Operating Systems

- An operating system (OS) is a software program that
    1. enables the computer hardware to communicate and operate with the computer software;
    2. provides interface between the computer and its user.

OS Types

As computers have progressed and developed so have the types of operating systems. Below is a basic list of the different types of operating systems and a few examples of operating systems that fall into each of the categories. Many computer operating systems will fall into more than one of the below categories.

Real-time operating system (RTOS)

Real-time operating systems are used to control machinery, scientific instruments, and industrial systems (e.g. flight control systems in aircraft, car computers etc.).

RTOS is specially designed to run applications with very precise timing and a high degree of reliability. This can be especially important in measurement and automation systems where downtime is costly or a program delay could cause a safety hazard. In a complex machine, having a part move more quickly just because system resources are available may be just as catastrophic as having it not move at all because the system is busy (would you like to have Windows as your autopilot or in control of robots? ☺).

To be considered "real-time", an operating system **must have a known maximum time for each of the critical operations** that it performs (or at least be able to guarantee that maximum most of the time).

Single User – Multitasking OS

A single-user multitasking operating system is an operating system that allows a single user to simultaneously run multiple applications on a computer. This type of operating system is found in personal desktop and laptop computers. The most popular single-user multitasking operating systems include Microsoft Windows and MacOS.

What is really happening is that the processor is switching between each program, giving each one a small slice of processor time. This happens fast enough to create the illusion of several programs running at once.

Multi-user OS

A multi-user operating system allows multiple users to use the same computer at the same time.

Linux is a typical multi-user OS: there can be logged on multiple users at the same machine at a time.

Desktop Windows (like Win 7, 8, or 10) can keep multiple user accounts, but one user can be logged only at a time – they are more-over members of the previous category.

Server Operating Systems

A server operating system, also called a server OS, is an operating system specifically designed to run on servers, which are specialized computers that operate within a client/server architecture to serve the requests of client computers on the network.
Server operating systems help enable and facilitate typical server roles such as Web server, mail server, file server, database server, application server and print server.

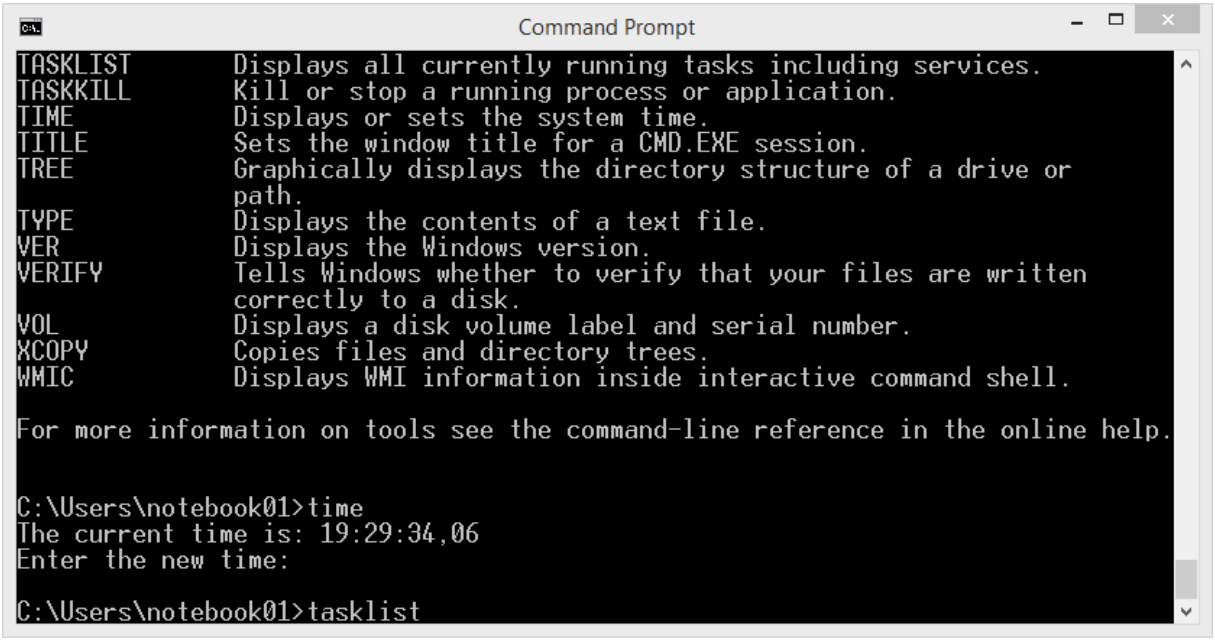
Examples

Windows Server 2012, 2016; Ubuntu Server; Redhat; Debian; various UNIX systems (FreeBSD, OpenBSD …)

User Interface

Command Line Interface

A command line interface or **CLI** is a tool for interacting with computers, often using a text terminal. Commands are entered as lines of text (that is, sequences of typed characters) from a keyboard, and output is also received as text. CLIs originated when teletype machines were connected to computers in the 1950s. In terms of immediate interaction and feedback, they represented an advance over the use of punch cards.



+ *Can be quicker to perform tasks if you know the instructions to write*
+ *Is light on computer resources in comparison to a GUI*

– *Not intuitive, can be hard for beginner users to get to grips with*
– *May take more time to perform complex tasks that could be executed with a few clicks in a GUI*

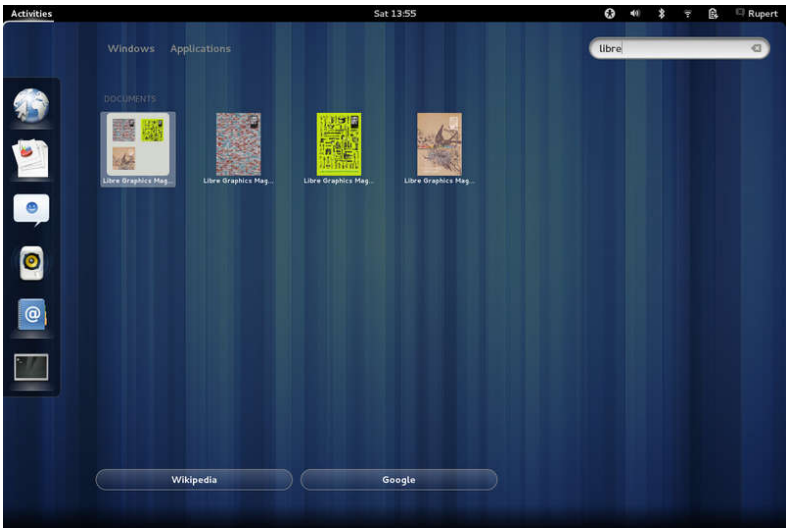
CLI Interfaces

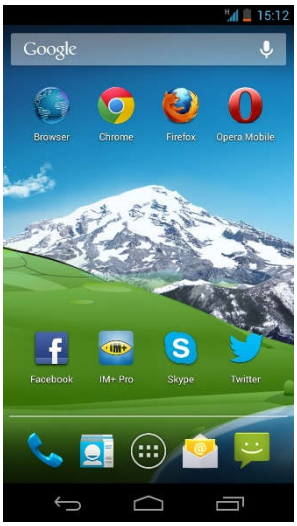
- Windows: Command Prompt (cmd.exe), PowerShell
- Linux, MacOS: bash, csh (C Shell), ksh (Korn Shell)

GUI - Graphical User Interface

GUI Operating System contains graphics and icons and is commonly navigated by using a touch or a pointing device (e.g. a computer mouse). It is intuitive, however hardware demanding. A GUI uses windows, icons, and menus to carry out commands such as opening files, deleting files, moving files, etc. and although many GUI Operating Systems are operated by using a mouse, the keyboard can also be used by using keyboard shortcuts or arrow keys.

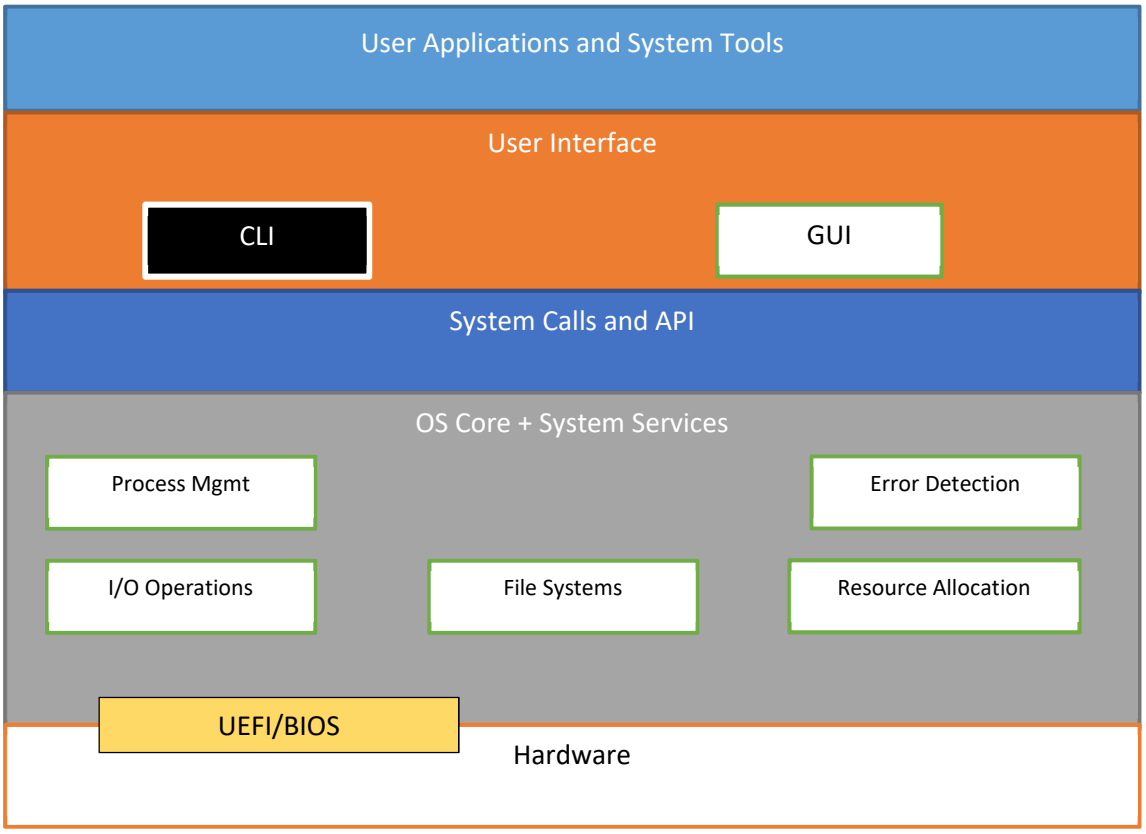
GUI Operating Systems are much easier for end-users to learn and use because commands do not need to be known or memorized. Because of their ease of use, GUI Operating Systems have become the dominant operating system used by end-users today.





OS Structure

The structure varies from OS to OS; however, there are several logical parts/goals, which are done by all OS:


User Applications and System Tools
User Interface
CLI
GUI
System Calls and API
OS Core + System Services
Process Mgmt
Error Detection
I/O Operations
File Systems
Resource Allocation
UEFI/BIOS
Hardware

Process Management

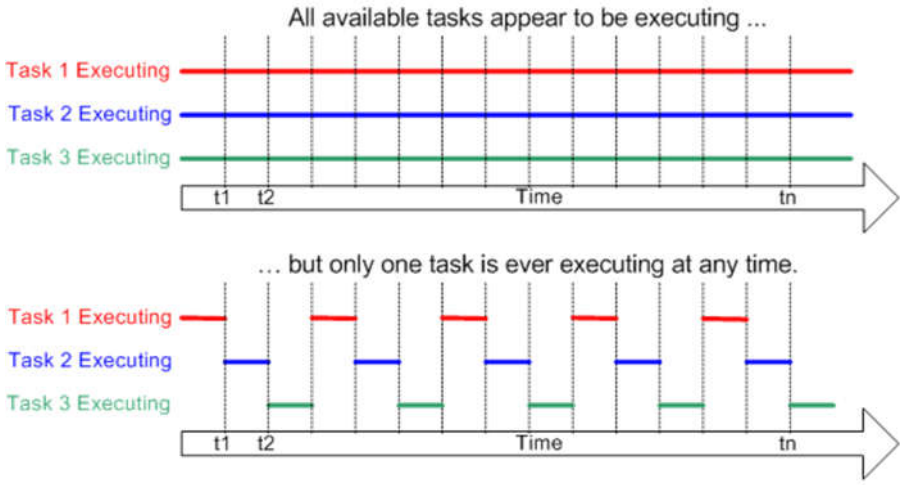
- **Process** – (very simplified definition) a running program = code + data + CPU status (registers, flags etc.)
- Process Management Goals
    1. How to create a new process
    2. How to delete a process
    3. How to run multiple processes simultaneously
    4. How to allow communication among processes (e.g. how to copy a picture from web browser into a text processor)
    5. How to handle processes, which are not responding

Multitasking

Multitasking is the ability of an operating system to allow multiple software processes to run at the same time, even on one single core CPU (in such as case it switches among the running task very quickly, creating an illusion of the parallel processing).

There are 3 basic types:

- **non-preemptive** – a process is utilizing all CPU until it is completed. All other processes have to wait. Jurassic times.
- **cooperative** – one program is in foreground, other running programs are in background; background programs can process data only if the foreground process does nothing. Ancient method.
- **preemptive** – operating system fully manages all running programs and can interrupt them at any time (standard for desktop and server OS). Usually there are short intervals, when a process may use the CPU, and then its execution is stopped, its status is stored, and another process is provided access to CPU.



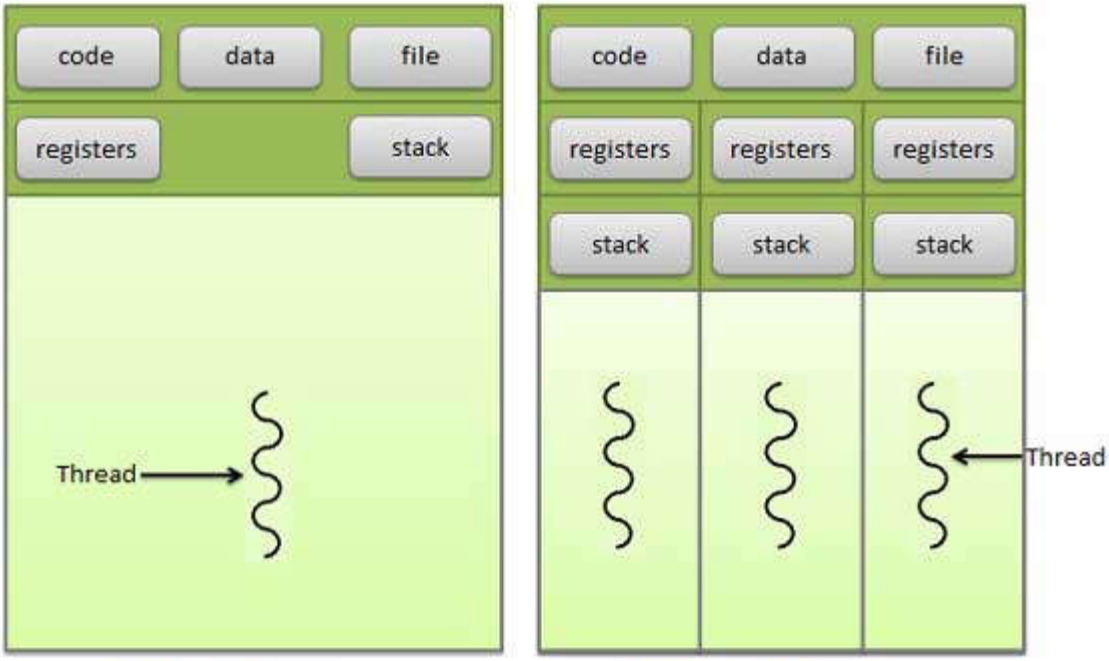
All modern OS support full multitasking (Windows, Linux, MacOS, Android, iOS). There are some limitations in case of the Android and iOS – due to the nature of devices they serve, there are limited option how to **display** multiple running applications at a time.

Multithreading

Operating systems that allow different parts of a program to run concurrently (e.g. Microsoft Word uses several threads: one checks the spelling, another displays the typed

text, another saves files in regular time intervals … - all of them are running simultaneously and are parts of the same program, or – more precise - process).

Threads provide a way to improve application performance through parallelism.



Memory Management

- Primary memory – a large block of bytes stored in memory modules (RAM)
- **Goals**
  - To control, what parts of memory are free and which are in use + by which process
  - To free memory, when a process is stopped
  - How to handle a situation, when the processes require more memory than is physically available (usually solved by swapping – blocks of memory, which are not currently used, are moved to a special part of the hard disk, so some memory can be freed – the user experiences intensive work of the hard disk)

File Management

- File = a collection of related data defined by its creator. There must be a system, which helps to determine such as collection and guarantees its integrity (data cannot be lost "just by the way") – **file system** (FAT32, NTFS, ext3, reiserfs …)
- Goals and actions
  - Creation/deletion of files and folders/directories
  - Mapping the files to different types of storage (optical drives, flash disks, magnetic hard disks, …)
  - Locks for already opened files – if a file is open, then the access is limited or entirely blocked – depends on the lock settings
  - File/folder permissions (who can open, delete, change, rename them …)

I/O System Management

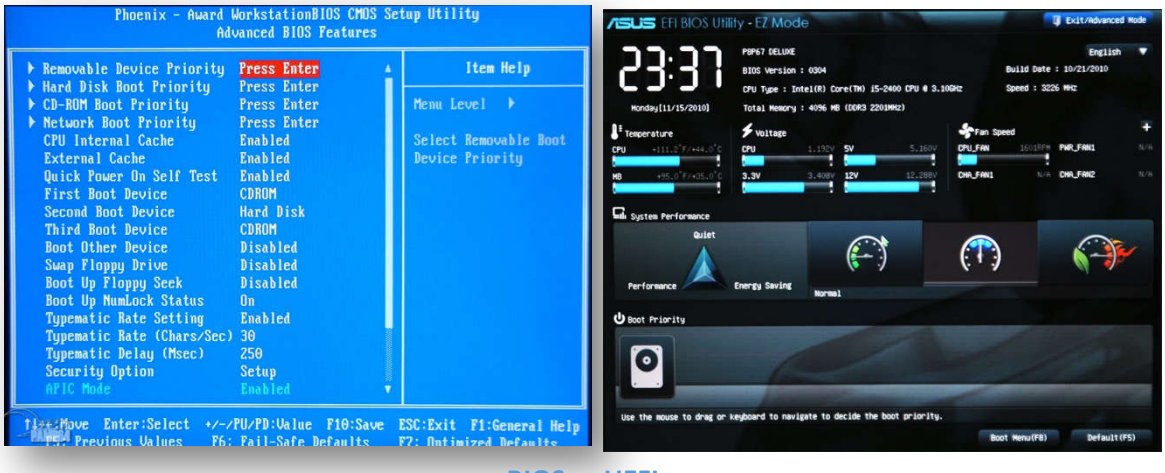
- It simplifies the use of different hardware and hides all specific features of the devices, so applications or users can utilize them as simply as possible (e.g. for an application there is no

difference, if the pointer is controlled by a mouse or a touchpad; the OS must hide their differences)

- **Device drivers** – computer programs, which act as translators between the device and operating system/application; OS/application issues a general command, which is transformed into a specific code "comprehensible" for the device. Usually developed by the hardware producers.
- **UEFI** - **Unified Extensible Firmware Interface**: software interface between OS and computer firmware (firmware is type of software, which directly interacts with hardware and provides services for higher levels). Its goals are:
    - to identify and test important parts of the system (CPU, buses, memory, drives, graphics card, keyboard) – POST test when the computer starts
    - boot an OS from the local drive or from network
    - settings of the computer system – connected drives, peripherals, clock rate …

  UEFI is required by MacOS for quite some time; Microsoft supported some UEFI functions since Windows Vista, full support is available in Windows 8 and newer.

- **BIOS** – Basic Input/Output System – predecessor of the UEFI with many limitations (e.g. 16 bit environment, boot is possible from drive limited to 2 TB; only 4 partitions are supported).



BIOS vs UEFI

Network Management

- A network connection is a vital part of the OS nowadays – almost all applications utilize the network communication (updates, error reports, webpage download/upload ...)
- Problems solved by OS
    - Networking hardware management
    - Identification of the system in a network (IP, host name …)
    - Data routing to/from a network
    - Communication with multiple networks

How to determine Windows specific data (version, service pack)

- Start Menu → Run (or ⊞ + R) → type `winver`
- Start Menu → point on the *Computer* icon → Right click → System Properties (keyboard shortcut: ⊞ + Pause)
- Start Menu → Run (or ⊞ + R) → type `DxDiag` – it starts the DirectX dialogue with more properties.
- Start Menu → About Your PC

Resource Monitor

Resmon.exe – details about processes and assigned resources

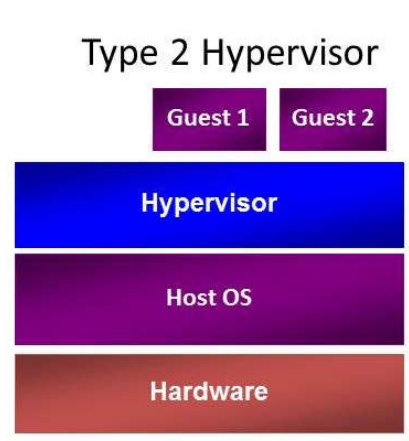
Virtual Machine

VM – also known as Virtual Computer

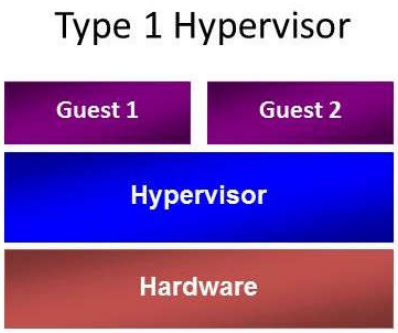
It is software emulation of the computer system – CPU, memory, drive, graphics card, chipset etc. The emulation can be a host for an OS.

There 2 essential types of VM:

- **Type-1 Hypervisors**: also bare-metal hypervisors – part of the software acts like OS, so it can be installed directly, without an underlying OS. Examples: VMware ESXi, Citrix XenServer.
- **Type-2 Hypervisors**: more-over conventional applications, which require an underlying OS. Examples: Oracle VirtualBox, VMware Workstation.



Examples:
Virtual PC & Virtual Server
VMware Workstation
KVM



Examples:
Hyper-V
Xen
VMware ESX

The virtual machine uses are various:

- Test of different OS (e.g. Windows users can try Linux distributions)
- Test of software in various OS
- Better utilisation of hardware – the same physical server can host multiple OS simultaneously
- Simple system backup – backup of a virtual machine is simple – e.g. in the form of a snapshot

What you should be able to demonstrate

- Find OS version
- Show current resources utilization (Task Manager, Resource Monitor)
- Application Management – where to find list of installed apps/uninstall them/add features to OS
- Device Management – available devices, driver version; how to look for updated drivers
- Settings – personalization (wallpaper, colours …)
- User account management, time & date, language
- How to boot to a different OS (e.g. from the flash drive)
- How to get to UEFI or BIOS