**SOFTWARE AND OPERATING SYSTEMS**

Everyone uses a computer to accomplish something, whether that’s browsing the web, writing a novel, making graphics, playing video games etc. Whatever the case may be, they need to interact with their operating system to do. For IT specialists it’s essential to understand how operating systems works so they can help someone accomplish the task they set out to do.

**Computer system components**

A computer system can be divided roughly into three components: the hardware, computer software and the user (picture 1). Computer software consists of application programs and operating system.



Picture 1

**Operating system**

A lot of us hear the term operating system and think of the interfaces of our desktops and phones like menus, buttons and backgrounds. But operating system a little more complex than that.

Operating system is a program manages a computer’s hardware. It provides a means for proper use application programs and acts as an intermediary between the computer user and computer hardware. Using operating system you can utilize hardware components like CPU, memory, I/O devices, like keyboard, mouse etc. The user cannot give commands directly to the CPU in machine language, the CPU can’t interact directly with the user. So there need to be an intermediary which communicates and translates all interactions between the user and the CPU.

Operating system consists of two main parts, the kernel and the user space (picture 2). It controls and coordinates the use of hardware with different application programs.



Picture 2

By application programs we mean the software tools or utilities which the user run to carry out their tasks. And these programs can utilize the hardware resources through the help of operating systems. Application programs include web browsers, word processors and text formatters, spreadsheets, database systems, compilers, games.

**User space**

The user space implies everything outside of the kernel. The users thinks that they interact with directly like system programs, user interface and so on.

User interface generally is a touch screen, where users interact with the system by pressing and swiping fingers across the screen. Some computers have little or no user view. User interface can take several forms. One is a **command-line interface (CLI),** which uses text commands and methods for entering them. **Batch interface**, in which commands and directives to control those commands are entered into files, and those files are executed. **Graphical user interface (GUI)** is a window system with a pointing device to direct I/O, choose from menus and make selections and a keyboard to enter text.

**Kernel**

The kernel is the central, the most important part of operating system. It performs all tasks, like running processes, managing files and hardware devices.

**Kernel. File management**

The kernel does files storage in file management. It can store information on several different types, files can be anything a word document, pdf, a picture, a media file etc. The kernel is responsible for the following activities in connection with file management:

* creating and deleting files
* creating and deleting directories to organize files
* supporting primitives for manipulating files and directories
* mapping files onto secondary storage
* backing up files on stable storage media

**Kernel. Process management (multiprocessing, multiprogramming, multitasking, multithreading)**

Another important function of the kernel is process management. A process is a program that’s executing, like our internet browser or text editor. A program is an application that we can run, like Chrome. Everyone use computers everyday to do several tasks at once, it can be surfing a Google chrome while listening a music or playing video or writing in a text document and so on. The process scheduler is part of the kernel that makes all this tasks multitasking possible.

We can have many processes of the same program running at the same time. And kernel manages our resources efficiently, so that all the programs we want to use can be run. Our system is actually constantly running multiple processes that are necessary for it to function, so our kernel has to worry about all of these processes at once. What a program wants to run, a process needs to be created for it.

The kernel is responsible for the following activities in connection with process management:

* scheduling processes and threads on the CPUs
* creating and deleting both user and system processes
* suspending and resuming processes
* providing mechanisms for process synchronization
* providing mechanisms for process communication

**Multi-programming**

An operating system provides the environment within which programs are executed. And one of the important points of it is the ability to multiprogram. Multiprogramming is a parallel processing where the multiple programs can run simultaneously (picture 3). An operating system keeps several jobs in memory (picture) simultaneously, but the main memory too small to accommodate all jobs and send them to job pool. This pool consists of all processes residing on disk awaiting allocation of main memory. In a multiprogrammed system, the CPU simply switches from one to another job and starts executing it. The CPU executes this jobs until it is interrupted by external factor or it goes for an I/O task.



Picture 3

**Multi-tasking**

Multi-tasking (or time-sharing) is a logical extension of multiprogramming. It is when multiple applications running simultaneously (picture 4). For example, you are looking for information in the browser and edit some document in Microsoft Word at the same time.



Picture 4

**Multi-threading**

Multi-threading is a technique in which processes is divided into threads that can run concurrently. Multi-threading is useful for applications that perform a number of essentially independent tasks that do not need to be serialized. An example is a database server that listens for and processes numerous client requests. With multiple threads running within the same process, switching back and forth among threads involves less processor overhead than a major process switch between different processes.

***Example of multi-threading***(picture 5)

VLC media player, where one thread is used for opening the VLC media player, one thread for playing a particular song and another thread for adding new songs to the playlist.



Picture 5

**Kernel. Memory management**

Memory management one of the important functions of the kernel. It provides a way to allocate a memory portion to programs at their request.

The kernel is responsible for the following activities in connection with memory management:

* monitor which part of memory are currently used and who is using them
* select which processes and data to move into and out of memory
* allocating and deallocating memory space as needed

The CPU can load instructions only from memory, so any programs to run must be stored there. Computer storage roughly divided into primary storage, secondary and tertiary storage.

Primary storage is the main memory directly accessible by the CPU. Types of **primary storage** is:

ROM (read-only memory) is unchangeable memory (pic)



RAM (random-access memory) (pic)



Cache memory

**Secondary storage** is an extension of main memory which not accessible directly to CPU. Types of secondary storage:

Hard disk



Memory card



Flash card

**Tertiary storage**

Types of tertiary storage includes:

Magnetic tapes



Optical disks



Optical tapes



**Kernel. Memory management. Virtual machine**

Virtual machine is a software computer used as an emulation of a computer system. It is based on real computer and provides functionality of a physical computer. The architecture of the virtual machine more complex than that of the physical machine(picture). For virtual machine to work, the hosted physical machine provides software called hypervisor.

****

Picture

**Kernel. I/O management**

Another important function of the kernel is managing input/output devices. Input/output devices include monitors, keyboards, mouse, hard disk, speakers, Bluetooth, webcameras, network adapters and so on. All these I/O devices are managed by the kernel, it takes help from device drivers to handle all I/O devices. I/O doesn’t mean the transfer of data between user and devices, the devices also need to be able to talk to each other.

**Evolution of operating systems**

Early computers allowed only one program to be executed at a time (picture). This program had complete control and had access to all the system’s resources. Today, as you see contemporary computer systems allow multiple programs to be loaded into memory and executed concurrently. One of the characteristics of modern operating systems are multiprogramming, multitasking and multi-threading.



Picture Evolution of operating system

Examples:



**List of operating systems (desktop/mobile)**

As you know there are two types of operating systems: mobile (software that allows smartphones, tablets and other devices to run applications and programs) and desktop (software allows our computers)

Examples of mobile: OS Apple IOS, Google Android, Symbian Nokia and so on

Examples of desktop: We all know the three biggest operating systems used today: Microsoft Windows, Unix and Mac OS



