Memory & Storage Management.

- Starts at 9:05pm

Agenda

Memory Management

Memory and its types

How OS manages memory

Virtual Memory

Linux Commands to Monitor Memory

Introduction to Storage

Practical Demo: Swap Creation

Memory Management.

how memory is allocated to different programs.

Types of memory.

1. Primary Memory.

RAM. → Random Acess Memory.

RAM provides **temporary storage** for the data and instructions that your CPU (Central Processing Unit) needs while performing tasks.

→ volatile.

→ extremely fast.

ROM → Read only memory.

non volatile.

→ long term memory.

→ BIOS.

→ unchanging data.

② Secondary memory.

→ hard disk, pendrives

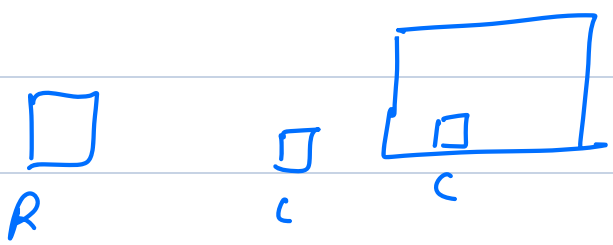
→ data is stored permanently.

③ Cache memory.

→ faster than the RAM

→ stored nearest to the CPU

→ CPU memory.



512KB L_1 cache → fastest cache → smallest, CPU

L_2 cache → slower than L_1

L_3 cache → slower than L_2

↓ speed is decreasing
↓ size is increasing.
↓ distance from CPU is increasing.

Cache Memory vs. RAM

Feature	Cache Memory	RAM (Main Memory)
Location	Integrated directly into or near the CPU	Located on the motherboard, farther from the CPU
Speed	Extremely fast	Slower compared to cache
Capacity	Very small (typically KB to a few MB)	Larger (usually GB)
Purpose	Temporarily store frequently used data for quick access by the CPU	Store data and instructions for currently running programs
Hierarchy Levels	Multiple levels: L1 (fastest), L2, L3	Single level of memory
Proximity to CPU	Closest memory to the CPU (often on the CPU chip)	Located on the motherboard, further from the CPU
Volatility	Volatile (data lost when power is off)	Volatile (data lost when power is off)

How OS manages memory.
→ Allocation.

static allocation

→ allocation is done at compile time.
→ fixed

int array[100]

→ can lead to memory wastage.
→ done using stacks. (LIFO)

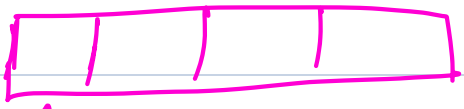
→ dynamic allocation.
→ malloc
→ calloc
→ free up that memory.

Python
l = []

dynamic_array = ()

Fragmentation.

Fragmentation happens when memory is allocated and deallocated dynamically (at runtime), causing the available free memory to become scattered and disorganized. This leads to two types of fragmentation:

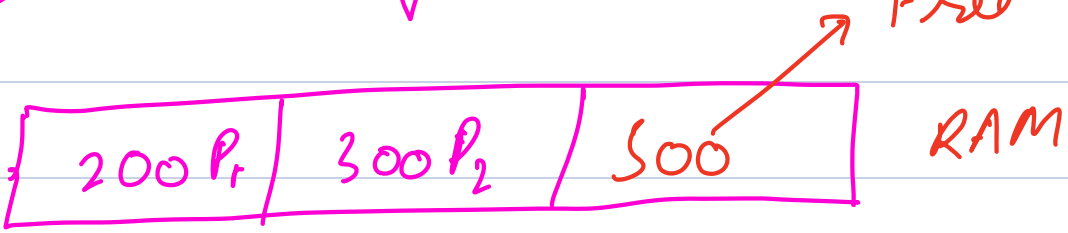


P → 4K 1KB → wasted

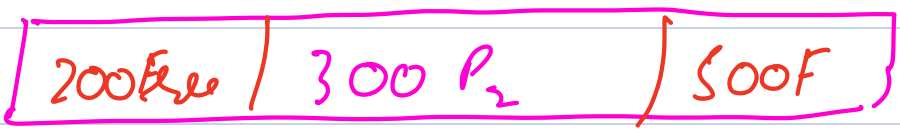
internal fragmentation.

P_1 → 200 bytes

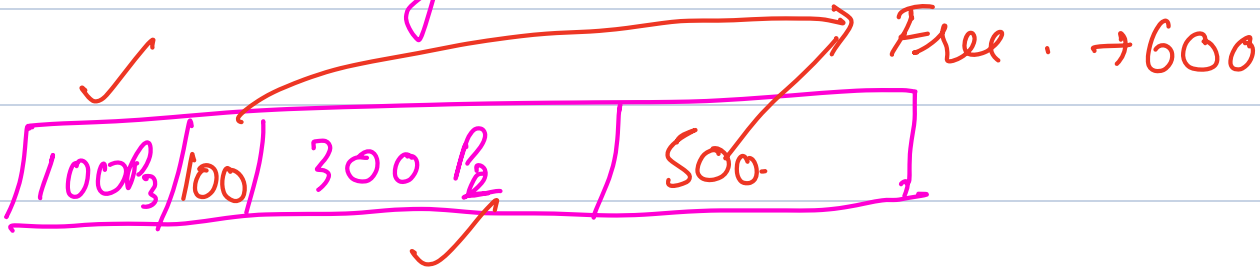
P_2 → 300 bytes



P_1 → deallocated



P_3 → 100 bytes



P_4 → 600 bytes

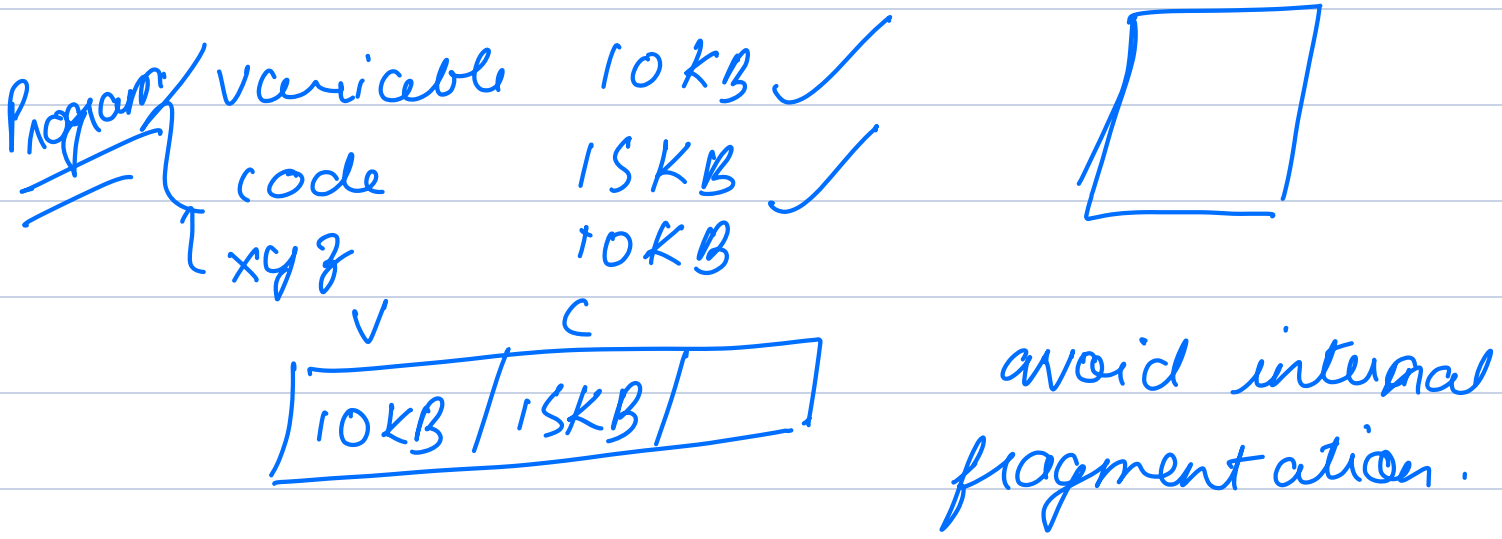
External fragmentation occurs when free memory is divided into small, non-contiguous blocks over time, preventing the

allocation of larger blocks.

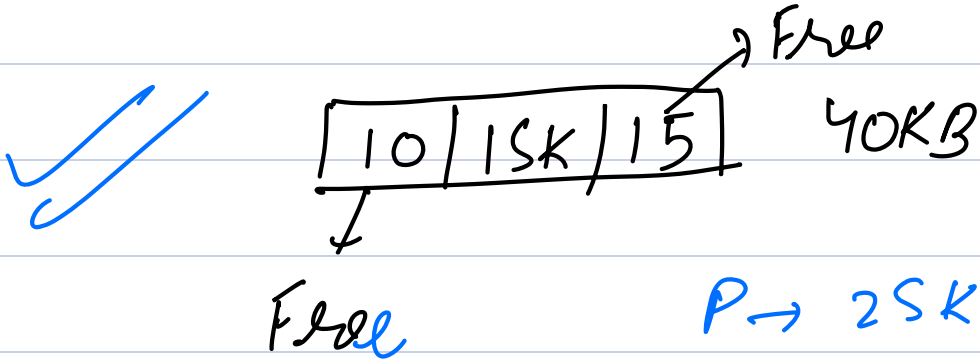
Even though there might be enough total free memory, it is scattered across different places, making it impossible to allocate a large contiguous block.

Paging & Segmentation.

Segmentation.



→ external fragmentation can still occur.



Segmentation is a memory management scheme that divides a process's memory into **variable-sized** segments

based on the program's logical divisions (like functions, arrays, objects, etc.).

→ Paging.

Pages have fixed size

Pages

Frames.

RAM

size of page = size of frame..

{variable
code

Page is a fixed size block. Imagine theres an add function -> divided into 2 halves of equal size.

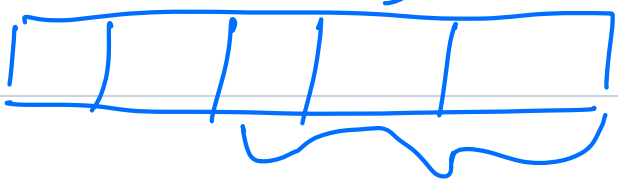
A **frame** is a fixed-size block of **physical memory** (RAM). Physical memory is divided into frames, and each frame is the same size as a page.

1:1 mapping

Page Frame.

→ No external fragmentation.

5 5 5 5 5



15 → 5
5
5
} 3 Pages.

→ internal fragmentation.

→ 3KB → 5 → Page size is 5

2KB wasted.

Paged Segmentation

Garbage Collection.

When objects in a program are no longer referenced,

the garbage collector identifies those objects and frees up their memory.

How GC works.

→ Marking → marks objects in use.

→ sweeping → clears up objects which

are not marked.

→ compacting. → moving objects around in memory to reduce fragmentation.

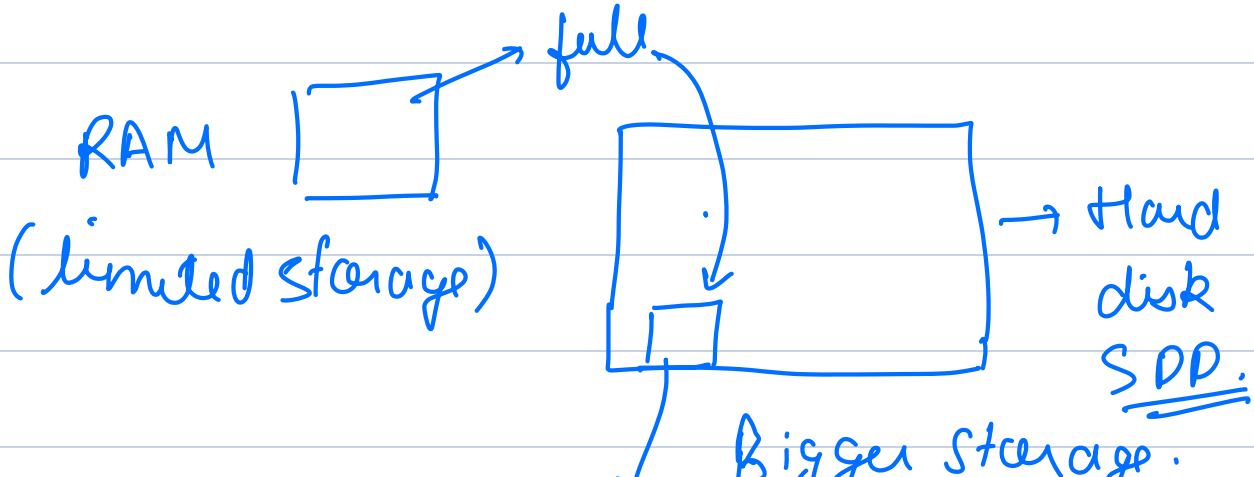
Break → 10:22 pm

Virtual memory.

RAM → is limited size

Virtual memory allows your computer to compensate for having limited RAM by using part of the hard drive as temporary RAM.

VM = RAM + SWAP.



disk

↓

file

↓

swap space.

swap space

8 GB → 10 GB

20 GB → swap space.

swap in

swap out.

→ free - h

→ top

→ cat /proc/meminfo

write a script

→ check free memory.

→ execute some action if free memory is below certain amount.

vmstat | awk 'NR==3 {print $4}'

`procs`:

- **`r`**: Number of processes waiting for run time (running or runnable).

- **`b`**: Number of processes in uninterruptible sleep (blocked).

2. **`memory`**:

- **`swpd`**: Amount of virtual memory used (in kilobytes).

- **`free`**: Amount of free memory (in kilobytes).

- **`buff`**: Amount of memory used as buffers (in kilobytes).

    - **Buffer memory** is a temporary storage area that holds data while it is being transferred between two locations.

- **`cache`**: Amount of memory used for file cache (in kilobytes).

3. **`swap`**:

- **`si`**: Amount of memory swapped in from disk (in kilobytes).

- **`so`**: Amount of memory swapped out to disk (in kilobytes).

4. **`io`**:

- **`bi`**: Blocks received from a block device (in blocks per second).

- **`bo`**: Blocks sent to a block device (in blocks per second).

5. **`system`**:

- **`in`**: Number of interrupts per second.

- **`cs`**: Number of context switches per second.

    - **The CPU will execute one task for a short period (a few milliseconds),**save its state,** **and then switch to

another task.** This happens so quickly that to a human user, it feels like the tasks are happening at the same time.**c

1. **`cpu`**:

- **`us`**: Time spent running user processes (in percentage).

- **`sy`**: Time spent running system (kernel) processes (in percentage).

- **`id`**: Time spent idle (in percentage).

- **`wa`**: Time spent waiting for I/O (in percentage).

- **`st`**: Time stolen from a virtual machine (in percentage).

→ sudo swapon --show

fallocate → preallocates a space for a file.

→ sudo fallocate -l 1G /swapfile

→ sudo chmod 600 /swapfile

→ sudo mkswap /swapfile

↓

mark as swap space.

→ sudo swapon /swapfile

Breakdown of `/swapfile swap swap defaults 0 0`: /etc/fstab.

1. **`/swapfile`**:

   - This is the path to the **swap file** on your system. Instead of using a separate partition (like `/dev/sda3` for swap), the system is using a file located at `/swapfile` as virtual memory.

2. **`swap` (second occurrence)**:

   - This specifies the **filesystem type**. In this case, it is `swap`, which tells the system that this is a swap area.

3. **`swap` (third occurrence)**:

   - Again, this specifies that this line is configuring a swap area.

4. **`defaults`**:

   - This specifies the default mount options for the swap file. For swap, the `defaults` setting typically works without issues and means the following options are applied:
     - **rw**: Read/write access.
     - **suid**: Allow set-user-ID and set-group-ID bits.
     - **dev**: Interpret character or block special devices on the filesystem.
     - **exec**: Permit execution of binaries.
     - **auto**: Automatically mount at boot.
     - **nouser**: Only root can mount.

- **async**: Asynchronous input/output.

5. **`0` (dump)**:

- This field controls whether the **dump** utility will back up the filesystem. Since this is a swap file, there is no need to back it up, so the value is `0` (no backup).

6. **`0` (fsck pass)**:

- This field determines the order in which the filesystem will be checked by **fsck** at boot time. Since swap areas do not need to be checked by `fsck`, the value is set to `0`, meaning no check.

- - **0**: Do not check.

- **1**: Check this filesystem first (usually reserved for the root `/` filesystem).

- **2**: Check this filesystem after filesystems with `pass` value of 1.

History.

history

1 clear

2 free -h

3 top

4 cat /proc/meminfo

5 clear

```
 6  vmstat
 7  vmstat -h
 8  vmstat -S
 9  vmstat -S M
10  clear
11  vmstat -S M
12  free -h
13  vmstat | awk '{print $4}'
14  vmstat | awk '{NR==3 print $4}'
15  vmstat | awk 'NR==3 {print $4}'
16  vmstat 2
17  sudo swapon --show
18  clear
19  sudo swapon --show
20  sudo fallocate -l 1G /swapfile
21  cd /
22  ls -lrt
23  sudo chmod 600 /swapfile
24  ls -lrt
25  sudo mkswap /swapfile
26  sudo swapon /swapfile
27  sudo swapon --show
```

```
28  free -h
29  vi /etc/fstab
30  df -h
31  sudo vi /etc/fstab
32  free -h
33  df -h
34  vmstat
35  history
ubuntu@ip-172-31-43-107:/$
```