

What is an Operating Systems (OS)?

It is a special program



Can control other programs

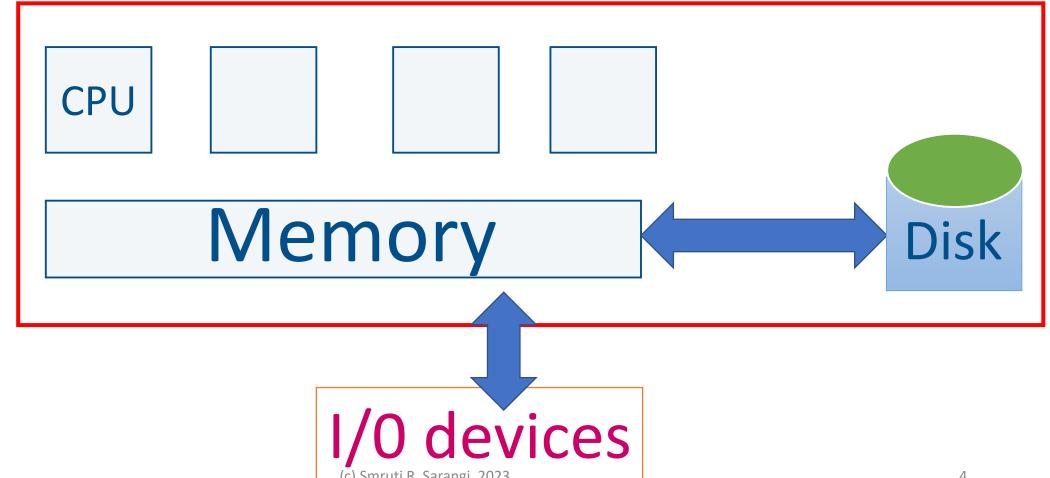
Exercises control over hardware

Background Required

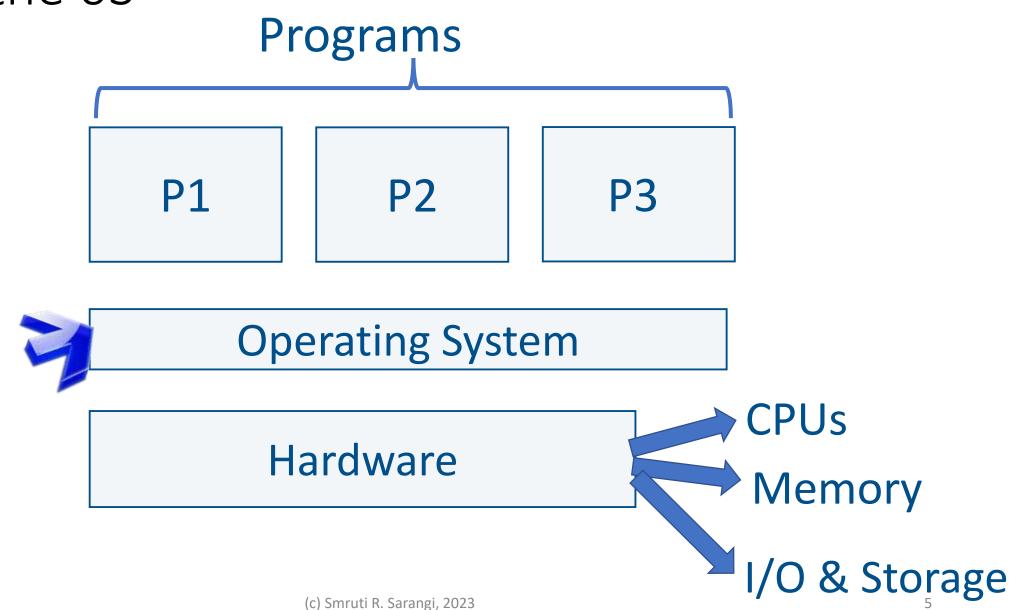
C/C++ programming proficiency Computer Architecture Data Structures

Key Components





Place of the OS





Why do we need a special program, the OS?

- Many programs share common hardware
- The access to common hardware needs to be regulated
- Hardware: CPU, memory, I/O and storage
 - Schedule the CPU
 - Interface with devices
 - Manage storage
 - Manage memory
 - Security
 - Make it easy to build distributed systems
 - Makes it easy to work with HW

Basic Idea



A process is an instance of a running program

Arbitration

- At any point of time multiple processes are active
- The processes have different HW requirements
- The OS arbitrates between the processes



- Accessing HW is very complex
- Elaborate protocols
- Same code needs to run on multiple machines
- Provide a common interface: device derivers in the OS

Access

(c) Smruti R. Sarangi, 2023

Space of operating Systems

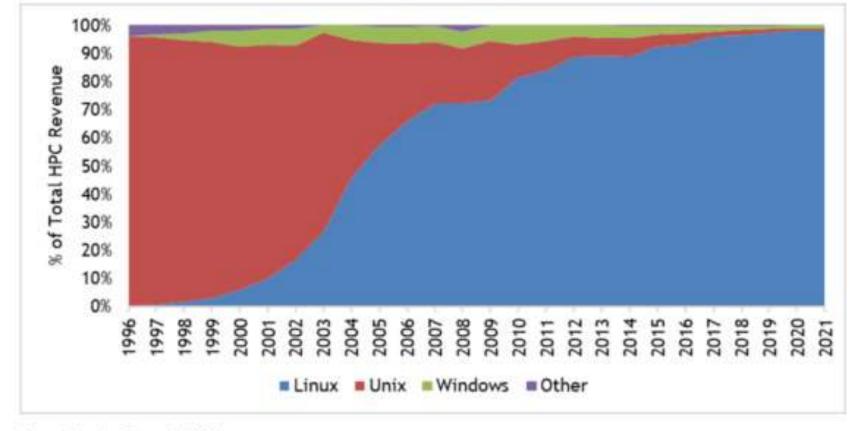
HPC and server Desktop/laptop Mobile

HPC OS market share



Only Linux now

Percent of HPC Server Revenue by Operating System



Source: Hyperion Research, 2022

History of Linux : milestones

1991

Linus Torvalds started extending the MINIX OS

1992

Released under the GPL license. Available on the web.

2000 - 2010

Takes over the server and HPC market

2012 - 2020

Android (Linux-based) becomes the most popular mobile Operating System

Why teach a real —world OS?

- Conveys a realistic picture
- Provides real-world skills





We will learn generic concepts in the context of Linux



Open Source Design

Freely available source code



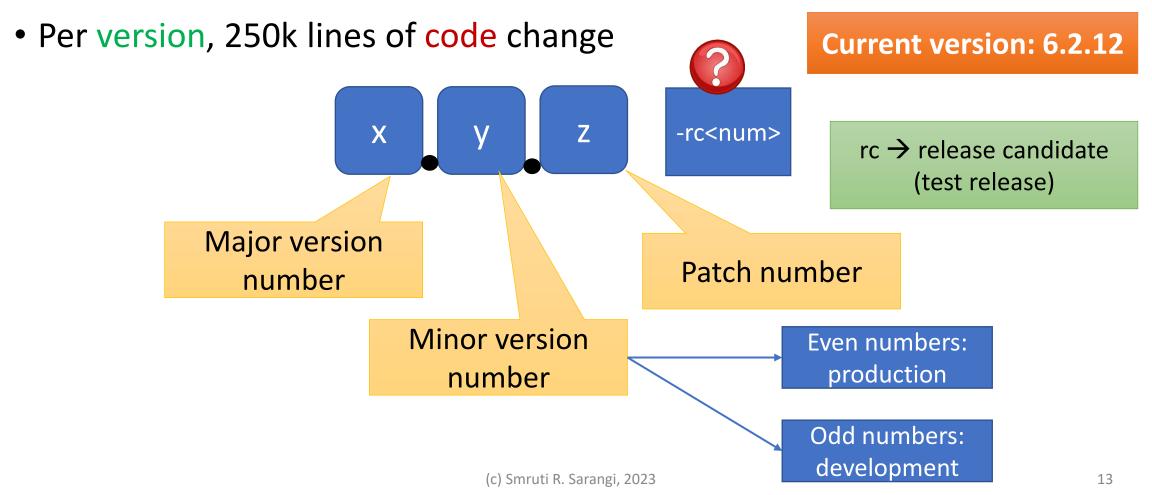
www.kernel.org

https://elixir.bootlin.com/linux/latest/source/kernel

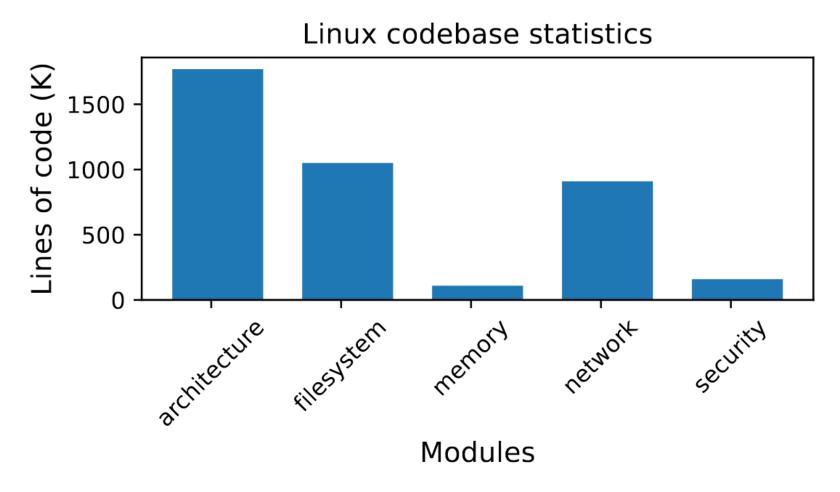
Version 6

Some Kernel Source Code Statistics

Roughly 25 million lines of source code



70% of the code is in drivers, rest in \rightarrow



Major Subsystems in Linux (Directory-wise)

All the code

init: boot code

Process manager

kernel: process scheduler, synchronization, time management, event management, debugging

virt: Support for running a guest OS as a regular program.

Memory and I/O

mm: physical and virtual memory manager

fs: file systems

block: generic layer for all block-based storage devices like hard disks

io_uring: data structures for I/O support R. Sarangi, 2023

arch: All the assembly code that is architecture dependent

HW-dependent code

drivers: All the kernel routines that run the devices. Translate generic OS calls to HW-specific instructions.

Factoid



Android, Chrome OS, Tizen (Samsung), Web OS (LG) are all based on Linux.



srsarangi@cse.iitd.ac.in





