OPERATING SYSTEM

An operating system (OS) can be defined as a set of programs that controls, supervises and supports a computer system’s hardware and application packages. The fundamental task of an OS is to manage the hardware carefully to achieve the best possible performance.

OPERATING SYSTEM OBJECTIVES

1. To facilitate communication between the computer system and user. The interface through which users issue commands is part of the OS.
2. To facilitate communication among computer system components. OS helps the movement of internal instructions and data between peripheral devices, processor, programs and computer’s storage.
3. To maximize throughput. OS coordinates system resources to maximize the amount of processing per unit time.
4. To minimize the time needed to execute a user command.
5. To optimize the use of computer system resources. OS constantly keeps track of what tasks need to be done and what resources (processor, RAM, peripheral devices) are available to accomplish these tasks. Resource-allocation decisions need to be made at computer speeds.

A SHORT HISTORY OF OPERATING SYSTEMS

The earliest computers had no operating systems - the users had to access computer resources only via machine language programs...

First Generation Computers: (UNIVAC and IBM’s701)

Hardware: already know the hardware – vacuum tubes as ON/OFF switches...
Magnetic drums as the primary memory, directly connected to the ALU
Punched cards as secondary storage

SOFTWARE:
Expressed in long strings of bits that the machine dealt with
ADD wage income and interest income

...1110101010101010101010000001011011011110001...

Arranging the digits into groups of three or four made life a little easier for programmers, but not much...
Writing programs in machine language was error-prone, as well as time-consuming.
The first big software breakthrough was the development of assembly language, which allowed programmers to use operation names and variable symbols.
ADD A and B

The programs were written in assembly language and punched on cards - then each instruction was converted by the computer itself under the direction of a program called the assembler into the string of 0 and 1 that the machine could manipulate.

Second Generation Computers:

Hardware: transistors come into play - the dominant component of a CPU
Primary memory radically transformed – magnetic disk!!! Components managed on printed circuit boards

SOFTWARE:
High-level languages were generated which resembled the English language and syntax.
LET C=A+B

The job of translating such a statement into 1s and 0s required the function of a complex program called the translator or compiler.

Assembly language was a definite improvement over machine language, but still a pain in the neck to work with!!!
Third Generation Computers: (1965-1971)

Hardware:
- Exponential growth in computer industry;
- Technology on the rise = need for upward compatibility, or the ability to run any program written on these machines to run on larger machines of the same series.

SOFTWARE:
- Became more sophisticated;
- Number of high-level languages began to grow;
- Languages more adapted to specific applications were created;
- Expanded power of CPU and increased quantity of input/output devices created a need for efficient management of these components.

Fourth Generation: (1971 to present)

Hardware:
- Microprocessors by Intel Corp.
- Apple...

SOFTWARE:
- The first commercially successful operating system for microprocessors was Digital Research's CP/M.
- IBM decided to license a version of Microsoft's OS program to run on its PC in 1981: PC-DOS and MS-DOS.
- More recently, AT&T's UNIX operating system has been gaining popularity as a highly efficient system.

Managing resources

Who gets where and when?

- Especially important during multitasking operating systems.

Job scheduling

- Assessing priorities among tasks, preparing jobs for execution, directing job flow through the computer system and cleaning up after job completion (resetting input/output devices, etc.).

Handling interruptions

- CPU is much faster than any of the I/O devices. Hence the time lost for each device call must be reduced. A scheme known as interrupt handling is used where the CPU pauses and handles the device call's needs before resuming execution of the program.

Data control

- Programs that control the input and output of data are an important part of every OS.
- Buffers are commonly used for temporary storage of data to reduce the demand on CPU.
- Buffers act as temporary storage areas between the CPU and I/O devices.

Scheduling

- Buffers provide for simultaneous processing of several I/O devices on a single processor.

Interrupts

- Interrupts are temporary suspensions of the running computer program so that a single program can be handled immediately.

A new class of computer programs = OPERATING SYSTEMS
Monitoring system status:

Errors can happen – need to monitor your own operations and functionality!! Control routines constantly check for errors, and resolve these situations as smoothly as possible.

**System crash** is usually a run-time error in the OS.

Two common causes are:
1) Branch instruction to some unintended memory location
2) Accidental erasure of part of the supervisor routine in primary storage.

In general, the larger and more complex the system, the more control points it will have against potential errors.

Utility programs:

Provide users with common necessary functions.

Library managers allow users to build and use their own collections of frequently needed software modules.

Linkers process the machine language code produced by assemblers or compilers and create the final executable module, ready to be run by computer.

File handling programs perform a number of low-level tasks for users, such as creating, deleting, moving, copying and converting programs and data files.

Sort/merge programs enable users to rearrange and combine their data files without having to write their own software.

PROVIDING SERVICES

- Sharing hardware among users
  In systems where there are more than one user at a time, the OS decides who gets what and when.

- Allowing users to share programs and data
  Systems that are accessible to multiple users simultaneously, identical programs need not be duplicated for each user, but shared.

OPERATING SYSTEM TYPES

OSs are organized according to:
- how they organize primary memory
- how many different programs they can execute concurrently
- how many users they can serve at the same time
- the setting in which they are to be used
- the basic design of its components

1) Single user, single task
2) Multiprocessing
3) Multiuser, single task
4) Multiuser, multitask
5) Single user, multitasking
6) Virtual storage
7) Real-time
8) Networking

Single user, single-tasking:
- Simplest type of OS - accommodates single user at a time...
- Can only run single program at a time...

Single user, multi-tasking:
- Multiprogramming is the ability to run more than one program at the same time...
- CPU very rapidly takes turns attending to each task, which makes it seem like the programs are running simultaneously...
- Need sophisticated CPUs so as not to decrease the system performance.

Multiuser, single-tasking:
- Each user has a terminal, connected to the central computer.
- The arrangement is that of a time-sharing.
- Time slices are so short, it looks as if all the users are working simultaneously.

Multiuser, multitasking:
- Sophisticated systems can combine the concepts of time-sharing and multitasking.
- Unix operating system is implemented as a multiuser, multitasking OS on minicomputers, mainframes and supercomputers.

Multiprocessing:
- Used with a computer system that has more than one CPU.
- Execution of several instructions in parallel - TRULY SIMULTANEOUSLY!!!
- Can be much faster than multitasking OSs.

Networking:
- Many individual computers connected together...
- Users can have their own stand-alone computers, while sharing peripheral devices or other hardware components, as well as software and databases...

Virtual storage:
- A memory-management tactic - employs an area of rapidly accessible secondary storage as an extension of primary memory...
- OSs usually implement virtual storage by using segmentation, paging or a combination of both...
  - Segmentation is the process of dividing up a program into a number of chunks, or segments, of different sizes...
  - Paging is like segmentation, except that the pages are of the same size...

Real-time Operating Systems:
- Control computers that interact with their environments to perform work.
- Process control systems take input data from sensors and analyze them, then initiate actions that change the processes that they control.
- Process monitor systems take input data from sensors but merely report these data...
- Increasingly used for industrial and military applications.
- E.g. automated environmental monitoring for air and water pollution, Medical analysis systems, air and automobile traffic control, factory production etc...

MS-DOS
- (Microsoft's Disk Operating System)
- Text-based operating system = commands issued directly to the DOS by entering them on the keyboard, one character at a time...
- (command prompt)
- For example, C:\>copy c:\myfile.txt a:\yourfile.txt
- copies a file from one disk to another
- Need serious education in how to operate DOS, all the commands required for instructing the computer to do certain tasks...
Macintosh OS

The original Apple Macintosh computer employed a powerful 32-bit microprocessor, 1984, Motorola 68000

The computer was revolutionary in design and achieved high degree of user-friendliness:

Mainly due to its OS, which had very little to do with MS-DOS

Based on the Small talk programming language developed at Xerox's Palo Alto center, MacOS was made up of several levels:

- **AT THE TOP:** MultiFinder and the Finder
  - Both system programs, each stored in its own disk file, they manage the Macintosh "desktop" by letting you run your application programs, set up disks, organize, copy and delete files.
- **THEN:** the System File
  - Contains fonts, graphic images, text messages and some of the program code that makes up the MacOS.
- **LOWEST LEVEL:** powerful set of subprograms permanently encoded on a ROM chip that is built into the computer.

The idea behind the MacOS is that the computer should adapt to the user's needs, instead of the other way around !!!

Exceptional user interface:
- Allows user to instruct the computer using a mouse, to manipulate graphic objects on the screen and to select for files, file folders, actions or applications...
- Need very little instruction to learn how to operate the MacOS.

Apple Macintosh holds the patents for many of the Macintosh hardware components, which means that there are no legal Macintosh 'clones'.

While maintaining a high-profit margin for Macintosh, this means that it is not as widespread as other operating systems, such as MS-DOS.

UNIX

A multiuser, multitasking OS originally developed by Bell Laboratories and promoter by AT&T. (First version written in 1971)

The programs that comprise the UNIX OS are written mostly in the high-level programming language C.

Therefore it is relatively easy to transport UNIX to any computer that can run C.

Provides hundreds of utility programs that perform many useful tasks and which can be easily incorporated into programs under development.

UNIX can often be somewhat difficult to learn, however the OS is so flexible that it can be enhanced to present a more friendly user interface.

WINDOWING ENVIRONMENTS

**Graphic User Interfaces (GUI)** allow user to key in text commands or select commands with the click of a mouse.

Eliminates the need for memorizing and entering cumbersome commands.

Windows allows you to divide your screen into a number of different boxes, or windows, and run a separate program in each one.

"Desktop" concept !!!!

Processing activities or files are represented as icons.