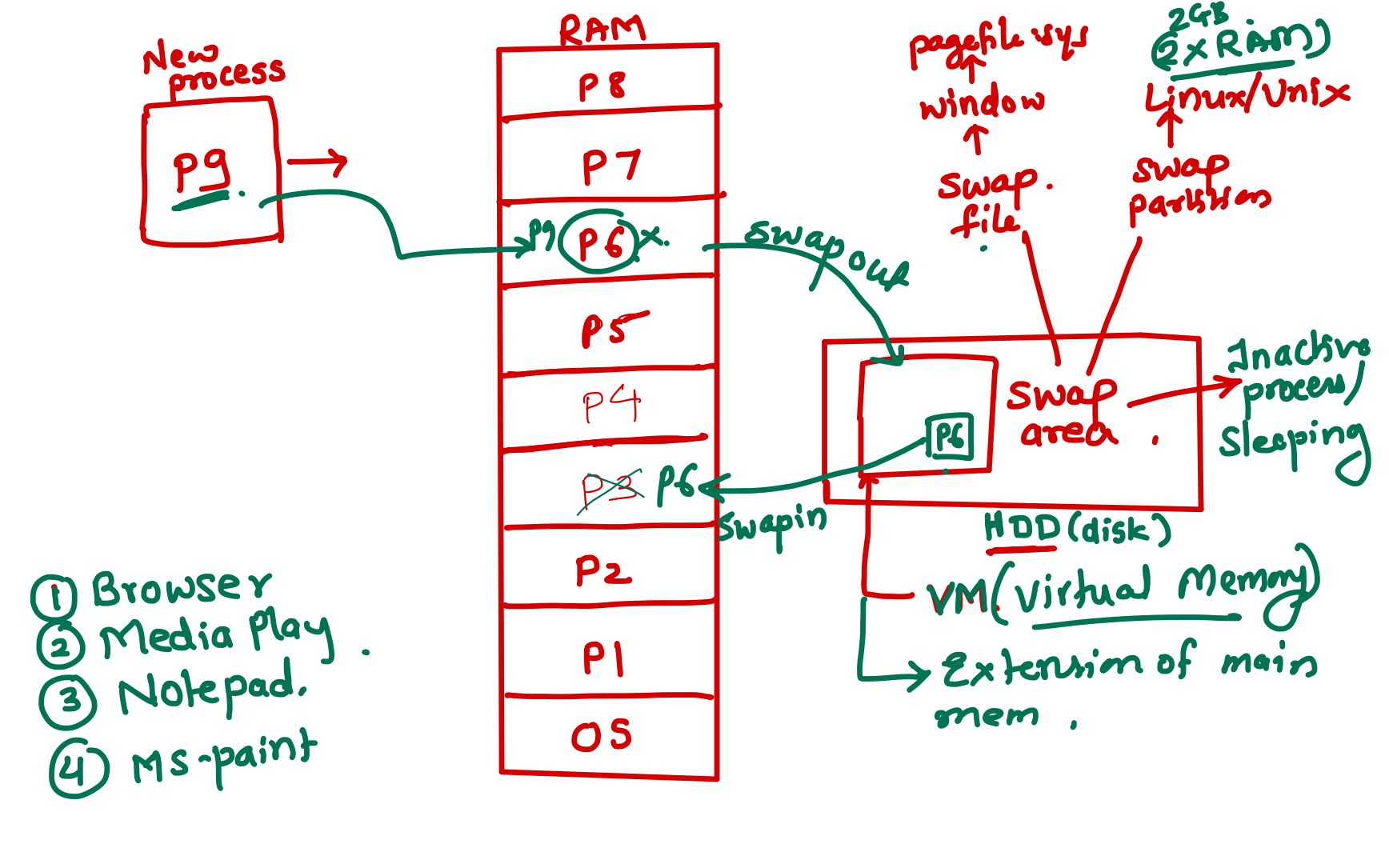


Dynamic Partition

- Memory is allocated to each process as per its availability in the RAM. After allocation and deallocation of few processes, RAM will have few used slots and few free slots.
- OS keep track of free slots in form of a table.
- For any new process, OS use one of the following mechanism to allocate the free slot.
  - First Fit: Allocate first free slot which can accommodate the process.
  - Best Fit: Allocate that free slot to the process in which minimum free space will remain.
  - Worst Fit: Allocate that free slot to the process in which maximum free space will remain.
- Statistically it is proven that First fit is faster algo; while best fit provides better memory utilization.
- Memory info (physical base address and size) of each process is stored in its PCB and will be loaded into MMU registers (base & limit) during context switch.

Memory Management

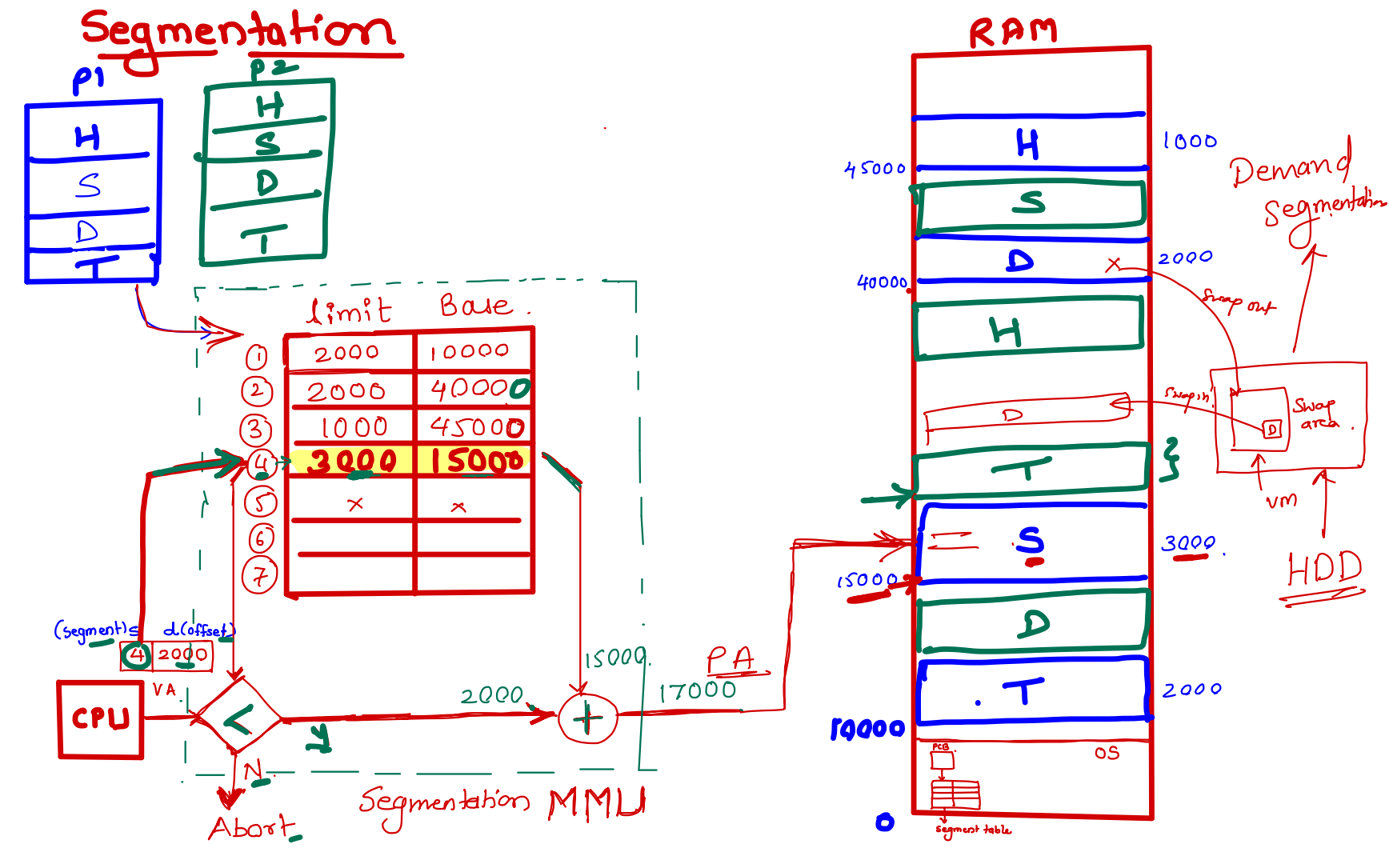
- CPU request virtual address (address of the process) and is converted into physical address by MMU as shown in diag.
- If invalid virtual address is requested by the CPU, process will be terminated.
- If amount of memory required for a process is available but not contiguous, then it is called as "external fragmentation".
- To resolve this problem, processes in memory can be shifted/moved so that max contiguous free space will be available. This is called as "compaction".


RAM
New process
P9
P8
P7
P9
P6
P5
P4
P3
P6
P2
P1
OS
swap out
swapin
pagefile.sys
window
Swap file
2GB (2×RAM)
Linux/Unix
Swap partition
P6
Swap area
Inactive process/ Sleeping
HDD (disk)
VM (Virtual Memory)
Extension of main mem.
1 Browser
2 Media Play
3 Notepad.
4 MS-paint

Memory Management

Virtual Memory

- The portion of the hard disk which is used by OS as an extension of RAM, is called as "virtual memory".
- If sufficient RAM is not available to execute a new program or grow existing process, then some of the inactive process is shifted from main memory (RAM), so that new program can execute in RAM (or existing process can grow). It is also called as "swap area" or "swap space".
- Shifting a process from RAM to swap area is called as "swap out" and shifting a process from swap to RAM is called as "swap in".
- In few OS, swap area is created in form of a partition. E.g. UNIX, Linux, ...
- In few OS, swap area is created in form of a file E.g. Windows (pagefile.sys), ...
- Virtual memory advantages:
  - Can execute more number of programs.
  - Can execute bigger sized programs.


Segmentation
P1
H
S
D
T
P2
H
S
D
T
limit
Base
1 2000 10000
2 2000 40000
3 1000 45000
4 3000 15000
5 x x
6
7
(segment)s
d(offset)
4 2000
CPU
VA
<
Y
N
Abort
2000
15000
+
PA
17000
Segmentation MMU
RAM
H
S
D
H
D
T
S
D
T
OS
PCB
Segment table
45000
40000
15000
10000
0
1000
2000
3000
2000
x
Swap out
Swap in
Swap area
D
Demand Segmentation
VM
HDD

Segmentation

* Instead of allocating contiguous memory for the whole process, contiguous memory for each segment can be allocated. This scheme is known as "segmentation".

* Since process does not need contiguous memory for entire process, external fragmentation will be reduced.

* In this scheme, PCB is associated with a segment table which contains base and limit (size) of each segment of the process.

* During context switch these values will be loaded into MMU segment table.

* CPU request virtual address in form of segment address and offset address.

* Based on segment address appripriate base-limit pair from MMU is used to calculate physical address as shown in diag.

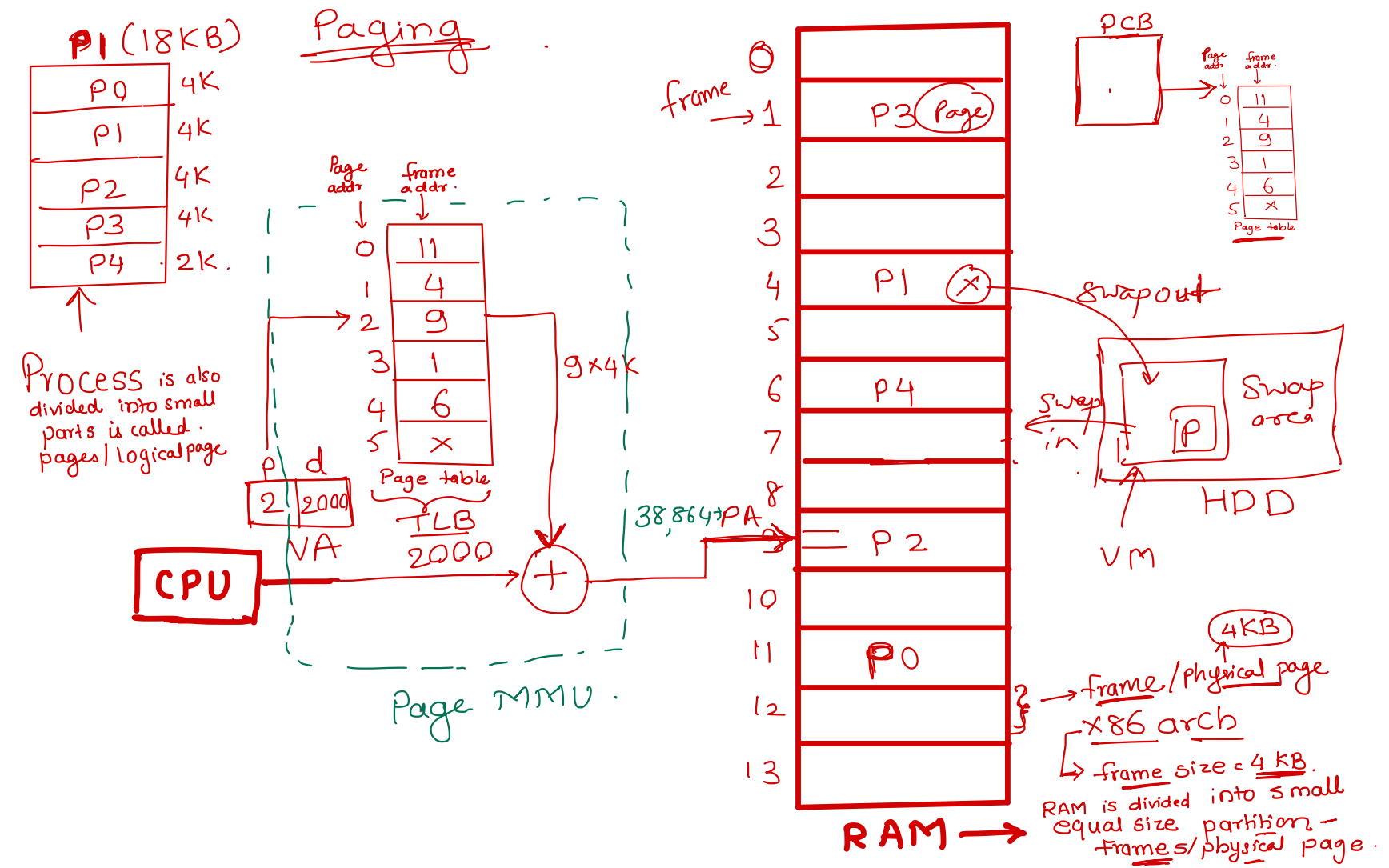
* MMU also contains STBR register which contains address of process's segment table in the RAM.

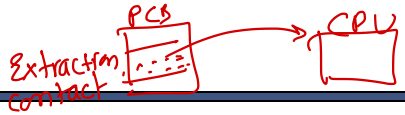
Demand Segmentation

* If virtual memory concept is used along with segmentation scheme, in case low memory, OS may swap out a segment of inactive process.

* When that process again start executing and ask for same segment (swapped out), the segment will be loaded back in the RAM. This is called as "demand segmentation".

* If segment is present in main memory, its entry in seg table is said to be valid. If segment is swapped out, its entry in segment table is said to be invalid.


P1 (18KB)
P0
P1
P2
P3
P4
4K
4K
4K
4K
2K.
Process is also divided into small parts is called pages / logical page
Paging
Page addr.
frame addr.
0 11
1 4
2 9
3 1
4 6
5 X
Page table
TLB
2000
p d
2 2000
VA
CPU
9×4K
38,864 PA
Page MMU
frame
0
1 P3 Page
2
3
4 P1
5
6 P4
7
8
9 P2
10
11 P0
12
13
RAM
PCB
Page addr.
frame addr.
0 11
1 4
2 9
3 1
4 6
5 X
Page table
Swap out
Swap in
Swap area
HDD
VM
4KB
frame / physical page
x86 arch
frame size = 4 KB.
RAM is divided into small equal size partition - frames / physical page.



Paging

* RAM is divided into small equal sized partitions called as "frames" / "physical pages".
* Process is divided into small equal sized parts called as "pages" or "logical/virtual pages".
* page size = frame size.
* One page is allocated to one empty frame.
* OS keep track of free frames in form of a linked list.
* Each PCB is associated with a table storing mapping of page address to frame address. This table is called as "page table".
* During context switch this table contents are loaded into MMU.
* CPU requests a virtual address in form of page address and offset address. It will be converted into physical address as shown in diag.
* MMU also contains a PTBR, which keeps address of page table in RAM.
* If a page is not utilizing entire frame allocated to it (i.e. page contents are less than frame size), then it is called as "internal fragmentation".
* Frame size can be configured in the hardware. It can be 1KB, 2KB or 4KB, ...

Memory Management

Demand Paging

* When virtual memory is used with paging memory management, pages can be swapped out in case of low memory.

- The pages will be loaded into main memory, when they are requested by the CPU. This is called as "demand paging".

- A page table entry (PTE) is valid if page is present in main memory , otherwise entry is invalid.
- What are the reason for invalid entry page is swap out or entry is invalid
- If CPU requesting any page that is not in main memory(i.e. PTE is invalid), then it is a page fault exception.
- **Page fault Page fault handler(OS generate)**

  Check if address due to which page fault occurred is valid. If not , then abort the process.

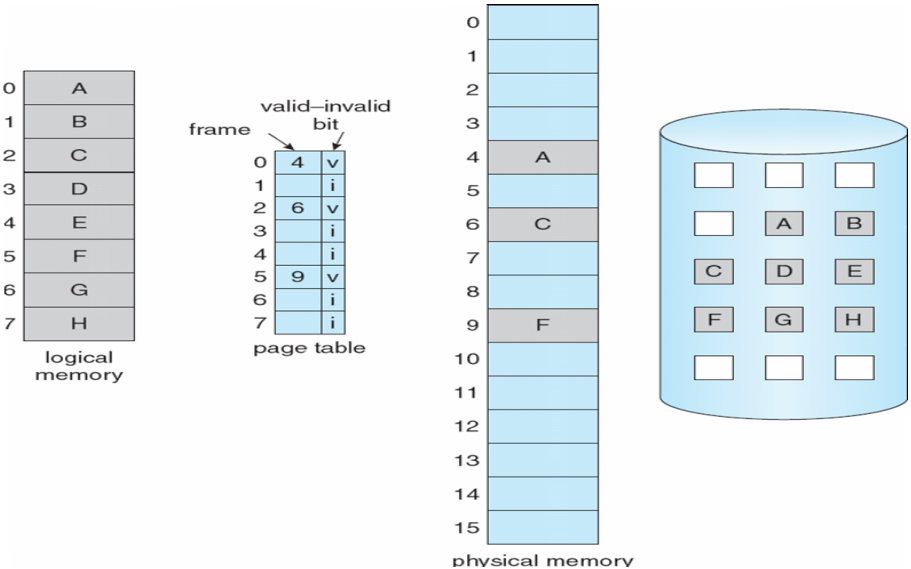
  Allocate an empty frame.

  If page is on swap area, swap in page in that frame

  Update PTE

  Re-execute instruction at which page fault occurred.

Page Table when some pages are not in Main Memory



- If no frame is empty, then OS needs to swap-out some page from memory this is called as “victim page”
- Page replacement algorithm also to decide victim page.

- **Page replacement algorithm**

  FIFO – First in First out

  Optimal

  LRU – List Recently used

  MRU - Most Recently used

  LFU - List Frequently used

  MFU - Most Frequently used