

Basics of Operating Systems 8-

Before go through the hard topics of operating system, you must have to learn about the basics of operating system such as what a operating system is history of operating system, how operating system works etc.

Operating System is a layer of software that are basically a collection of programs that is used to keep track and use correctly all the computer component such as memory, printers, disks, keyboard display and network interface etc.

You can also say that the job OS is to manage all these devices and provide user program with a simpler interface to the hardware.

Operating system is the medium b/w the computer hardware / components and users. Because without operating system, users can't communicate with these computer hardware.

As you already known that to run computer components or to use computer components, an operating system must required.

Banking System	Airline Reservation	Web browser	} Application Program
Compilers	Editors	Command Interpreter	
	Operating System		} System Program
	Machine Language		
	Microarchitecture		} Hardware
	Physical Devices		

As you can see from the figure given above hardware is at the bottom, which itself can be composed of two or more layers / levels. In it, the lowest level contains physical devices such as wires, power supplies, integrated circuits chips (IC chips) or cathode ray tubes (CRT) etc -

After this level or layer, micro-architecture is the next level. In this level, the physical devices are grouped together to form functional units.

Generally the micro-architecture level contains some registers internal to the CPU or Central Processing Unit and a data path containing an ALU or Arithmetical Logic Unit.

The next level is the Machine Language typically has b/w 50 and 300 instructions. In this level the input/output devices are controlled by loading values into special devices registers.

An operating system must be provided to hide all the complexity. Operating System hides all the hardware components and give the programmer a more convenient set of instruction to work with.

Now in the next level is the rest of the system softwares and application software. All the application program comes after the system programs.

Generally application programs are purchased or may be written by a user or programmer to solve their particular problems.

Network Basics :-

This document covers the basis of how networking works, and how to use different devices to build networks. Computer networking has existed for many years and as time has passed the technologies have become faster and less expensive. Networks are made up of various devices — computers, switches, routers — connected together by cables or wireless signals. Understanding the basis of how networks are put together is an important step in building a wireless network in a community or neighborhood. This module covers the concepts of :-

1. Clients and servers — how services such as e-mail and web pages connect using networks.
2. IP address — how devices on a network can be found.
3. Network hubs, switches and cables — the hardware building blocks of any network.
4. Routers and firewalls — how to organize and control the flow of traffic on a network.

Software Engineering Basics :-

The goal of software engineering is to be a discipline that provides models and processes that lead to the production of well documented, maintainable software in a manner that is predictable. For a mature process it should be possible to determine in advance how much time and effort will be required to produce the final product. This can only be done using data from past experience which requires that one measure the software process.

Models of the development process are essential if it is to be possible to predict the cost of a software project. Some of what you will see in this course may seem obvious or just common sense. Experience will convince you that most software projects fail because they ignore this common sense.

whichever model of the software process you use they will all have some thing in common

- i) A specification phase
- ii) A design phase
- iii) A planning phase
- iv) A module specification phase
- v) A testing phase.

Software engineering also calls for the ability to make precise specification of the behaviour of software system. Several methods exist for doing this. Algebraic specification uses formal techniques based upon specification languages. The Booch method is a graphical technique for describing and thus specifying the behaviour of a system. In fact no one system is likely to be able to do everything, the best software engineers will use a mixture of these techniques.