INDEX

* INTRODUCTION
* HISTORY
* COMPONENTS
* FUNCTIONS
* CHARACTERISTICS
* DIVERSITY AND ITS PORTABILITY
* REFERENCES

 INTRODUCTION

An **operating system** is a collection of software that manages computer hardware resources and provides common services for computer programs. It is vital component of the system software in a computer system.

It is a layer of software which takes care of technical aspects of computer’s operation. It shields the user of machine from low –level details of machine’s operation and provides frequently needed facilities. Application programs usually require an operating system to function.

For hardware functions such as input and output and memory allocation, the operating system acts an intermediary between programs and computer hardware. Since the operating system is in charge of computer, all requests to use its resources and devices need to go through the OS. Operating system can be found on almost every device that contains a computer-from cellular phones and video game consoles to supercomputers and web servers.

 

The Examples of popular modern operating systems includes Android, IOS, Linux, Mac OS X, Microsoft windows, Windows Phone etc. And among these Microsoft Windows is dominating the market with over 90% market share.

Microsoft Windows is a series of graphical interface operating systems developed, marketed and sold by Microsoft. Microsoft introduced an operating environment named Windows on November 20, 1985 as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUI). The most common versions of Windows for personal computers, mobile devices, server computers and embedded devices are respectively Windows 8, Windows phone 8, Windows Server 2012 and Windows Embedded 8 Industry. And in this report our main focus will be on Windows Operating Systems.

 

 

 **HISTORY**

Windows has had a long history filled with lots of dates, names and version numbers. There are a few key milestones in the history of windows, and here we are speaking at an architectural level, not legal or financial.

It all begins, as always, with v.1.0, which for our purposes is unremarkable if not for the fact that it was the first version of windows. Windows was a 16-bit operating system came out back in 1985-a time when you really had to work hard to fill up 100kb of space.

Windows 2.0 came and went uneventfully, and then came the very popular Windows 3.0 in 1990. In 1992 a minor update, Windows 3.1, was released. This is also, however, a time when things started to get interesting. Windows was just at v3.0 but already there was a completely redesigned version of Windows in the works. Windows NT, windows NT3.1 quickly followed Windows 3.1 in 1993, and unlike Windows 3.1, Windows NT3.1 was 32-bit not 16-bit. Windows NT introduced the Win32 API, which was used in Windows 95 as well and continues on to this day-with significant improvements of course. Windows NT was also written with multiple different architectures in mind, although it didn’t stray far from x86.

 

It was supposed to be the new base for future versions of Windows, but unfortunately this was not to be. Windows 95 got built on Windows 3.1 instead, and so was Windows 98, and –the one that should not be named, but we still will – Windows ME. Meanwhile a line of Windows versions based on Windows NT continued, with Windows NT 4, and then windows 2000(which is essentially Windows NT 5).

 

Windows XP finally merged the old Windows 3.1,95,98 series of Windows with the NT series. It was the also first version of Windows to support the full capabilities of 64-bit processors in a special Windows XP Professional 64-bit edition, which unlike the 32-bit version of Windows XP. And then the Windows Vista was a major update to the Windows platform, and brought Windows to v 6.0, and then came the Windows 7 which is based on Windows NT 6.1.

Then with development of Windows 8, we have another major change, perhaps the most major change to Windows yet. Windows finally fulfills the promise of Windows as kernel that can run on multiple platforms, by embracing ARM. Another interesting development with Windows 8 is the merger of the Mobile and Desktop versions of Windows, just like there was the merger of the Windows NT and Windows 9x back in 2001 with the release of XP. Windows NT 6.2 now runs on desktops and laptops as Windows 8, servers as Windows server 2012, on tablets as Windows RT, and on phones as Windows phone 8.

  

Whereas other operating systems such as UNIX: was developed by some members of MULTICS team at the bell labs in the late 1960’s by many of the same people who helped create the C programming language. Then the LINUX OS came few years later which was developed by Linus Torvalds.

In June of 1978 Apple introduced Apple DOS 3.1, the first operating system for Apple computers. It was in 1997 when Apple introduced its Macintosh operating system series with Mac OS 8, Mac OS 9 in 1999 etc. other series are as follows:

2001- Mac OS X 10.0 code named Cheetah in March, 2001.

2001- Mac OS X10.1 code named Puma in September, 2001.

2002- Mac OS X 10.2 code named Jaguar in August, 2002.

2003- Mac OS X10.3 code named Panther in October, 2003.

2004- Mac OS X 10.4 code named Tiger in June, 2004.

##  Components

The components of an operating system all exist in order to make the different parts of a computer work together. All user software needs to go through the operating system in order to use any of the hardware, whether it as simple as a mouse or keyboard or as complex as an Internet component.

### KERNEL

A kernel connects the application software to the hardware of a computer.

With the aid of the firmware and device drivers, the kernel provides the most basic level of control over all of the computer's hardware devices. It manages memory access for programs in the RAM, it determines which programs get access to which hardware resources, it sets up or resets the CPU's operating states for optimal operation at all times, and it organizes the data for long-term non-volatile storage with file systems on such media as disks, tapes, flash memory, etc.

 

### NETWORKING

Currently most operating systems support a variety of networking protocols, hardware, and applications for using them. This means that computers running dissimilar operating systems can participate in a common network for sharing resources such as computing, files, printers, and scanners using either wired or wireless connections. Networks can essentially allow a computer's operating system to access the resources of a remote computer to support the same functions as it could if those resources were connected directly to the local computer. This includes everything from simple communication, to using networked file systems or even sharing another computer's graphics or sound hardware. Some network services allow the resources of a computer to be accessed transparently, such as SSH which allows networked users direct access to a computer's command line interface.

Client/server networking allows a program on a computer, called a client, to connect via a network to another computer, called a server. Servers offer (or host) various services to other network computers and users. These services are usually provided through ports or numbered access points beyond the server's network address. Each port number is usually associated with a maximum of one running program, which is responsible for handling requests to that port. A daemon, being a user program, can in turn access the local hardware resources of that computer by passing requests to the operating system kernel.

Many operating systems support one or more vendor-specific or open networking protocols as well, for example, SNA on IBM systems, DECnet on systems from digital equipment corporation, and Microsoft-specific protocols (SMB) on Windows. Specific protocols for specific tasks may also be supported such as NFS for file access. Protocols like E-sound, or esd can be easily extended over the network to provide sound from local applications, on a remote system's sound hardware.

1. **SECURITY**

A computer being secure depends on a number of technologies working properly. A modern operating system provides access to a number of resources, which are available to software running on the system, and to external devices like networks via the kernel.

The operating system must be capable of distinguishing between requests which should be allowed to be processed, and others which should not be processed. While some systems may simply distinguish between "privileged" and "non-privileged", systems commonly have a form of requester *identity*, such as a user name. To establish identity there may be a process of *authentication*. Often a username must be quoted, and each username may have a password. Other methods of authentication, such as magnetic cards or biometric data, might be used instead. In some cases, especially connections from the network, resources may be accessed with no authentication at all (such as reading files over a network share). Also covered by the concept of requester identity is *authorization*; the particular services and resources accessible by the requester once logged into a system are tied to either the requester's user account or to the variously configured groups of users to which the requester belongs.

In addition to the allow/disallow model of security, a system with a high level of security will also offer auditing options. These would allow tracking of requests for access to resources (such as, "who has been reading this file?"). Internal security, or security from an already running program is only possible if all possibly harmful requests must be carried out through interrupts to the operating system kernel. If programs can directly access hardware and resources, they cannot be secured.

External security involves a request from outside the computer, such as a login at a connected console or some kind of network connection. External requests are often passed through device drivers to the operating system's kernel, where they can be passed onto applications, or carried out directly. Security of operating systems has long been a concern because of highly sensitive data held on computers, both of a commercial and military nature. The United States Government Department of defence (DOD) created the Trusted Computer System Evaluation Criteria (TCSEC) which is a standard that sets basic requirements for assessing the effectiveness of security. This became of vital importance to operating system makers, because the TCSEC was used to evaluate, classify and select trusted operating systems being considered for the processing, storage and retrieval of sensitive classified information.

Network services include offerings such as file sharing, print services, email, web sites, and file transfer protocols (FTP), most of which can have compromised security. At the front line of security are hardware devices known as firewalls or intrusion detection/prevention systems. At the operating system level, there are a number of software firewalls available, as well as intrusion detection/prevention systems. Most modern operating systems include a software firewall, which is enabled by default. A software firewall can be configured to allow or deny network traffic to or from a service or application running on the operating system. Therefore, one can install and be running an insecure service, such as Telnet or FTP, and not have to be threatened by a security breach because the firewall would deny all traffic trying to connect to the service on that port.

An alternative strategy, and the only sandbox strategy available in systems that do not meet the Goldberg virtualisation requirements, is the operating system not running user programs as native code, but instead either emulates a processor or provides a host for a p-code based system such as Java.

Internal security is especially relevant for multi-user systems; it allows each user of the system to have private files that the other users cannot tamper with or read. Internal security is also vital if auditing is to be of any use, since a program can potentially bypass the operating system, inclusive of bypassing auditing.

1. **USER INTERFACE**

Every computer that is to be operated by an individual requires a user interface. The user interface is usually referred to as a shell and is essential if human interaction is to be supported. The user interface views the directory structure and requests services from the operating system that will acquire data from input hardware devices, such as a keyboard, mouse or credit card reader, and requests operating system services to display prompts, status messages and such on output hardware devices, such as a video monitor or printer. The two most common forms of a user interface have historically been the command line interface, where computer commands are typed out line-by-line, and the graphical user interface, where a visual environment (most commonly a WIMP) is present.

#### Graphical user interfaces

Programs take the form of images on the screen, and the files, folders (directories), and applications take the form of icons and symbols. A mouse is used to navigate the computer. Most of the modern computer systems support graphical user interfaces (GUI), and often include them. In some computer systems, such as the original implementation of Mac OS, the GUI is integrated into the kernel.

While technically a graphical user interface is not an operating system service, incorporating support for one into the operating system kernel can allow the GUI to be more responsive by reducing the number of context switches required for the GUI to perform its output functions. Other operating systems are modular, separating the graphics subsystem from the kernel and the Operating System. In the 1980s UNIX, VMS and many others had operating systems that were built this way. Linux and Mac OS X are also built this way. Modern releases of Microsoft Windows such as Windows Vista implement a graphics subsystem that is mostly in user-space; however the graphics drawing routines of versions between Windows NT 4.0 and Windows server 2003 exist mostly in kernel space. Windows 9x had very little distinction between the interface and the kernel.

Many computer operating systems allow the user to install or create any user interface they desire. A number of Windows shell replacements have been released for Microsoft Windows, which offer alternatives to the included Windows shell, but the shell itself cannot be separated from Windows.

Graphical user interfaces evolve over time. For example, Windows has modified its user interface almost every time a new major version of Windows is released, and the Mac OS GUI changed dramatically with the introduction of Mac OS X in 1999.

**FUNCTIONS OF AN OPERATING SYSTEM**

1. **BOOTING THE COMPUTER**

The process of starting or restarting the computer is known as booting. A cold boot is when you turn on a computer that has been turned off completely. A warm boot is using the operating system to restart the computer.

1. **PROVIDES A USER INTERFACE**

A user interacts with software through the user interface. The two main types of user interface are: Command line & a Graphical User interface (GUI). With a command line interface, the user interacts with the operating system by typing commands to perform specific tasks, eg.DOS (disc operating system). With the GUI, the user interacts with the operating system by using mouse to access windows, tasks and menus, eg Windows vista or Windows 7.

1. **HANDLE SYSTEM RESOURCES**

The operating system also handles system resources such as the computer's memory and sharing of the central processing unit (CPU) time by various applications or peripheral devices. Programs and input methods are constantly competing for the attention of the CPU and demand memory, storage and input/output bandwidth. The operating system ensures that each application gets the necessary resources it needs in order to maximise the functionality of the overall system.

1. **PROVIDES FILE MANAGEMENT**

The operating system also handles the organisation and tracking of files and directories (folders) saved or retrieved from a computer disk. The file management system allows the user to perform such tasks as creating files and directories, renaming files, coping and moving files, and deleting files. The operating system keeps track of where files are located on the hard drive through the type of file system. The type two main types of file system are File Allocation table (FAT) or New Technology File system (NTFS).

1. **TYPES OF FILE SYSTEM**
* File Allocation table (FAT)
* New Technology file system (NTFS)

File Allocation table (FAT) uses the file allocation table which records, which clusters are used and unused and where files are located within the clusters.

NTFS is a file system introduced by Microsoft and it has a number of advantages over the previous file system, named FAT32 (File Allocation Table).

One major advantage of NTFS is that it includes features to improve reliablity. For example, the new technology file system includes fault tolerance, which automatically repairs hard drive errors without displaying error messages. It also keeps detailed transaction logs, which tracks hard drive errors. This can help prevent hard disk failures and makes it possible to recover files if the hard drive does fail.

NTFS also allows permissions (such as read, write, and execute) to be set for individual directories and files.



 

**CHARACTERISTICS OF OPERATING SYSTEMS**

As we have stated, operating systems are normally unique to their manufactures and the hardware in which they are run. Generally, when a new computer is installed, operational software suitable to that hardware is purchased. User wants reliable operational software that can effectively support their process activities.

Though operational software varies between manufacturers, it has similar characteristics. Modern hardware, because of its sophistication, requires that operating systems meet certain specific standards. For example, considering the present state of the field, an operating system must support some form of online processing. Functions normally associated with operational software are:

1. **JOB MANAGEMENT**
A very important responsibility of any operational software is the scheduling of jobs to be handled by a computer system. This is one of the main tasks of the job management function. The operating system sets up the order in which programs are processed, and defines the sequence in which particular jobs are executed. The term job queue is often used to describe the series of jobs awaiting execution. The operating system weighs a variety of factors in creating the job queue. These include which jobs are currently being processed, the system's resources being used, which resources will be needed to handle upcoming programs, the priority of the job compared to other tasks, and any special processing requirements to which the system must respond.

The operational software must be able to assess these factors and control the order in which jobs are processed.

**2. RESOURCE MANAGEMENT**
The management of resources in a computer system is another major concern of the operating system. Obviously, a program cannot use a device if that hardware is unavailable. As we have seen, the operational software oversees the execution of all programs. It also monitors the devices being used. [To accomplish this, it establishes a table in which programs are matched against the devices they are using or will use.](http://elearning.ccnu.edu.cn/ermsweb/libs/cwlib/%E8%AE%A1%E7%AE%97%E6%9C%BA%E4%B8%93%E4%B8%9A%E8%8B%B1%E8%AF%AD/context/7-2.swf) The operating system checks this table to approve or deny use of a specific device.

**3.** **CONTROL OF I/O OPERATIONS**
Allocation of a system's resources is closely tied to the operational software's control of I/O operations. As access is often necessary to a particular device before I/O operations may begin, the operating system must coordinate I/O operations and the devices on which they are performed. In effect, it sets up a directory of programs undergoing execution and the devices they must use in completing I/O operations. Using control statements, jobs may call for specific devices. This lets users read data from specific sites or print information at selected offices. Taking advantage of this facility, data read from one location may be distributed throughout computerized system.

To facilitate execution of I/O operations, most operating systems have a standard set of control instructions to handle the processing of all input and output instructions. These standard instructions, referred to as the input/output control system (IOCS), are an integral part of most operating systems. They simplify the means by which all programs being processed may undertake I/O operations.

 

 **DIVERSITY AND ITS PORTABILITY**

Application software is generally written for use on a specific operating system, and sometimes even for specific hardware. When porting the application to run on other OS, the functionality required by that application may be implemented differently by that OS (the names of functions, meanings of arguments etc.) requiring the application to be adapted, changed or otherwise maintained.

The cost in supporting in operating systems diversity can be avoided by instead writing applications against software programs like JAVA or Qt. These abstractions have already borne the cost of adaption to specific operating systems and their system libraries.

Another approach is for operating system vendors to adopt standards for example, POSIX or OS abstraction layers provide commonalities that reduce porting costs.

 REFERENCES

[www.wikipedia.org](http://www.wikipedia.org)

[www.thinkdigit.com](http://www.thinkdigit.com)

[www.tutorials.com](http://www.tutorials.com)

[www.google.com](http://www.google.com)

magazines