Operating Systems

Dr. Tuğberk Kocatekin
Istanbul Arel University

OS: Operating System

- Software that coordinate a computer’s internal activities and oversee its communication with the outside world.
- Makes the hardware a useful tool.
- Several types:
  - UNIX
  - Unix like (FREEBSD, GNU/Linux, etc.)
  - Windows
- There are also mobile operating systems: Android, IOS, Blackberry OS, etc.

UNIX

- Core of Mac OS & Solaris.
- Created in Bell Labs by *Ken Thompson*. Originally written in Assembly language. Later re-written with C language.
- Very important step in computing.

Additional information:

Brian Kernighan - Unix: A history and a memoir (book)

GNU / Linux

- Linux is a kernel. Created by *Linus Torvalds*.
- GNU (GNU is not Unix) created by *Richard Stallman*.
- GNU and Linux comes together to provide an operating system.
- Open source, therefore several distributions (distro)
    - Debian
    - RedHat
    - Ubuntu
    - ..
- Very widely used. Not in personal computing, but almost all servers are using Linux as an OS.

Open source vs. Free Software

- Open source means you can view the source code.
- It does not mean that you are using the program compiled with that code.
- Unless you can compile the code, you cannot be sure which code you are running.
- You can make money with open source: maintenance
- Free software can be bought, but distributing and using it is free.

- Detailed information on free software:
    - https://www.gnu.org/philosophy/free-sw.en.html

History of Operating Systems

- Execution of problem (job) was isolated. The computer can only run one program.
- An OS was created to run prepared jobs one at a time.
- Later came **batch processing:** Collected jobs in a single batch and ran.
  - Job Queue: FIFO (Modern OS has priorities instead of job queue)
- However, there is also a computer operator. When the user gives the job, it is done. No editing, nothing.
- New OS were created where they allow a program to communicate with the user through remote terminals: **interactive processing.**

History of Operating Systems (cont.)

- Computers are still executing only one job at a time. Therefore multiple users cannot use the computer at the same time.
- **Time-sharing** was the solution. In order to implement it, **multiprogramming** technique is used.
- Here, time is divided into intervals and execution of each job is restricted to only one interval at a time. When an interval ends, the current job is set aside and another is allowed to execute.
- Rapidly changing between jobs gives the illusion that they are being executed in the same time.
- Early time-sharing systems could be used with 30 users simultaneously.

Multitasking

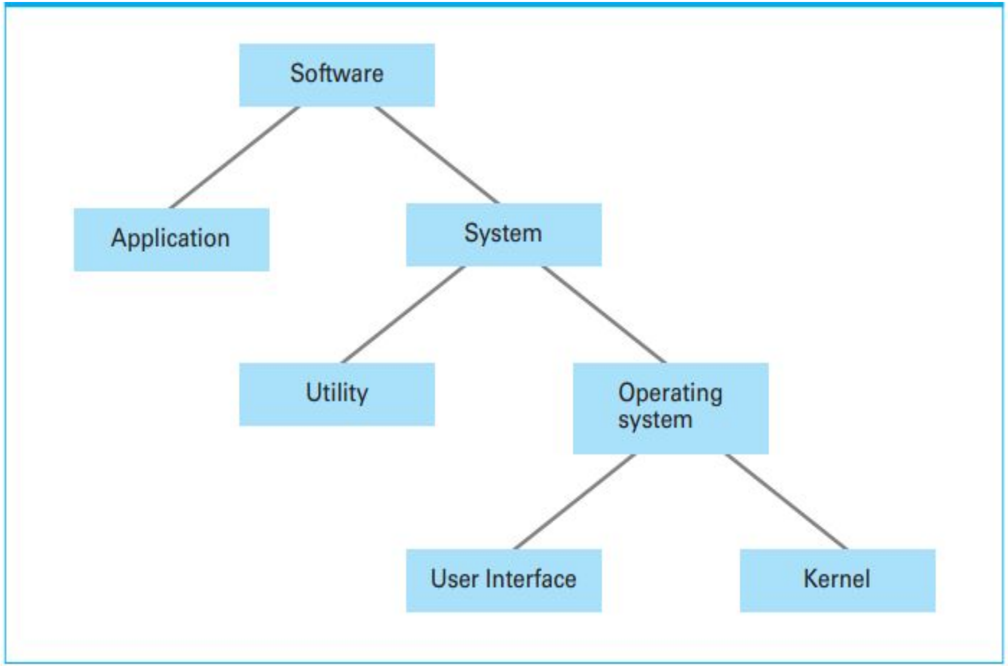
- **Time-sharing** refers to multiple users sharing access to a common computer

- **Multitasking** refers to one user executing several tasks simultaneously.

OS Architecture

- Two categories
  - **Application Software**: *Spreadsheets, publishing systems, program development, games, etc.*
  - **System software:** Tasks common to computers.
    - Operating System
      - User Interface
      - Kernel
    - Utility Software: Software designed to help analyze, configure, optimize &/or maintain a computer.
      - E.g.: *Backup Software, Package Managers, Anti-Virus Software, etc.*

OS Architecture (cont.)



Communicating with the user

- Computer needs to communicate with the user. That is called a user interface. This may be graphical or command line.
    - Shell (Bash, Zsh, Powershell, etc.)
    - GUI
        - Window managers (Style of GUI)
            - In Linux there are lots of window managers to choose from. (i3, xmonad, etc.)
- Earlier versions of Windows was a GUI application. Used to run it from MS-DOS via **win** command.

Kernel

- Contains software components that perform the very basic functions required by the computer installation. (Linux is an open source kernel)
  - File manager
  - Directory path
  - Device drivers
  - Memory manager
  - Scheduler & Dispatcher

File manager

- Coordinates machine’s storage
- Where the files are?
- Who has permission to view what?
- Which portions are open to new files?
- Bundles files into directories (folders).
    - Hierarchy in directories. There are subdirectories.
    - E.g.: animals/prehistoric/dinosaurs

Device Drivers

- Software units communicating with hardware.
- Each driver is specifically designed for the type of device (printer, monitor, etc)
- By using drivers, application software don’t have to deal with them and just rely on drivers.

Memory manager

- For multitasking, programs running concurrently needs space in the memory.
- Coordinates the machine’s use of main memory.
- Must find and assign memory space for the needs of program.
- It needs to keep track of memory areas which are no longer used.

- Sometimes the memory is not enough. That case, a portion of hdd is reserved to be used as primary memory (however, not fast enough). This procedure is called **paging**. The memory provided by paging is called **virtual memory**.

Firmware

- Before the OS is loaded, how does your keyboard work?
- ROM contains a collection of software routines for performing fundamental tasks. These are called **firmware**.
- BIOS & EFI are well-known examples.

What happens when the computer is turned on?

- We need a program in the main memory when the computer is first turned on. However, RAM is volatile. Therefore, a small portion of RAM is constructed from special non-volatile memory cells. That is called ROM.
- **Boot loader** is permanently stored in ROM. (Traditionally)
- Booting is the process of executing the boot loader and starting the OS.
- Mobile phones generally store all OS in ROM. This is not feasible for computers (updates, etc).

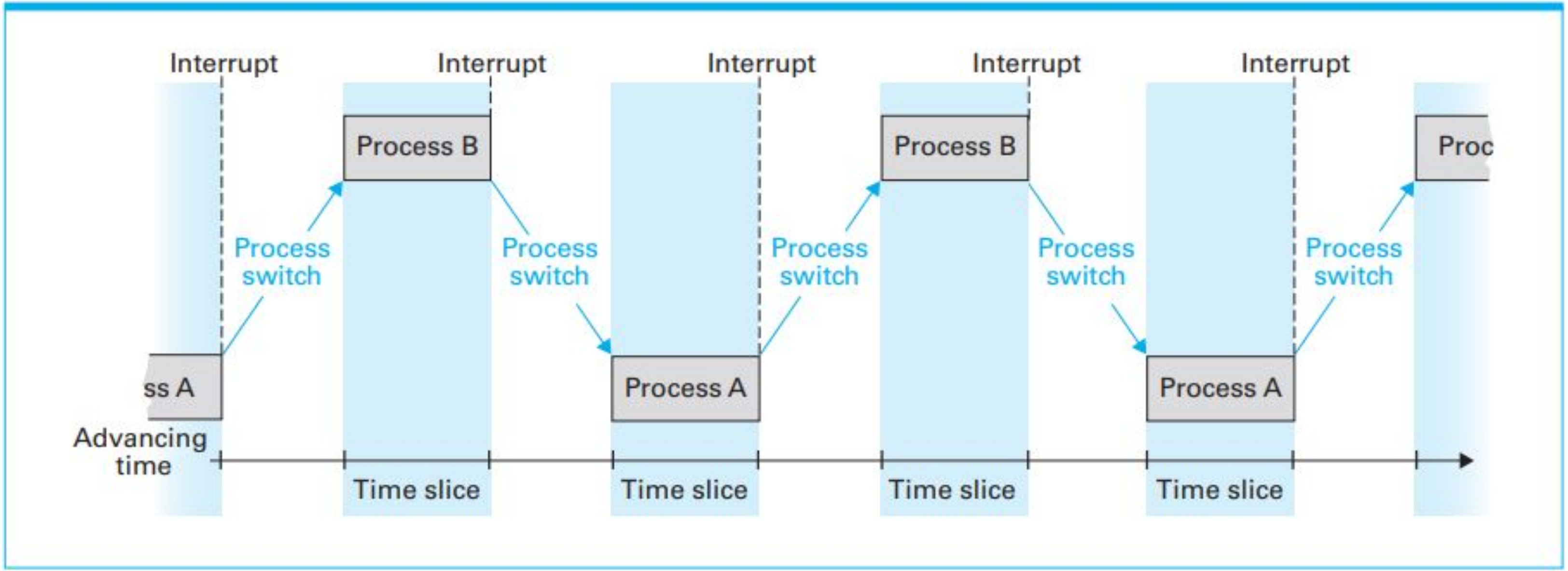
What is a process?

- The activity of executing a program under the control of OS.
- **Process state**: Current status of the activity. (Check your task manager)
- Coordinating execution of processes are handled by scheduler and dispatcher.
  - Scheduler: Maintains a record of the processes present in the computer system. When the user wants to run an app, scheduler adds that to the pool of current processes.
  - Dispatcher: Oversees the execution of scheduled processes.

Context switch / Process switch

- In a time-sharing / multitasking system each task is divided into short segments each called a **time slice (multiprogramming).**
- Each task has a time-slice.
- CPU changes its attention from process to process.
- These are done by the **dispatcher**.
- Each time the dispatcher gives a time slice to a process, initiates a timer circuit. When the timer ends, the circuit generates an **interrupt**.

Context & Process Switch



Concurrency & Parallelism

- What is the difference?
- Concurrency is context switch. Everything is not working in parallel but looks like it.
- Parallelism: Tasks work at different processors at the same time. You need multiple processors.

Semaphores

- Let’s say an application want to use the printer. OS needs to decide whether to give access to the printer but first check whether the printer is already being used by another process.
- OS needs to keep track of whether printer is being used.
- A properly implemented flag is called a **semaphore**.

Deadlock

- Where two or more processes are blocked because they are waiting for a resource, it is called deadlock.
  - One process may have access to printer but be waiting for CD; while another has access to CD player but is waiting for printer.
  - There may be no process left in process table. (kill, end process)