Unit 11 □ Multi User Operating Systems

Structure

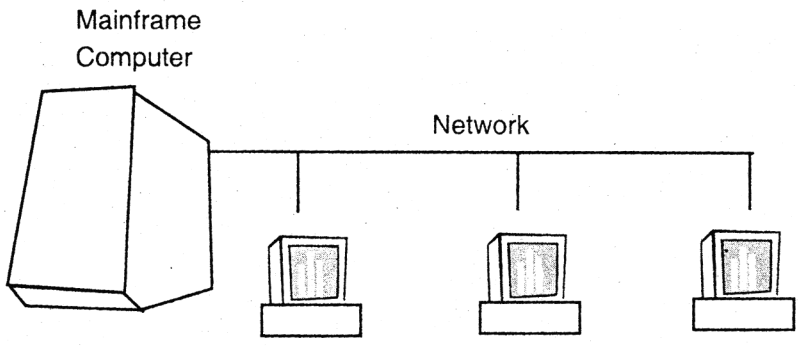
11.0 Objectives

The objectives of the Unit are to :

- Introduce the concept of multi-user and multi-tasking operating system
- Provide basics of UNIX/LINUX
- List selected commands of UNIX/LINUX
- Present brief description of Windows NT

11.1 Introduction

A multi-user operating system lets more than one user access the computer system at one time. Access to the computer system is normally provided via a network, so that users access the computer remotely using a terminal or other computer.



The operating system for a large multi-user computer system with many terminals is much more complex than a single-user operating system. It must manage and run

all user requests, ensuring they do not interfere with each other. Devices that are serial in nature (devices which can only be used by one user at a time, like printers and disks) must be shared amongst all those requesting them (so that all the output documents are not jumbled up). If each user tried to send their document to the printer at the same time, the end result would be garbage. Instead, documents are sent to a queue, and each document is printed in its entirety before the next document to be printed is retrieved from the queue.

In addition, the operating system provides each user with an interface that accepts, interprets and executes user commands or programmes. This interface is commonly called a SHELL or command line interpreter (CLI). In some systems this might be a simple text mode line-by-line entry using keywords (such as MSDOS or UNIX), in other systems it might be highly graphical using windows and a pointing device such as a mouse (such as X-Windows).

The advantage of having a multi-user operating system is that normally the hardware is very expensive, and it lets a number of users share this expensive resource. This means the cost is divided amongst the users. It also makes better use of the resources. Since the resources are shared, they are more likely to be in use than sitting idle being unproductive.

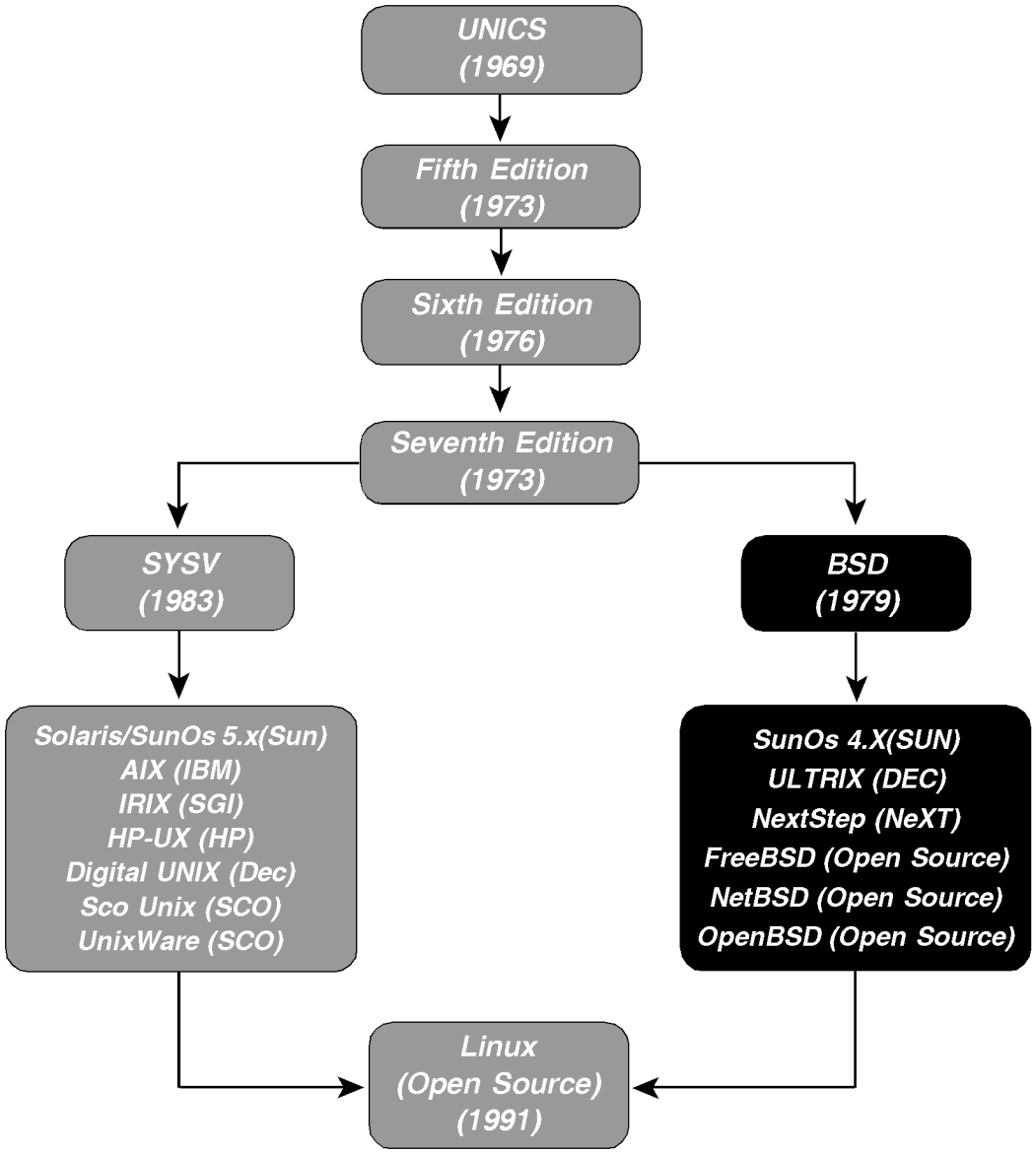
One problem with multi-user computer systems is that as more users access it, the performance becomes slower and slower. Another disadvantage is the cost of hardware, as a multi-user operating system requires a lot of disk space and memory. In addition, the actual software for multi-user operating systems tend to cost more than single-user operating systems.

11.2 UNIX and LINUX

UNIX is a powerful computer operating system originally developed at AT&T Bell Laboratories. It is very popular among the scientific, engineering, and academic communities due to its multi-user and multi-tasking environment, flexibility and portability, electronic mail and networking capabilities, and the numerous programming, text processing and scientific utilities are available. It has also gained widespread acceptance in government and business. Over the years, two major forms (with several vendors' variants of each) of UNIX have evolved : AT & T UNIX System V and the University of California at Berkeley's Berkeley Software Distribution (BSD).

11.2.1 Brief History

In the late 1960s, researchers from General Electric, MIT and Bell Labs launched a joint project to develop an ambitious multi-user, multi-tasking OS for mainframe computers known as MULTICS (Multiplexed Information and Computing System). MULTICS did inspire Ken Thompson, who was a researcher at Bell Labs, to write a simpler operating system himself. He wrote a simpler version of MULTICS on a PDP7 in assembler and called his attempt UNICS (Uniplexed Information and Computing System, eventually shortened to UNIX).



Ken Thompson then teamed up with Dennis Ritchie, the author of the first C compiler in 1973. They rewrote the UNIX kernel in C-this was a big step forwards in terms of the system's portability-and released the Fifth Edition of UNIX to universities in 1974. The Seventh Edition, released in 1978, marked a split in UNIX development into two main branches : SYSV (System 5) and BSD (Berkeley Software Distribution). BSD arose from the University of California at Berkeley where Ken Thompson spent a sabbatical year. Students at Berkeley and other research institutions continued its development. AT&T and other commercial companies developed SYSV. UNIX flavors based on SYSV have traditionally been more conservative, but better supported than BSD-based flavors.

Linux is a free open source UNIX OS for PCs that was originally developed in 1991 by Linus Torvalds, a Finnish undergraduate student. Linux is neither pure SYSV nor pure BSD. Instead, incorporates some features from each (e.g. SYSV- style startup files but BSD-style file system layout) and aims to conform to a set of IEEE standards called POSIX (Portable Operating System Interface).

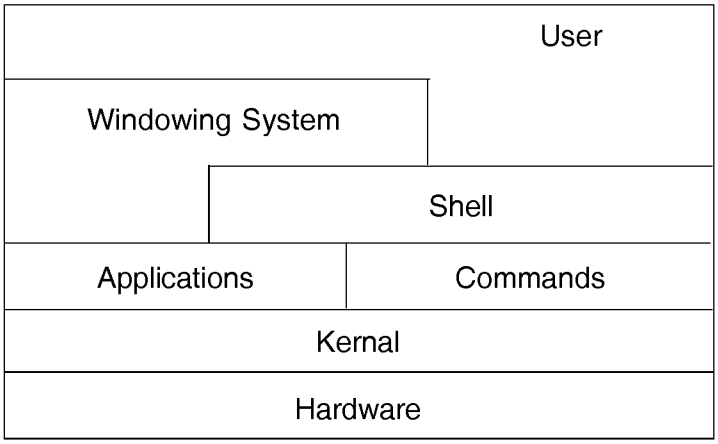
The open source nature of Linux means that the source code for the linux kernel is freely available so that anyone can add features and correct deficiencies. The open source apporach has not just successfully been applied to kernel code, but also to application programmes for Linux.

As Linux has become more popular, several different development streams or distributions have emerged, e.g. Redhat, Slackware, Mandrake, Debian, and Caldera, etc. A distribution comprises a prepackaged kernel, system uitilities, GUI interfaces and application programmes.

11.2.2 Architecture of the UNIX LINUX Operating System

Several layers of interaction are occurring between the computer hardware and user. The first layer is the *Kernel,* which runs on the actual machine hardware and manages all interaction with the hardware. All *applications* and *commands* in UNIX interact with the kernel, rather than the hardware directly, and they make up the second layer. On top of the applications and commands is the command-interpreter programme, the *shell,* which manages the interaction between user, applications, and the available UNIX commands. Most UNIX commands are separate programmes, distinct from the kernel. A final layer, which may or may not be present is a *windowing system* such as X. The windowing system usually interacts with the shell, but it can also interact directly with applications. The final "Layer" is user. User will interact with the entire operating system through just the shell, or through a combination of the shell

and the window system. The figure below gives a visual representation of the layers of UNIX.



Linux has all of the components of a typical OS. They are :

- **Kernel :** The Linux kernel includes device drive support for a large number of PC hardware devices, advanced processor and memory management features, and support for many different types of file systems.
- **Shells and GUIs :** Users interacts with the system through a command interpreter programme called the *Shell.* Linux supports two forms of command input : through textual command line shells similar to those found on most UNIX systems (e.g.sh-the Bourne shell, bash-the Bourne again shell and csh-the C shell) and through graphical interfaces (GUIs) such as the KDE and GNOME window managers. Remote connection to a server will typically be through a command line shell. In addition to processing user command requests, UNIX shells have their own syntax and control constructs. User can use these shell commands to make processing more efficient, or to automate repetitive tasks. User can even store a sequence of shell commands in a file, called a *shell script,* and run it just like an ordinary programme.
- **System Utilities :** Virtually every system utility has been ported to Linux. These system utilities are designed to be powerful tools that do a single task extremely well. Users can often solve problems by interconnecting these tools instead of writing a large monolithic application programme. Like other UNIX flavors, Linux's system utilities also include server programmes called daemons, which provide remote network and administration services. A daemon is usally spawned automatically at system startup and spends most of its time lying dormant waiting for some event to occur.

- **Application Programmes :** Linux distributions typically come with several useful application programmes as standard. Examples include the emacs editor, xv (an image viewer), gcc (a C comoplier), g++ (a C++ compiler), and soffice (StarOffice, which is an MS-Office style clone that can read and write Word, Excel and PowerPoint files). Redhat Linux also comes with rpm, the Redhat Package Manager, which makes it easy to install and uninstall application programmes.

11.2.3 Basic UNIX/LINUX Elements

Six basic elements of UNIX are :

1. Commands : are the instructions user give the system to tell it what to do.
2. Files : are collections of data that have been given *filenames.* A file is analogous to a container in which user can store documents, raw data, or programmes. A single file might contain the text of a research project, statistical data, or an equation processing formula. Files are stored in *directories.*
3. Directories : is similar to a file cabinet drawer that contains many files. A directory can also contain other directories. Every directory has a name, like files.
4. User environment : is a collection of items that describe or modify how user-computing session will be carried out. It contains things such as where the commands are located and which printer to send user output to.
5. Processes : A command or application running on the computer is called a *process.*
6. Jobs : The sequence of instructions given to the computer from the time user initiate a particular task until it ends it is called a *job.* A job may have one or more processes in it.

11.2.4 Syntax of UNIX Commands

A UNIX command line consists of the name of the UNIX command followed by its arguments. The option flags act like adverbs by modifying the action of the command, and filenames and expressions act like objects of the verb. The general syntax for a UNIX command is :

Command -flag options *file/expression*

Flags need not always be specified separately, each with their own preceding dash. Many times, the flags can be listed one after the other after a single dash. User should follow the following rules with UNIX commands :

1. UNIX commands are case-sensitive, but most are lowercase.
2. UNIX commands can only be entered at the shell prompt.
3. UNIX command lines must end with a RETURN.
4. UNIX options often begin with a "–" (minus sign).
5. More than one option can be included with many commands.

11.2.5 UNIX/LINUX File System

The UNIX operating system is built around the concept of a file system which is used to store all of the information that consitutes the long-term state of the system. This state includes the operating system kernel itself, the executable files for the commands supported by the operating system. Configuration information, temporary work-files, user data, and various special files that are used to give controlled access to system hardware and operating system functions. Every item stored in a UNIX file system belongs to one of four types :

1. **Ordinary files :** Ordinary files can contain text, data, or programme information. Files cannot contain other files or directories. Unlike other operating systems, UNIX filenames are not broken into a name part and an extension part (although extensions are still frequently used as a means to classify files). Instead they can contain any keyboard character except for '/' and be up to 256 characters long (note however that characters such as*, ?, # and & have special meaning in most shells and should not threrefore be used in filenames). Putting spaces in filenames also makes them difficult to manipulate-rather use the underscore '_'.
2. **Directories :** Directories are containers or folders that hold files, and other directories.
3. **Devices :** To provide applications with easy access to hardware devices, UNIX allows them to be used in much the same way as ordinary files. There are two types of devices in UNIX -block-oriented devices which transfer data in blocks (e.g. hard disks) and character-oriented devices that transfer data on a byte-by-byte basis (e.g. modems and dumb terminals).
4. **Links :** A link is a pointer to another file. There are two types of links - a hard link to a file is indistinguishable from the file itself. A soft link (or symbolic link) provides an indirect pointer or shortcut to a file. A soft link is implemented as a directory file entry containing a pathname.

The UNIX file system is laid out out as a hierarchical tree structure, which is anchored at a special top-level directory known as the root (designated by a slash '/'). Because of the tree structure, a directory can have many child directories, but only one parent directory.

To specify a location in the directory hierarchy, we must specify a path through the tree. The path to a location can be defined by an absolute path from the root /, or as a relative path from the current working directory. To specify a path, each directory along the route from the source to the destination must be included in the path, with each directory in the sequence being separated by a slash. To help with the specification of relative paths, UNIX provides the shorthand "." for the current directory and ".." for the parent directory. For example, the absolute path to the directory "play" is /home/will/play, while the relative path to this directory from "zeb" is ../will/play.

The following table shows some typical directories that may be found on UNIX systems and briefly describes their contents. Note that although certain subdirectories appear as part of a seamless logical file system, they do not need be present on the same hard disk device; some may even be located on a remote machine and accessed across a network.

Directory	**Typical Contents**
/	The "root" directory
/bin	Essential low-level system utilities
/user/bin	Higher-level system utilities and application programmes
/sbin	Superuser system utilities (for performing system administration tasks)
/lib	Programme libraries (collections of system calls that can be included in programmes by a complier) for low-level system utilities
/usr/lib	Programme libraries for higher.level user programmes
/tmp	Temporary file storage space (can be used by any user)
/home or	User home directories containing personal file space for each user.
/homes	Each directory is named after the login of the user.
/etc	UNIX system configuration and information files
/dev	Hardware devices
/proc	A pseudo-filesystem which is used as an interface to the kernel. Includes a sub-directory for each active programme (or process).

Some UNIX files begin with a period, for example, **.cshrc** or **.login.** Files that begin with a period will not appear in a normal directory listing and are usually UNIX environment and application setup files.

11.2.13 File and Directory permissions

It is important to protect user. The UNIX operating system maintains information, known as permissions, for every file and directory on the system. Every file or directory in a UNIX file system has three types of permissions (or protections) that define whether certain actions can be carried out. The permissions are :

1. read (**r**) A user who has read permission for a file may look at its contents or make a copy of it. For a directory, read permission enables a user to find out what files are in that directory.
2. write (**w**) A user who has *write* permission for a file can alter or remove the contents of that file. For a directory, the user can create and delete files in that directory.
3. execute (**x**) A user who has *execute* permission for a file can cause the content of that file to be executed (provided that it is executable). For a directory, execute permission allows a user to change to that directory.

For each file and directory, the read, write, and execute permissions may be set separately for each of the following classes of users :

- User (u) : The user who owns the file or directory.
- Group (g) : Several users purposely grouped together so that they can share access to each other's files.
- Others (o) : The remainder of the authorized users of the system.

The primary command that displays information about files and directories is **ls.** The **-l** option will display the information in a long format. You can get information about a single UNIX file by using **ls–i** *filename.*

Each file or subdirectory entry in a directory listing obtained with the -l option consists of seven fields :

1. permission mode,
2. link count,
3. owner name,
4. group name,
5. file size in bytes,
6. time of last modification, and
7. the filename (the group name appears only if the "g" flag is also specified, as in **ls–ig**).

The first 10 characters make up the mode field. If the first character is a "d" then the item listed is a directory; if it is a "–" then the item is a file; if it is an "l" then it is a link to another file.

Characters 2 through 4 refer to the owner's permissions, characters 5 through 7 to the group's permissions (groups are defined by the system administrator), and the last three to the general public's permissions. If a particular permission is set, the appropriate letter appears in the corresponding position; otherwise, a dash indicates that the permission is not given.

The second field in the output from **ls–i** is the number of links to the file. In most cases it is one, but other users may make links to your files. thus increasing the link count. A special warning to people using links to other people's files : your "copies" of their files can be counted against them by the file quota system available on certain UNIX variants. That is why making links other than symbolic links to other people's files is strongly discouraged. The third field gives the userid of the owner of the file. The group name follows in the fourth field (if the **-g** option is used in conjunction with **-l**). The next two fields give the size of the file (in bytes) and the date and time at which the file was last modified. The last field gives the name of the file.

```
ls -i myfile
-rw-r--r-- 1 owner 588 Jul 15 14 : 39 myfile
```

A file's owner can change any or all of the permissions with the **chmod** (**ch**ange **mod**e) command. The **chmod** command allows user to dictate the type of access permission that user want each file to have. In the previous example the current permissions for **myfile** are read for everybody, write for the owner, and execute by no one.

The arguments supplied to **chmod** and a symbolic specification of the changes required, folowed by one or more filenames. The specification consists of whose permissions are to be changed : **u** for user (owner), **g** for group, **o** for others, or some combination thereof (**a** (**all**) has the same effect as **ugo**), how they are to be changed (+ adds a permission, -removes a permission, and = sets the specified permissions, removing the other ones) and which permission to add or remove (**r** for read, **w** for write, and **x** for execute). For example, to remove all the permissions from **myfile :**

```
chmod a-rwx myfile
ls-i myfile
------- 1 owner 588 Jul 15 14 : 41 myfile
```

(Note : **chmod a = myfile** achieves the same effect.)

To allow read and write permissions for all users :

```
chmod ugo+rw myfile
ls-i myfile
-rw-rw-rw 1 owner 588 Jul 15 14 : 42 myfile
```

To remove write permission for your groups and other users :

```
chmod go-w myfile
ls-i myfile
-rw-r-r-1 owner 588 Jul 15 14 : 42 myfile
```

Finally, to allow only read permission to all users :

```
chmod a=r myfile
ls-i myfile
-r-r-r- 1 owner 58 Jul 15 14 : 43 myfile
```

Now the file is protected by allowing only read access; it cannot be written to or executed by anyone, *including user.* Protecting a file against writing by its owner is a safeguard against accidental overwriting, although *not* against accidental deletion.

11.2.14 Wildcard Characters

Using wildcard characters that allow you to copy, list, move, remove, etc. items with similar names is a great help in manipulating files and directories.

1. The symbol? will match any single character in that position in the file name.
2. The symbol * will match zero or more characters in the name.
3. Characters enclosed in brackets [and] will match any one of the given characters in the given position in the name. A consecutive sequence of characters can be designated by [char char].

11.2.15 Processes

Every command or programme runing under UNIX is called a *process.* A sequence of related processes is called a *job.* Your applications and even your shell itself are processes. The windowing system is also a process, or a collection of processes. The UNIX kernel manages the processes on the system, usually without distinguishing among them. UNIX is a multi-tasking system—it allows you to continue to work in the foreground while running one or more jobs in the background. It also runs the processes of many users simultaneously. You could even log off and come back later if the background jobs do not require interaction with you.

11.2.16 Running Background Jobs

Putting a programme into an unattended state where it continues to execute is referred to as putting it (the process or job) into the *background.* (Running a programme on one machine and displaying its output on another via a windowing system like X is not considered backgrounding the job.). Adding an & (ampersand) at the end of the command line instructs UNIX to run the job in the background.
jobname&

11.3 Windows NT

Windows NT is a family of operating systems produced by Microsoft, and was succeeded by Windows 2000 (still based on Windows NT). Although the company decided to drop "NT" from the Windows naming scheme for marketing, it internally still uses Windows NT as a means of reference. The latest available operating systems based on NT are Windows XP and Windows Server 2003.

11.3.1 Development

When development started in November 1988, Windows NT was to be known as OS/2 3.0, the third version of the operating system developed jointly by Microsoft and IBM. Microsoft hired a group of developers from Digital Equipment Corporation led by Dave Cutler to build Windows NT and many elements reflect earlier DEC experience with VMS and RSX-11. The OS is designed to run on multiple instruction set architectures, with the kernel separated from the hardware by a hardware abstraction layer. APIs are implemented as subsystems atop the publicly undocumented native API; it was this that allowed the late adoption of the Windows API. Originally a microkernel design, subsequent releases have integrated more functions into the kernel for better performance. Windows NT was the first operating system to use Unicode internally.

11.3.2 Releases

Windows NT Releases

NT Ver.	Marketing Name	Editions	Release date	Build
NT 3.1	Windows NT 3.1	Workstation, Advanced Server	July 1993	528
NT 3.5	Windows NT 3.5	Workstation, Server	September 1994	807
NT 3.51	Windows NT 3.51	Workstation, Server	May 1995	1057

NT Ver.	Marketing Name	Editions	Release date	Build
NT 4.0	Windows NT 4.0	Workstation, Server, Enterprise Edition, Terminal Server, Embedded	July 1996	1381
NT 5.0	Windows 2000	Professional, Server, advanced Server, Datacenter Server	February 2000	2195
NT 5.1	Windows XP	Home, Professional Media Center (2004 & 2005), Tablet PC, Starter, Embedded, N	October 2001	2600
NT 5.2	Windows Server 2003	Standard, Enterprise, Datacenter, Web, XP Pro x64	March 2003	3790
NT 6.0	Windows Vista	Starter, Home Basic, Home Premium, Professional, Small Business, Enterprise, Ultimate (x64 editions will be available too)	2006 (expected)	Unknown
NT 6.0 +	Longhorn (codename)	Server Unknown	2007 (expected)	Unkown

The first release was given version number 3.1. The NT version is no longer marketed, but is said to reflect the degree of changes to the core of the operating system. The build number is an internal figure used by Microsoft's developers.

11.3.3 Suported platforms

Windoes NT 3.1 ran on Intel IA-32 x86, DEC Alpha, and MIPS R4000 processors. Windows NT 3.51 added support for PowerPC processors. Intergraph Corporation

ported Windows NT to its Clipper architecture and later SPARC, but neither version was sold to the public. Windows NT 4.0 was the last major release to support Alpha, MIPS, or PowerPC, though development of Windows 2000 for Alpha continued until 1999, when Compaq stopped support for Windows NT on that architecture. Only 2 of the Windows NT 4.0 variants (IA-32 and Alpha) have a full set of service packs available. All of the other ports done by 3rd parties (Motorola, Intergraph, etc.) have few, if any, publicly available updates.

Windows XP 64-Bit, windows Server 2003 Enterprise, and Windows Server 2003 Datacenter support Intel's IA-64 processors. Microsoft had released four editions for the AMD 64 : Windows XP Professional x64 Edition, Windows Server 2003 Standard x64 Edition, Windows Server 2003 Enterprise x64 Edition and Windows Server 2003 Datacenter x64 Edition.

The Xbox uses a heavily modified and stripped down Windows 2000 kernel. This kernel was heavily modified again for the Xbox 360 which runs on Power PC. This version is not for sale, and little is generally known about it.

11.3.4 Windows NT File systems

It supports different file systems. The following table presents an overview of supported file system :

File System	Description
FAT File system	Used with DOS, it can only support partitions up to 4 G. No spaces are allowed in the file name.
FAT 32 or VFAT File System	VFAT-Virtual File Allocation Table introduced by Windows 95. Some documentation says NTWS cannot use FAT32. • Filenames up to 255 characters long. • Names begin with a letter and exclude " / \ [ ] : ; | = , ^ * ? • The last part is the extension but spaces can be used • It supports file attributes used by DOS such as read-only, archive, system, and hidden. • Won't support running POSIX applications. RISC computers can only boot from FAT files systems. FAT file systems support dual booting of operating systems. FAT partitions provide no local security, only share level security. • Filenames up to 255 characters long • File names preserve case but are not case sensitive. • Exclude " / \ < > : | * ?

File System	Description
NTFS File System	• Supports built in file compression as a file attribute. Compression is applied to files in a folder if that folder has its compression attribute set. Also optionally sub folders and their contents may be compressed. Compression is not supported if the cluster size is above 4K in size. Moved files retain their compression attribute, but if they are copied they will assume the compression attribute of the target folder. • Provides automatic transaction tracking of disk activity for correcting corrupt or failed operations. • Supports auditing. • Provides sector sparing. • There is a recycle bin for each user • Windows 16 bit and DOS environments can't use this filesystem. • A master file table is used to save individual file, boot sector, disk structure, and file recovery information. • Automatically makes 11 character DOS file names. When the first 8 characters of long filenames match, the first four DOS file names use the first for characters of the long name, the ~and 1, then ~2, etc. After the fourth duplicate name, the first two characters are used, then the next four characters are hashed, then the ~ character then a number. The first two duplicate file names may be: DOCU~1.DOC and DOCU~2.DOC. The long extension is used as part of the extension for the 8.3 filename alias. Directory entries used by long filenames include 1 for the 8.3 alias and 1 for each 13 characters in the long filename. • Provides file logging ability and file recovery. • Supports POSIX. • Maximum file or partition size of 16 exabytes. • Supports file sharing with MacIntosh clints.

11.3.5 Features

The section presents various features of Windows NT

Microsoft Operating System	Features
Windows NT Server	• Supports 256 clients on the RAS or DUN server. • Supports unlimited clients for resource access • Supports unlimited outbound connections. • Can support up to 4 processors and 32 processors if OEM version of the system. • Peer Web Services (PWS) provided for small workgroup web publishing. • Can import or export directory replication. • Includes System Management Server, SNA Server, and SQL Server, IIS (Internet Information Server), DNS Name Server for fully qualified internet domain names. • Apple Share server that allows access to MacIntosh and can act as a gateway to Netware servers • Supports DHCP, BOOTP, WINS and multi-protocol routing. • Can remotely reboot Windows 95 clients. • Supports disk fault tolerance with mirroring, striping and hot fixing • Supports directory replication which replicates directories and files to other computers on the network. • Support RAID
Windows NT Workstation	• Can support one remote dial in session as a RAS or DUN server. • Supports up to 10 clients for resource access. • Any number of peer to peer outbound connections. • Can support up to 2 processors and 32 processors if OEM version of the system. • Peer web servers. • Can import directory replication and not export. • No disk fault tolerance support.
Windows NT Server & Workstation	• A 32 bit operating system • Supports 16 and 32 bit applications. • There is no direct hardware access. • Provides memory protection. • Logon is mandatory. The CTRL-ALT-DEL key combination is used to logon since it disables user mode and TSR programs in NT.

	• Expanded memory support-Supports up to 4 GB of RAM. • Preemptive Multitasking - Threads may be assigned relative priorities. • Expanded File system support-Supports the New Technology File system (NTFS) for increased security and can track disk transactions which will aid in the recovery of data should loss of power occur. It supports up to 16 exabytes of disk space. Supports FAT, VFAT, and the CDFS file systems. • Local and shared (extensive) security. • Multiple platform support-The Hardware Abstraction Layer (HAL) isolates the operating system from the platform. Supports both CISC and RISC platforms. • Can operate multiple microprocessors on one computer using symmetric multiprocessing. • Can run OS2 1.3 but not 2.0

11.3.6 Windows NT Structure

NT runs in two modes :

1. Kernel mode (Ring 0)-Executive which runs in protected memory mode with full privileges.
2. User mode (Ring 2) - Runs with privileges to access its own memory area.

User applications and environmental subsystems execute in this mode. Applications are allocated a virtual 4GB of memory with 2 for the user and 2 for executive services. NT is modular in nature allowing it to have cross platform portability due primarily to the HAL module described below. The NT Architecture has 5 layers.

3. Application - Runs in user mode.
4. Subsystems-Runs in user mode.
5. Executive Services-Runs in Kernel mode.
6. Kernel-Runs in kernel mode.
7. HAL-Runs in kernel mode.

The NT architecture model in more detail, from the top down :

1. User Level-The environmental subsystem and user applications execute at this level which runs in Ring 3, a non-privileged processor mode. User mode code can be preempted, is pageable, and can be context switched. User applications must use executive services to access devices or memory.
2. Kernel Level also called executive services run in the protected mode of the processor ring. Cannot be paged or context switched.

11.3.7 A Comparison of LINUX and WINDOWS

Parameters	WINDOWS	LINUX
Flavors	All the flavors of Windows come from Microsoft, Windows has two main lines: "Win9x", which consists of Windows 95, 98, 98SE and Me, and "NT class" which consists of Windows NT, 2000 and XP.	The various distributions of Linux come from different companies (i.e. Linspire, Red Hat, SuSE, Mandrake, Knoppix, Slackware). There are many special purpose
Customization	Linux is customizable in a way that Windows is not. Microsoft line of division is Professional and Home edition.	versions of Linux above and beyond the full distributions described above. For example. NASLite is a version of Linux that runs off a single floppy disk and converts an old computer into a file server. This ultra small edition of Linux is capable of networking; file sharing and being a web server.
GUI	The Windows GUI has changed from Windows 3.1 to Windows 95 (drastically) to Windows 2000 (slightly) to Windows XP (fairly large). Windows XP has a themes feature that offers some customiza-tion of the look and feel of the GUI. The Windows GUI is an integral component of the OS.	Typically provides two GUIs, KDE and Gnome. The GUI is optional. Speed, efficiency and reliability are all increased by running a server instance of Linux without a GUI, something that server versions of Windows cannot do. The detached nature of the Linux GUI makes remote control and remote administration of a Linux computer simpler and more natural than a Windows computer.
Text Mode Interface	Windows users sometimes call it a DOS prompt. Each version of Windows has a single command interpreter, but the different flavors of Windows have different interpreters. In general, the command interpreters in the Windows 9x series are very similar to each other and	Linux users refer to it as a shell. Linux like all versions of Unix, supports multiple command interpreters, but it usually uses one called BASH (Bourne Again Shell). Others are the Korn shell.

Parameters	WINDOWS	LINUX
	the NT class versions of Windows (NT, 2000, XP) also have similar command interpreters. There are however differences between a Windows 9x command interpreter and one in an NT class flavor of Windows.	
Cost	For desktop or home use Windows is expensive. Microsoft allows a single copy of Windows to be used on only one computer. Starting with Windows XP, they use software to enforce this rule (activation).	For desktop or home use. Linux is very cheap or free. For server use, Linux is very cheap compared to Windows. Once you have purchased Linux, you can run it on any number of computers for no additional charge.
Installation	Installing Windows from scratch is much easier than installing Linux from scratch	The different distributions of Linux have their own installation programs. However, installation of multiple OS is feasible.
Running from CD	Windows can not run from a CD.	Normally Linux also runs from a hard disk. but there are quite a few versions of Linux that run *completely* form a CD without having to be installed to a hard disk (the term for this is a "Live" CD). Among the Linux distros that have a CD-only version are Knoppix, and SuSE (called Live-Eval) etc.
Application Software	There is more application software available for Windows.	
Viruses & Spyware	The vast majority of all malicious software (of all types) runs on Windows.	
Users & Passwords	Both Linux and Windows 2000/XP require a userid and password and boot time.	Both Linux and Windows can group users into groups and assign privileges to the group rather than to each individual user.
Software Restrictions		

Parameters	WINDOWS	LINUX
Supported Hardware	More hardware works with Windows than works with Linux.	
Hardware Support for OS		Linux runs on many different hard-ware platforms, not so with Windows.
Clustering		Linux has an edge here. It has been used to make enormous clusters of computers.
Multiple Users	Window is not a multi-user system. That is, Windows is designed to be used by one person at a time. Databses running under Windows allow concurrent access by multiple users, but the Operating System itself is designed to deal with a single human being at a time. Windows, of course, can run many progrms concurrently, as can Linux. There is a multi-user version of Windows called Terminal Server but this is not the Windows pre-installed on personal computers.	Linux is a multi-user system. Linux, like all Unix variants, is designed to handle multiple concurrent users.
Networking	They both do TCP/IP	Linux can also do Windows networking, which means that a Linux computer can appear on a network of Windows computers and share its files and printers. Linux machines can participate on a Windows based network and vice versa.
Hard Disk Partition	Windows must boot from a primary partition. Windows must boot from the first hard disk.	Linux can boot from either a primary partition or a logical partition inside an extended partition. Linux can boot from any hard disk in the computer.
File Systems	Windows uses FAT12, FAT16, FAT32 and/or NTFS with NTFS almost always being the best choice. Linux also has a number of its own native file systems.	The default the system for Linux used to be ext2, now it is typically ext3.
File Hierarchy	Windows uses a volume-based file hierarchy. Windows uses letters of the alphabet to represent different devices and different hard disk partitions.	Linux uses a unified scheme. In Linux all directories are attached to the root directory, which is identified by a forward-slash, "/".
Case Sensitivity	Not case sensitive with commands and file/folder names	Case sensitive with commands and file/directory names

References and Further Readings

1 2001 Introduction to UNIX : course outline (http://www.doc.ic.ac.uk/~wjk/UnixIntro/Lecture1.html). Visited last : 21/10/2005

2 2005 LINUX vs. Windows. (http://www.michaelhorowitz. com/Linux.vs. Windows. html). Visited last : 25/10/Windows.

3 2005 Windows NT. (http://en.wikipedia. org/wiki/Windows NT). Visited last : 25/10/2005

11.4 Exercise

1. Discuss architecture of Linux.
2. What is a multi tasking operating system ?
3. Discuss six basic elements of Unix.