

# 1 What is an Operating System ? Explain it's functions in detail.

An Operating system (OS) is a software which acts as an interface between the end user and computer hardware.

Every general-purpose computer must have an operating system to run other programs and applications. Computer operating systems perform basic tasks, such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the storage drives, and controlling peripheral devices, such as printers.

For large systems, the operating system has even greater responsibilities and powers. It is like a traffic cop — it makes sure that different programs and users running at the same time do not interfere with each other. The operating system is also responsible for security, ensuring that unauthorized users do not access the system.

Most software applications are written for operating systems, which lets the operating system do a lot of the heavy lifting. For example, when you run Minecraft, you run it on an operating system. Minecraft doesn't have to know exactly how each different hardware component works. Minecraft uses a variety of operating system functions, and the operating system translates those into low-level hardware instructions. This saves the developers of Minecraft and every other program that runs on an operating system a lot of trouble.

## 1.1 Functions of an Operating System:

- **Process Management:** Process management helps OS to create and delete processes. It also provides mechanisms for synchronization and communication among processes.
- **Memory Management:** Memory management module performs the task of allocation and de-allocation of memory space to programs in need of this resources.
- **File Management:** It manages all the file-related activities such as organization storage, retrieval, naming, sharing, and protection of files.
- **Device Management:** Device management keeps tracks of all devices. This module also responsible for this task is known as the I/O controller. It also performs the task of allocation and de-allocation of the devices.
- **I/O System Management:** One of the main objects of any OS is to hide the peculiarities of that hardware devices from the user.
- **Secondary-Storage Management:** Systems have several levels of storage which includes primary storage, secondary storage, and cache storage. Instructions and data must be stored in primary storage or cache so that a running program can reference it.
- **Security:** Security module protects the data and information of a computer system against malware threat and authorized access.
- **Command interpretation:** This module is interpreting commands given by the and acting system resources to process that commands.
- **Networking:** A distributed system is a group of processors which do not share memory, hardware devices, or a clock. The processors communicate with one another through the network.
- **Job accounting:** Keeping track of time and resource used by various job and users.
- **Communication management:** Coordination and assignment of compilers, interpreters, and another software resource of the various users of the computer systems.

## 2 What is DOS ?

It is an acronym for **Disk Operating System**, in a general sense, DOS refers to just about any operating system. More commonly, it describes the operating system purchased by Microsoft Corporation from a system known as QDOS (86-DOS) in 1981 for IBM's line of personal computers. Though syntactically distinct, DOS shares similarities with a Unix shell. It has a command-line interface and analogs to many common Unix commands. However, DOS is a 16-bit, single-user operating system that does not support multi-tasking. It is far easier to administer than Unix, but less powerful. Compared to graphical interfaces such as Windows and Mac OS X, it's also not particularly user-friendly.

A disk operating system will load from a floppy disk each time a computer starts, and will access that disk for software to complete operations. As operating systems became more complicated and took up more space, they began to be permanently installed on hard drives, which are faster and more reliable than floppy disks, and can store more data. This was encouraged by a steady drop in hard drive prices.

The first versions of Windows (through Windows 95) actually ran on top of the DOS operating system. This is why so many DOS-related files (such as .INI, .DLL, and .COM files) are still used by Windows. However, the Windows operating system was rewritten for Windows NT (New Technology), which enabled Windows to run on its own, without using DOS. Later versions of Windows, such as Windows 2000, XP, and Vista, also do not require DOS.

With the development of Windows, MS-DOS has faded in importance. However, you can still run some DOS commands at the command prompt in current versions of Windows, which can be useful in situations where a graphical interface is less efficient.

Several DOS alternatives and/or enhancements are available, including FreeDOS and DR-DOS. FreeDOS is a version of DOS that is freely distributed under the GNU General Public License. Though not completely compatible with MS-DOS, it will run many DOS programs.

## 3 What is the difference between Windows and Linux ?

**Windows** is a licensed operating system in which source code is inaccessible. It is designed for the individuals with the perspective of having no computer programming knowledge and for business and other commercial users. It is very simple and straightforward to use.

Windows is extensible, portable and assists multiple operating environments, symmetric multiprocessing and client-server computing. It offers integrated caching, virtual memory, and preemptive scheduling.

**Linux** is a free and open source operating system based on Unix standards. It provides programming interface as well as user interface compatible with Unix based systems and provides large variety applications. A Linux system also contains many separately developed elements, resulting in Unix system which is fully compatible and free from proprietary code.

The traditional monolithic kernel is employed in Linux kernel for performance purpose, but its modular feature allows most drivers to dynamically loaded and unloaded at runtime. Linux protects processes and is a multiuser system. Interprocess communication is supported by both of mechanisms such as message queue, shared memory and semaphore.

An abstract layer is used in Linux to govern the different file systems, but to users, the file system looks like a hierarchical directory tree. It also supports networked, device-oriented and virtual file systems. Disk storage is accessed through a page cache which is unified with the virtual memory systems. To minimize the duplication of shared data among different processes the memory management system uses page sharing and copy-on-write.

### 3.1 Key Differences Between Linux vs Windows

- Linux is open source operating system whereas Windows OS is commercial.
- Linux has access to source code and alters the code as per user need whereas Windows does not have access to source code.
- Linux will run faster than windows latest editions even with a modern desktop environment and features of the operating system whereas windows are slow on older hardware.

- Linux distributions don't collect user data whereas Windows collect all the user details which lead to privacy concern.
- Linux is more reliable than windows as in Linux we can kill application if they hung through "kill -9" command whereas, in windows, we need to try multiple times to kill it.
- Linux supports a wide variety of free software's than windows but windows have a large collection of video game software.
- In Linux software cost is almost free as all programs, utilities, complex applications such as open office are free but windows also have many free programs and utilities but most of the programs are commercial.
- Linux is highly secure because it's easy to identify bugs and fix whereas Windows has a large user base and becomes a target for developers of viruses and malware.
- Linux is used by corporate organizations as servers and operating system for security purpose at Google, Facebook, twitter etc. whereas windows are mostly used by gamers and business users.
- Linux and windows have same priority over hardware and driver support in the present situation.

## 4 Write all the External and Internal commands.

**Internal commands** are something which is built into the shell. For the shell built in commands, the execution speed is really high. It is because no process needs to be spawned for executing it. For example, when using the "cd" command, no process is created. The current directory simply gets changed on executing it.

**External Commands** are not built into the shell. These are executables present in a separate file. When an external command has to be executed, a new process has to be spawned and the command gets executed. For example, when you execute the "cat" command, which usually is at `/usr/bin`, the executable `/usr/bin/cat` gets executed.

### 4.1 Internal Commands:

```

job_spec [&]
history [-c] [-d offset] [n] or history -anrw [filename] or history >
(( expression ))
if COMMANDS; then COMMANDS; [ elif COMMANDS; then COMMANDS; ]... [ e>
. filename [arguments]
jobs [-lnprs] [job_spec ...] or jobs -x command [args]
:
kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill ->
[ arg... ]
let arg [arg ...]
[[ expression ]]
local [option] name[=value] ...
alias [-p] [name[=value] ... ]
logout [n]
bg [job_spec ...]
mapfile [-d delim] [-n count] [-O origin] [-s count] [-t] [-u fd] [->
bind [-lpsvPSVX] [-m keymap] [-f filename] [-q name] [-u name] [-r ke>
popd [-n] [+N | -N]
break [n]
printf [-v var] format [arguments]
builtin [shell-builtin [arg ...]]
pushd [-n] [+N | -N | dir]
caller [expr]

```

```

pwd [-LP]
case WORD in [PATTERN [| PATTERN]...) COMMANDS ;;]... esac
read [-ers] [-a array] [-d delim] [-i text] [-n nchars] [-N nchars] >
cd [-L|[-P [-e]] [-@]] [dir]
readarray [-d delim] [-n count] [-O origin] [-s count] [-t] [-u fd] >
command [-pVv] command [arg ...]
readonly [-aAf] [name[=value] ...] or readonly -p
compgen [-abcdefgjkuv] [-o option] [-A action] [-G globpat] [-W word]>
return [n]
complete [-abcdefgjkuv] [-pr] [-DEI] [-o option] [-A action] [-G glo>
select NAME [in WORDS ... ;] do COMMANDS; done
compgpt [-o|+o option] [-DEI] [name ...]
set [-abefghkmnptuvxBCHP] [-o option-name] [--] [arg ...]
continue [n]
shift [n]
coproc [NAME] command [redirections]
shopt [-pqsu] [-o] [optname ...]
declare [-aAfFgilnrtux] [-p] [name[=value] ...]
source filename [arguments]
dirs [-clpv] [+N] [-N]
suspend [-f]
disown [-h] [-ar] [jobspec ... | pid ...]
test [expr]
echo [-neE] [arg ...]
time [-p] pipeline
enable [-a] [-dnps] [-f filename] [name ...]
times
eval [arg ...]
trap [-lp] [[arg] signal_spec ...]
exec [-cl] [-a name] [command [arguments ...]] [redirection ...]
true
exit [n]
type [-afptP] name [name ...]
export [-fn] [name[=value] ...] or export -p
typeset [-aAfFgilnrtux] [-p] name[=value] ...
false
ulimit [-SHabcdefiklmnpqrstuvxPT] [limit]
fc [-e ename] [-lnr] [first] [last] or fc -s [pat=rep] [command]
umask [-p] [-S] [mode]
fg [job_spec]
unalias [-a] name [name ...]
for NAME [in WORDS ... ] ; do COMMANDS; done
unset [-f] [-v] [-n] [name ...]
for (( exp1; exp2; exp3 )); do COMMANDS; done
until COMMANDS; do COMMANDS; done
function name { COMMANDS ; } or name () { COMMANDS ; }
variables - Names and meanings of some shell variables
getopts optstring name [arg]
wait [-fn] [id ...]
hash [-lr] [-p pathname] [-dt] [name ...]
while COMMANDS; do COMMANDS; done
help [-dms] [pattern ...]
{ COMMANDS ; }

```

## 4.2 External Commands:

NF	debconf-copydb
VGAAuthService	debconf-escape
'['	debconf-set-selections
aa-enabled	debconf-show
aa-exec	delpart
acpi_listen	delv
add-apt-repository	dh_bash-completion
addpart	diff
apport-bug	diff3
apport-cli	dig
apport-collect	dircolors
apport-unpack	dirmngr
apropos	dirmngr-client
apt	dirname
apt-add-repository	do-release-upgrade
apt-cache	dpkg
apt-cdrom	dpkg-deb
apt-config	dpkg-divert
apt-extracttemplates	dpkg-maintscript-helper
apt-ftparchive	dpkg-query
apt-get	dpkg-split
apt-key	dpkg-statoverride
apt-mark	dpkg-trigger
apt-sortpkgs	du
arch	dumpkeys
at	eatmydata
atq	ec2metadata
atrm	edit
automat-visualize3	editor
awk	eject
b2sum	enc2xs
base32	encguess
base64	env
basename	envsubst
bashbug	eqn
batch	ex
bc	expand
bootctl	expiry
bsd-from	expr
bsd-write	factor
busctl	faillog
c_rehash	gpg-zip
cal	gpgconf
calendar	gpgparsemail
captoinfo	gpgsm
catchsegv	gpgsplit
catman	gpgv
cautious-launcher	gpics
cftp3	groff
chac1	grog
chage	grops
chardet3	grotty
chardetect3	groups
chatr	growpart

chcon	gtbl
check-language-support	h2ph
chfn	h2xs
chrt	hd
chsh	head
ckbcomp	helpztags
ckeygen3	hexdump
cksum	host
clear	hostid
clear_console	hostnamectl
cloud-id	htop
cloud-init	hwe-support-status
cloud-init-per	i386
cmp	iconv
codepage	id
comm	ischroot
compose	iscsiadm
conch3	join
corelist	json_pp
cpan	jsondiff
cpan5.26-x86_64-linux-gnu	jsonpatch
crontab	jsonpointer
csplit	jsonschema
ctail	kbdinfo
ctstat	kbxutil
curl	keep-one-running
cut	kernel-install
dbus-cleanup-sockets	killall
deallocvt	less
deb-systemd-helper	lessecho
deb-systemd-invoke	lessfile
debconf	lesskey
debconf-apt-progress	lesspipe
debconf-communicate	lexgrog
nsupdate	sftp
ntfsdecrypt	sg
numfmt	sha1sum
od	sha224sum
on_ac_power	sha256sum
openssl	sha384sum
pager	sha512sum
partx	shasum
passwd	showconsolefont
paste	showkey
pastebinit	shred
patch	shuf
pathchk	skill
pbget	slabtop
pbput	slogin