

# Chapter 01: Introduction

- Introduction to OS
- Operations
- System Components
- Overview of UNIX OS
- UNIX Utility Commands
- UNIX APIs and characteristics

## OS - Handout 01

Match the following

OS Name	Owner
Ubuntu	Apple
Android	Amazon
Fedora	Huawei
Fire OS	Android Inc
Windows	Microsoft
macOS	Canonical
Chrome OS	Google
HarmonyOS	RedHat
iOS	IBM

## Introduction.

Let us parse the phrase:

Operating System.

### Operating

- Operating is the gerund from the verb 'operate'
- Can be a transitive or intransitive verb.

#### Transitive:

- to control the functioning of (a machine, process or a system)

i.e; something other than itself.

Eg: The worker operates the machine.

#### Intransitive:

- to function in a specified manner (of a machine, process or a system)

Eg. Market forces were allowed to operate freely.

Which is the correct sense of "operating" in  
"Operating System"?

### System

A system is a set of components that act together to achieve an overall objective.

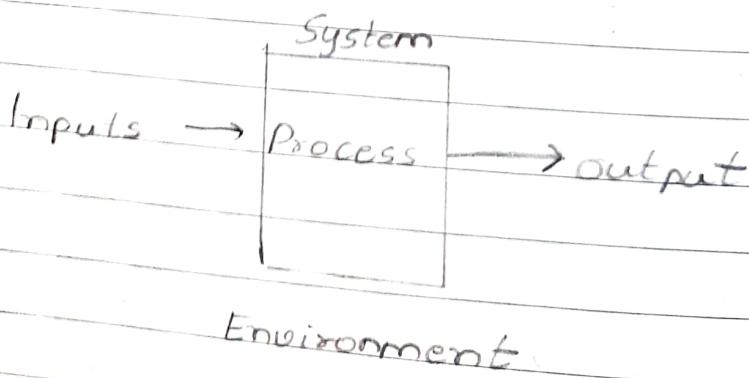
The components may themselves be systems on a smaller scale, in which case they might be called sub-systems.

(now)

The system state is a set of relevant system properties at a given instant of time.

The system environment is a set of artifacts outside the system whose behavior can influence the system in some way.

Any artifact (typically information) that crosses the logical boundary between a system and its environment is either input to or output from a system.



## Analytic Reduction and Atomism

The philosopher and mathematician Descartes is credited with the principle of analytic reduction,

- an approach where a complex problem is divided into distinct subproblems that are analyzed separately.

This approach is subject to following assumptions:

- The division into parts will not alter the phenomena under study
- Each component, studied singly, behaves the same way as it does when part of the whole
- The principles and issues that govern the coalescing and interactions of the components to form the whole are well understood.

## Separation of Concerns

Separation of Concerns is a design principle which states that each element of a program should address a single concern.

↳ off-the-shelf  
components

## Real World Issues - Discussion

### What we already know - or think we do

Devices for computing and communication have changed fundamental paradigms existing for generations

Newer devices have form-factors, compute and communication capabilities not dreamed of even by science-fiction authors a generation or two ago.

People want better form factors - the line has got thin between laptop/notebook and phone/personal digital assistant devices.

People want smaller laptops and more powerful mobile phones (better application performance, more functionality)

We don't know what 2030 holds for us.

### What we think we know

Computing is just about programming and algorithms

Improving the form factor of devices is the critical issue in their improvement.

\* mAh  $\rightarrow$  milliampere-hour

Users have a wide array of computing applications at their disposal

Having more computing power on tap will automatically mean much better applications

Users will always lap up devices with limited functionality as long as they are cheaper.

### What we usually don't think about

Many users now carry or use multiple devices, though this should not be necessary.

Users are making do with their devices in whatever way possible, rather than the devices enabling the users to do what they want.

Much more sophisticated uses of the features already present could be made

A better form factor does not itself translate into more usability.

The watts per flop of computing chips has not decreased at the same rate as the flops have increased, and the power efficiencies of computing technologies have not changed significantly.

FLOPS - Floating Point operations per Second is a measure of computer performance.

classical CS thinking pays no attention to emergent issues, of which an important one is power consumption.

The profile of the average user has changed significantly

- housewife or a garage mechanic than a computing geek

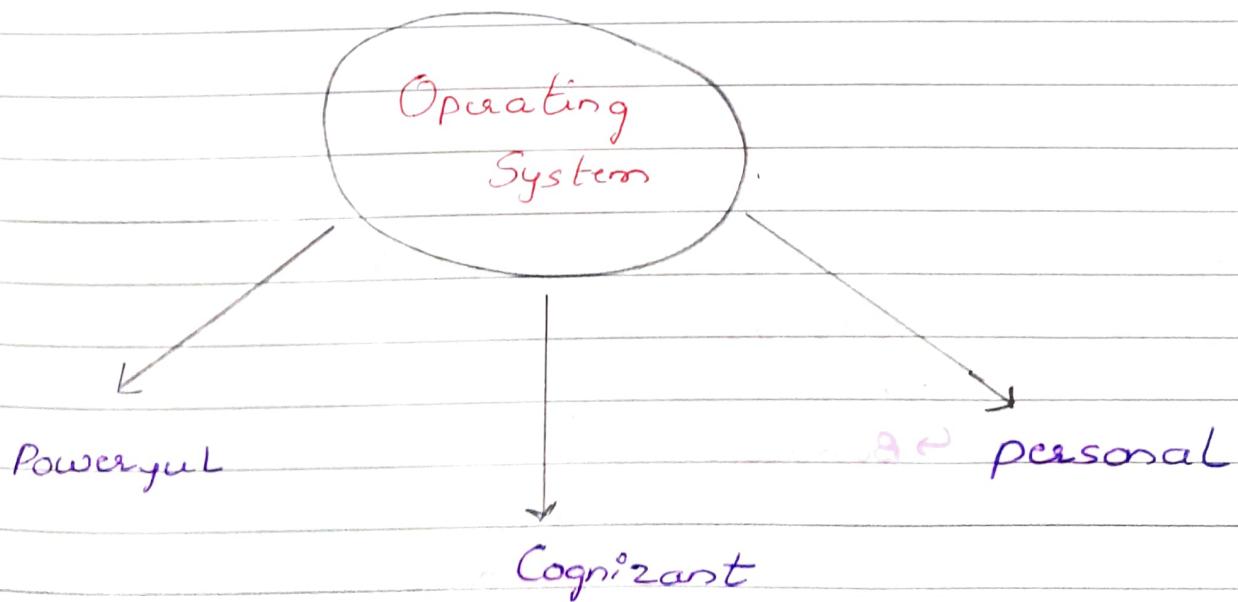
Computing for its own sake is not in fashion now

- not that it was in past

A whole new type of approach is needed in classical theory areas

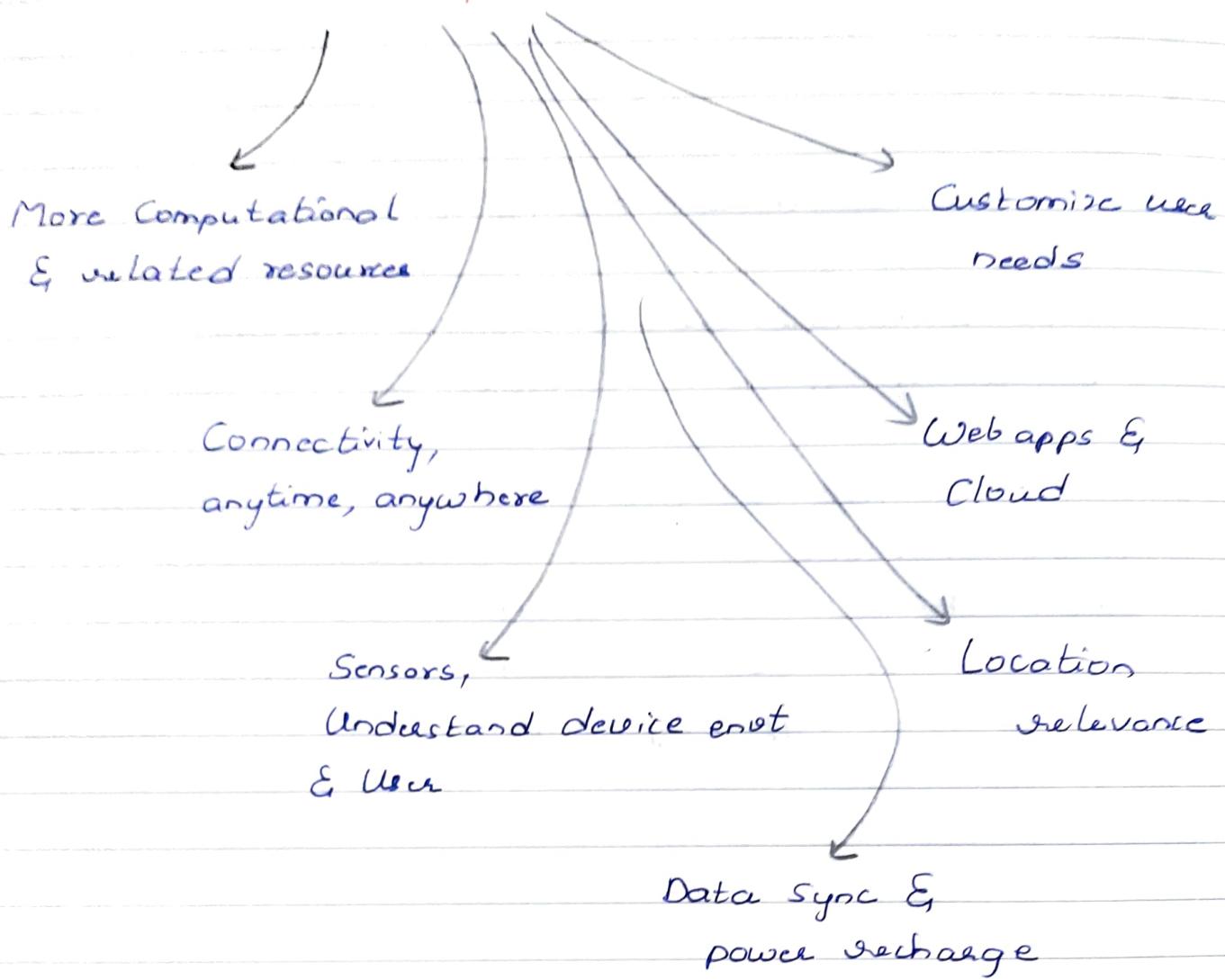
- Eg: Graph Theory

Need



## The - Scope

To name a few



## Technical Trade-offs in design of Devices

Lack of computation?

↳ Memory!

Lack of Memory?

↳ Bandwidth!

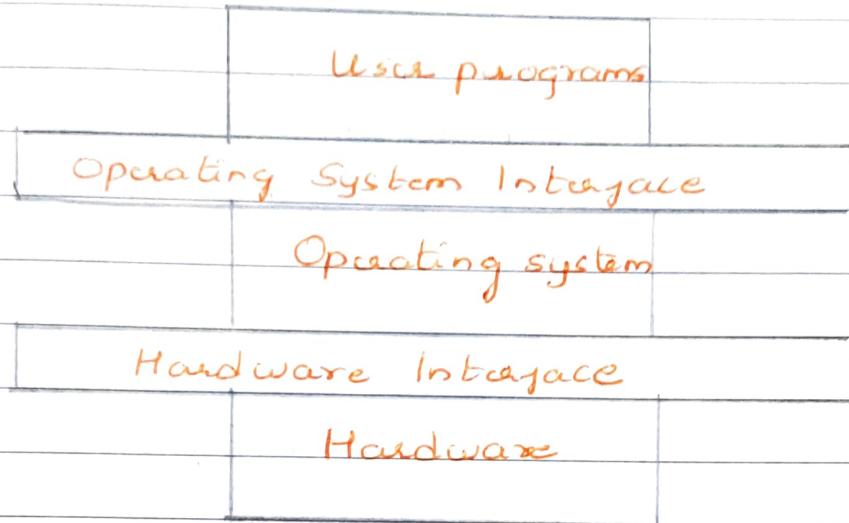
Lack of bandwidth?

Computation!

All 3 require power, but at different rates

## Operating System

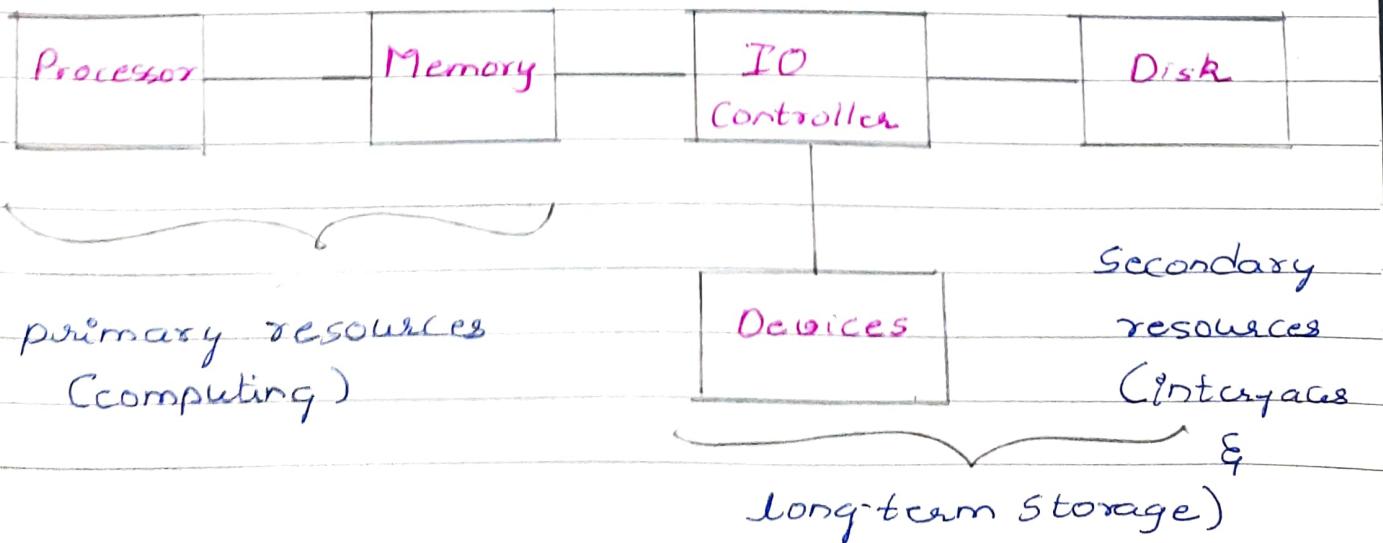
The Operating System is the layer between the hardware and the software.



### Alternate Definition:

A program that acts as an intermediary between a user of a computer and the computer hardware. A system program which controls all the computer's resources and provides a base upon which application programs can be written.

### Hardware Resources in a Computer System



## Do we Need an OS?

- It is not essential to have an OS to use a Computer
- There are many examples of specialized computers that do not use an operating system but, in each one, the duties of the OS are taken over by application code
- A general-purpose Computer System must be able to run a wide range of programs. For such a system, an OS is the most natural way to go.
- An OS is useful because it contains a library of reusable services
- It is a good investment to implement the necessary features once, in the OS, rather than in each application program
- An OS represents a reusable software

### Note:

There are many examples of specialized computers that do not use an OS, but, in each one, the duties of OS are taken over by application code.

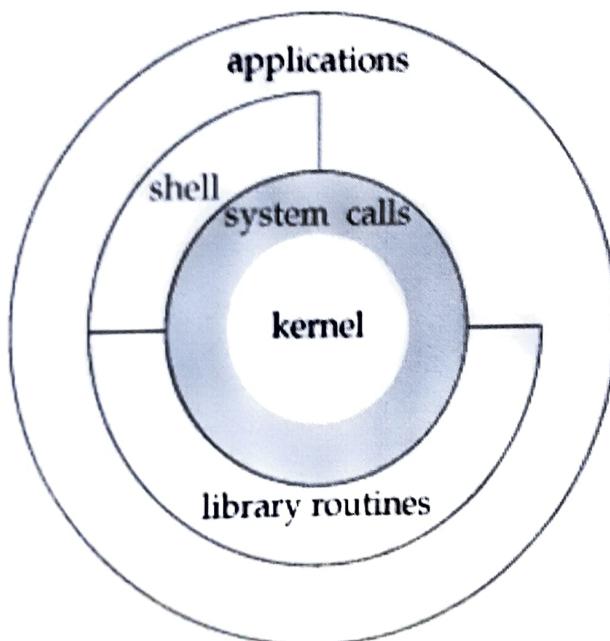
## Discussion

"There can be only one OS running on a Computer"



Only one OS is in direct control of the hardware at any one time.

## UNIX OS Architecture



The interface to the kernel is a layer of software called the system calls.

Libraries of common functions are built on top of the system call interface, but applications are free to use both.

The shell is a special application that provides an interface for running other applications.

In a broad sense, an operating system is the kernel and all other software that makes a computer useful and gives computer its personality. This other software includes system utilities, applications, shells, libraries of common functions, and so on.

Eg: Linux is the kernel used by GNU OS.

OS is a resource allocator

- manages all resources
- Decides between conflicting requests for efficient and fair resource use

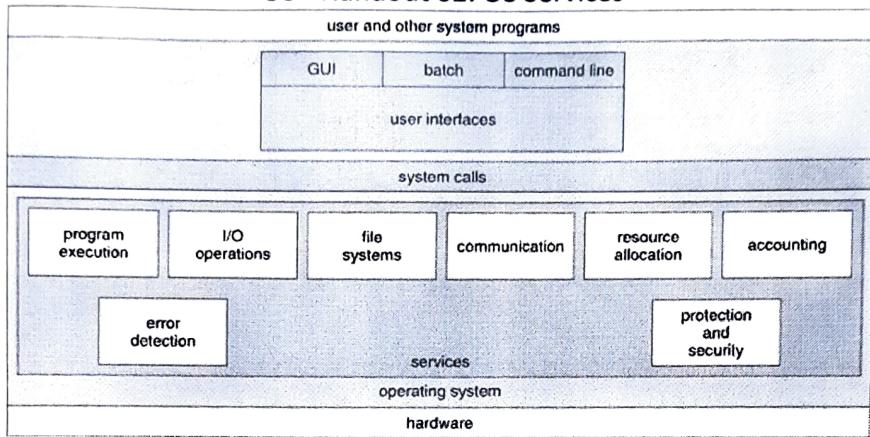
OS is a control program

- Controls execution of programs to prevent errors and improper use of the computer

Note: The user need not know about how the system call is implemented or what it does during execution. User only need to understand API and what the OS will do as a result of execution of that system call.

## OS Services

### OS - Handout 02: OS Services



## UNIX Commands

### - Tutorials