In the first half of the lecture, the following topics were discussed:

- Storage (Disks)
- Block Devices (Anything that looks/acts like a disk)
- RAID: (Redundant Array of Independent Disks) (or) (Redundant Array of Inexpensive Disks)

a) RAID 1 (Mirroring)
   Advantage: Reliability
   Disadvantage: Expensive

b) RAID 0 (Stripping)
   Advantage: Improves Speed
   Disadvantage: Will lose data in disk if there is any problem

c) RAID 4 (Parity)
d) RAID 5
e) RAID 6
f) RAID 10 (Mirroring + Stripping)

- Logical Volume Management
  a) Split
  b) Concatenation
  c) Snapshot
In the second half of the lecture the following topics were covered:-

**Flash**
- Computer storage is dominated by disk drives now migrating towards flash storage.
- Characteristics:
  a) Memory like DRAM but permanent.
  b) Its built on something called floating gate.
  c) Represented like:

---

  d) NAND flash - 5 or 10 times dense as DRAM
  \[ 1 \text{GB} \rightarrow 4 \text{GB} \]
  cost: $1 - $5 per GB.

**Difference between Flash and disk:**

a) **Power**:
   - Disks: Medium to high powered.
   - Flash: Very low to medium powered.

b) **Speed**:
   - Disks:
     i) High Sequential read & write.
     ii) Very low random I/O
   - Flash:
     i) Medium to high read (Le) Sequential or random.
     ii) Slow to medium write.
(3) **Write.**

Disks: **overwrite.**

Flash: **write/erase cycle**.

(If data needs to be written, entire block needs to be erased first before writing)

---

**How Flash Translation Layer (FTL) fits into a System:**

---

**FTL:**

1. Handles wear levelling.
2. Uniform Rewritable block interface.
   - Usually Expensive flash drives have better mapping tables & does a better job of wear levelling.
Types of Flash:

SLC → Each cell stores only one bit of information.
MLC → Stores more than 1 bit per cell by choosing between multiple levels of electrical charge to apply to the floating gates of its cells.

Currently used, 2 bits of info per cell results in 4 states per cell:

```
11
10
01
00
```

What to do with block storage?
Put filesystem on top of it.

Filesystem:

a) creating/removing files.
b) storing and organizing files for easy retrieval.

Naming:
```
[device] path - extension version
```

convention
File (can be defined as):
- array of bytes.
- length N
- bytes 0…N-1

File System Services:
(a) create/remove file, directory.
(b) read directory.
(c) read and write a file.
(d) read/write metadata.

(File System Implementation (Sw))

(File System Layout)

Virtual File System (VFS):
- key to how Linux interfaces with File System.

Kernel

VFS

extz DOS XFS NFS

20 or more file systems
my_open() & 3
my_close() & 3
x = 2 open_function = my_open
release_function = my_close
register_something (& x)

File System Layout:
- Locate a Root (2) Beginning of file system
  [In Unix called "superblock"]
- Need to be able to allocate and free disk blocks
  - " " " " " " directories
  - " " " " " " variable length files

Challenges:
- Preventing fragmentation.
- Robustness.

Journaling File System:
- Solution to the Robustness problem.
- Modifications to the file system are written sequentially to a journal.
  (is a circular buffer)
  ext3
  NTFS
  XFS (SGI)
  JFS (IBM)
  ZFS (SUN) ← Journaling is provided in these file systems.
Journal - also called as Redo log.

Log Structured File System -
- solution to the fragmentation problem.
- based on file system created for CDR type media.

- write

1. write A
2. write B
3. write part of A
   so the corresponding portion in actual A gets shadowed.

Need Garbage Collector - can remove actual B, part of A etc...

File might not be deleted until its snapshot is deleted.