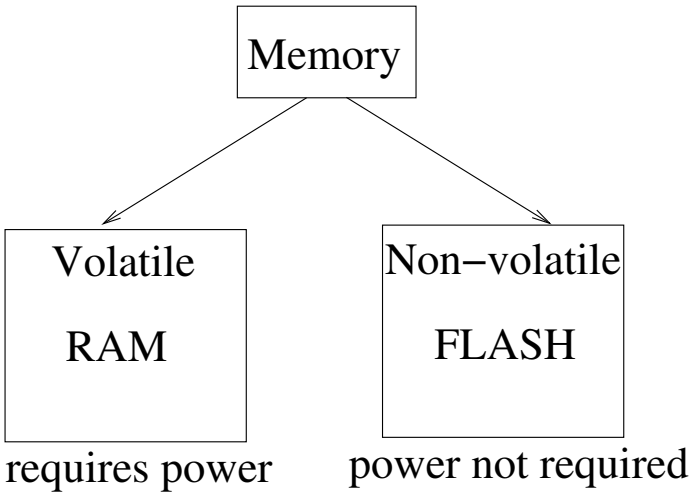
Organization of a computer

Control Unit

- Task is to repeatedly perform fetch-decode-execute cycle.
- It consists of
  - Two important registers PC, IR, and
  - Instruction decoder: circuit that takes opcode and delivers output and control signals to EU.
- The expanded form of fetch-decode-execute is as follow:
  - Fetch an instruction
  - Increment the PC
  - Fetch operands
  - Execute the instruction
  - Store the result.

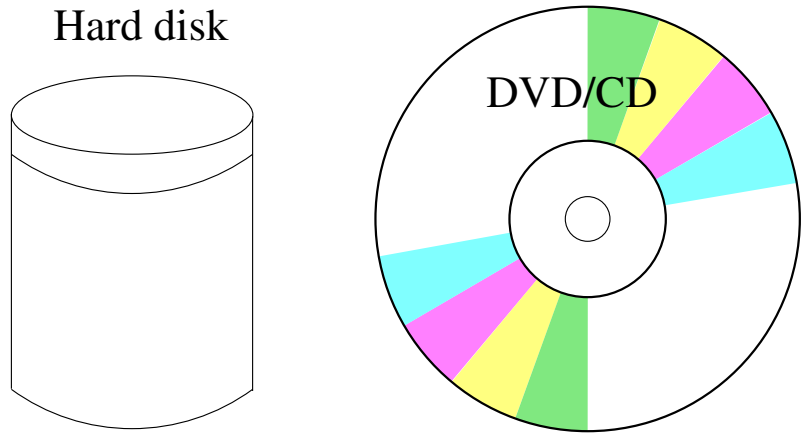
Organization of a computer

Memory



- Saves data/instructions before execution
- Saves intermediate results.

Storage



- Non-volatile, persistent, and stores them in folders.
- Slow compared to non-volatile memory.

Organization of a computer

Input/Output

- Input data provided by human, if large in size should be stored before used for program execution.
- Typical form of input is string (representing values). Stored in files in hard-disks.
- Results produced in human readable form, the output should also be stored before display.
- Typical output device is computer screen.

Program Organization

Analogy with Book

Sl No.	**Book**	**Program**
1	Title page	Heading
2	Table of contents	Listing of function locations
3	Chapter	Module or a package
4	Section	Function
5	Paragraph	Conceptual Block
6	Sentences	Statements
7	Word	Variable
8	Glossary	Variable declarations

Program Organization

Important for beginners...

- How do we log into a computer?
- How do we write a program into a computer's hard disk?
- How do we create a program?
- We use an editor: vi, gedit, emacs, nano, so on.
- Most editors tend to become slow for large files (exceeding 3000 lines).

Software Stack

Bare h/w is not enough

- Electrical circuits need electrical signals as input.
- There should be an interface for communication with m/c.
- Several levels of abstractions make it possible.

op code	operands	
10001011	01000101	11100000
8B	45	E0
add	R3	R7

Software Stack

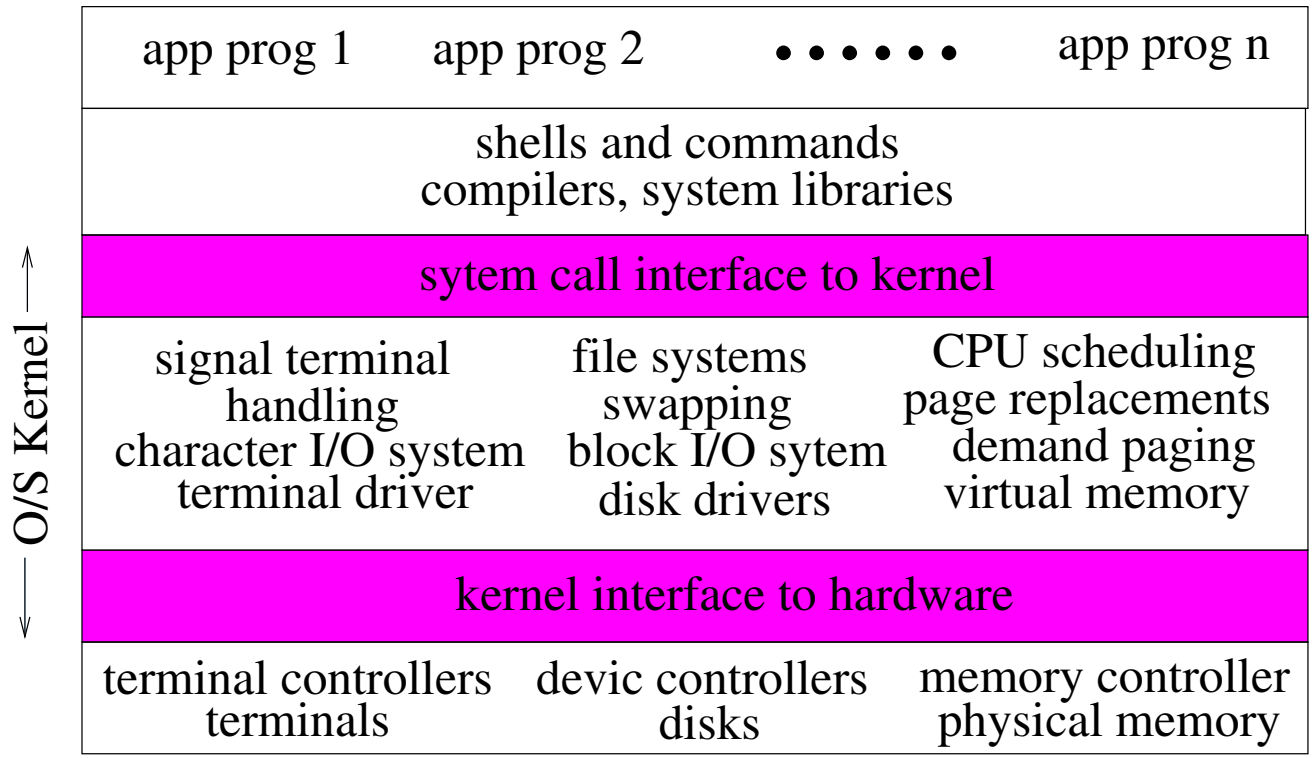
Layering Abstraction

- Next level of abstration is assembly language.
- Final level of abstraction is high level language.
- Not the end of story, even ML programs will need many library programs to be able to execute (output/input).
- In summary many layers of abstractions needed for execution of user programs.

Software Stack

Layering Abstraction

- These layers define the software stack of the machine.



Design of Algorithms

Definition

- A step-wise solution of a given problem
  - A step is a **finite** sequence of 1 or more simple and **precise** operations that can be work out by pen and paper in finite amount of time.
  - Common arithmetic operations.
  - Character manipulations in a string.
  - Reading a number, string, etc.
- The number of steps in a solution must be finite.
- So algorithm must **terminate**.

Design of Algorithms

Historical Note

- Algorithm is a corruption of the name of Al-Kharizmi.
- Mohammad ibn-Musa Al-Khwarizmi (resident of Khwarizm now Khiva in Uzbekistan) was a royal in the court of Baghdad during 780-850 AD.
- Known more as the author of the book: "Al-Kitab al-mukhtasar fi hisab al-gabr wal-muqabala"
- Fibonacci later popularized the preferred form of commercial calculations.

Design of Algorithms

Example

- Steps of calculation are broken down into smaller primitives.
- Execution of primitives solves the problem.
- Eg., multiplying two numbers with primitive:
  1. **determining parity**: odd or even.
  2. **adding two numbers**
  3. **doubling a number**
  4. **halving a number**

Design of Algorithms

Example

x	y	result
		0
123	467	467
61	934	1401
30	~~1868~~	1401
15	3736	5137
7	7472	12609
3	14944	27553
1	29888	555441

```
while (x > 0) {
        if (x is odd)
                result = result + y
        x =   x/2
        y = y + y
}
```

Design of Algorithms

Algorithm and Program

- Algorithm is a step-wise specification of solution.
- Program is an implementation of the algorithm.
- Program is written for a specific machine.
- Program may not terminate (Event driven, buggy programs)

Learning to using computers

Login

- Login requires a UserId and a Password.
- After login goto application on the panel, a popup window appears, click on System Tools then on the second popup click on Terminal.
- Now a terminal appears which actually runs command shell interpreter.
- We must first understand a subset of shell commands.

Learning to using computers

Basic Unix commands

- For files:
  - Create & access control: `touch`, `vi`, `chmod`
  - Copy, move and link: `mv`, `cp`, `rm`, `ln`
  - Information: `ls`, `more`, `cat`, `head`, `tail`, `wc`
  - File comparisons: `diff`, `comm`
  - Packing, compression & decompression: `tar`, `gzip`, `gunzip`
- For directories:
  - `pwd`, `mkdir`, `rmdir`, `mv`, `cd`
  - `cp`, and `rm` also can also be used.

Learning to using computers

Using Internet

- Many browsers are there: firefox, opera, lynx.
- Firefox is most popular.
- Set proxy to `vsnlproxy.iitk.ac.in`
- Provide `user id` and `password`.
- For mail use the web interface.