1. Introduction

The **operating system** turns hardware into beautiful abstractions.

They deal with **processes**, which are running images of a program.

All programs that we run behave as processes.

Users — **Processes** -(programing interface)- **OS** -(machine language)- **Computer Hardware**

What happens when the user types a key on the keyboard?

Originally, the CPU is executing some program code. Program counter points to the next instruction to be executed.

This generates a hardware interrupt, which requests the CPU to interrupt the currently executing code (when permitted). Interrupt vector is an array of function pointers pointing to interrupt handlers.

Finally, the interrupt handler returns control to the original code.

System Calls (syscalls)

Similar to ordinary function calls, but its implementation is inside the OS kernel.

System calls are the programming interface between processes and the OS kernel.

They are usually primitive and fundamental. — e.g. time()

What system calls should the OS provide? — process control, file management, device management, information maintenance, communications, and protection

The POSIX (Portable Operating System Interface) standardizes some common system calls.

Two modes of operation: kernel mode vs. user mode

The OS runs in kernel mode (a.k.a. privileged mode).

All other programs run in user mode.

Top-down view: OS is an extended machine.

- It provides a clean abstraction for processes to access the machine.

Bottom-up view: OS is a resource manager.

- It manages all parts of the system efficiently.

Process

A process is an execution instance of a program.

- Multiple processes can execute the same program code.
- A process is not bounded to execute just one program!

A process is an active entity.

- It maintains local states about the execution.

The shell (The UNIX command interpreter)

The shell is not part of the OS, but an important program shipped with the OS

Files

A file is a logical unit of information created by a process.

A file system (FS) is how the OS utilizes the storage device to manage files.

A file system maintains files, directories, allocated space, and free space.

A file system is independent of the OS. An OS can support multiple file systems.

Memory hierarchy

What happens when you run a program?

The program is loaded from hard disk to main memory.

During execution, instructions are fetched from main memory to CPU.