



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

EMBEDDED PROGRAMMING

ECE4025 – C1

Allen Ben Philipose – 18BIS0043 – Task 4 (Q1,4,5,7,9)

I

Explain Command Prompt basic command

Command prompt is used in a text based or "command line" interface such as UNIX, Linux or DOS. It is a symbol or series of characters at the beginning of a line that indicates that the system is ready to take input from the user. It prompts the user for a command.

The command prompt is often preceded by the current directory of the system the user is working with.

The default prompt in DOS would be the line `C:\` which indicates the user is working at the root level of the main C drive.

The default prompt in UNIX would be the line `~ allen$` where "allen" is the name of the current user. The tilde (~) indicates that the current directory is the user's home folder.

A user can type commands in the command prompt such as `cd /` which means change directory to root folder. The "cd" command allows the user to browse through different directories of files on a hard disk or network.

There are dozens of other commands that a user can type which can be used to list, move, delete, copy files, run programs, or perform other operations. While "cd" command is the same both in DOS and UNIX, many other commands have different syntax.

Since a command requires a specific input, knowing syntax is very essential to use CLI - command line interface

GUI - Graphical User Interface has replaced CLI as the primary input medium which eliminates the requirement of learning programming syntax by an average user, but having experience with CLI would help in achieving special and quick operations which would be tougher or impossible to do in Graphical user Interface.

cmd command

Description

call	calls a batch file from another one
cd	change directory
cls	clear screen
cmd	start command prompt
color	change console color
date	show/set date
dir	list directory content
echo	text output
exit	Exits command prompt
find	find files
hostname	display hostname
pause	pauses execution of batch file
runas	starts program as another user
shutdown	shutdown computer
sort	sort screen output
start	start an own video
taskkill	terminate process
tasklist	display applications
time	display/edit system time
timeout	wait only time
title	set title from input to prompt
ver	display OS version
w32tm	setting time synchronization

II Navigating file system, finding files, working with folders and text editing

Print working directory (pwd)

The command prints the current directory, telling where you are currently located in the filesystem. This command always prints out the absolute filepath from the root drive.

Absolute path is the full path from root directory (/) and it is best practice to use absolute paths when its used inside scripts.

`/usr/bin/ls`

Relative path is relative to the current working directory and can be displayed using dots

`../usr/bin/ls`

Change directory (cd)

The command lets you change to a different directory. The default working directory is the home directory.

`/home/allen`

Absolute and relative paths can be better understood with the cd command. Similar to pwd, dots are used to represent relative paths to a particular directory.

`$ cd ..` Represents the directory, one level above the current working directory

`$ cd /home` Absolute path to home directory

`$ cd /` Absolute path to root directory

When the working directory is deeply nested inside the filesystem, the `(~)` is used to immediately return to the home directory.

The `(-)` has a similar functionality and the command takes user to the previous folder/working directory.

`Nano editor` in Linux system is used to edit files and can be accessed by the `nano` command, in linux terminal

- `esc` normal mode
- `i` insert mode
- `q!` exit without saving
- `wq!` save and exit
- `w` save

`cat allen.txt` command can be typed in Linux terminal to read and print the contents of a file. This is the most commonly used method but there are other alternatives in bash script.

You can read a file line by line from command line using the code

```
while read line; do command; done <input.file
```

Here, while loop will reach each line of the file and store the content of the line in the variable `$line` which will be printed later.

Using Bash script we can construct this code,

```
#!/bin/bash
file = 'allen.txt'
i = 1
while read line; do
    echo "Line No. $i: $line"
    i = $((i+1))
done < $file
```

to print each line of the file separately

III Different compression and archiving tools used in Linux

Compression: Way of reducing the size of a file on a disk using different algorithms and mathematical calculations. Files are formatted in certain ways that make their general structure somewhat predictable, even if their content varies. Furthermore, content itself is often repeated

- Lossy conversion: Information lost to diminish Quality for saving space
- Lossless conversion: No data lost

Archiving: Generally implies backing it up and saving it to a secure location, often in a compressed format. Data from servers was often backed up onto tape archives, which are magnetic tapes used to store sequential data and to do that efficiently, "tar" program was created. This helped to address and manipulate many files in the filesystem with intact permissions and metadata. It helps to conveniently distribute, store, back up and manipulate groups of related file.

Most of compression and archive tools can be operated via command line interface in linux. Short commands can quickly compress data files to save storage space and bandwidth when sent over a network.

→ gzip :

GNU zip is one of the most commonly used compression methods. This tool plays a major role in web development, which is based on the deflate algorithm. Today, the application program in C can be used for extracting and packaging files in Linux, windows and MacOS

gzip builds 32,000 bytes data blocks which makes it feel obsolete in modern compression programs.

```
allen@linux: ~$ gzip allen/desktop/test.odt
```

Fast compression process, standard popular web server software but small block size and low compression ratio

→ bzip2 :

Loss free and high quality compression files with 3 layered compression method

- Burrow's wheeler transformation (900000 bytes)
- Move-to-front transformation
- Huffman coding

With the help of bzip2 recover, partially damaged archives can be atleast extracted and unpacked. This along with strong compression rates benefitted users but the process was very slow.

→ p7zip:

Uses Lempel-Ziv-Markov Algorithm (LZMA) which is a further development on deflate algorithm. It has password protection and an optional encryption using AES-256. It has an excellent ratio of compression and duration but very high system requirements

→ lzop:

Lempel-ziv-obershammer algorithm which offered very quick compression with high portability but lower compression speeds.

→ tar :

Archives only

```
tar -cf archive.tar test.txt test2.txt
```

Example

IV Types of files and file management in Linux

How file management is done in Linux ?

- Tree like structures of directories also called as folders
- Most operations are performed on files
- Command Line Interface gives a greater control during file management actions

7 operations on files in linux:

- Files listing
- Creating files
- Displaying file contents
- Copying a file
- Moving a file
- Renaming a file
- Deleting a file

3 Types of files in Linux:

- Regular files: Text, binary, images, etc...
Such files can be created using the touch command, being majority of files in Linux/Unix Systems
- Directories: Files that store a list of file names and other information related to those files. Referred to in windows as folders.
 - / → Root directory at base of system
 - /home/ → User's home directory

→ Special files: Represents a real physical peripheral device such as a printer which is used for I/O operations.

One operation on files: Creating files

For creating a file, we can use the `touch` command. It will create and open a new blank file if the filename doesn't exist. If existing, the old file will not be affected.

Example:

```
linux@allen: $ touch cat2.txt
```

```
linux@allen: $ ls
```

```
cat2.txt
```

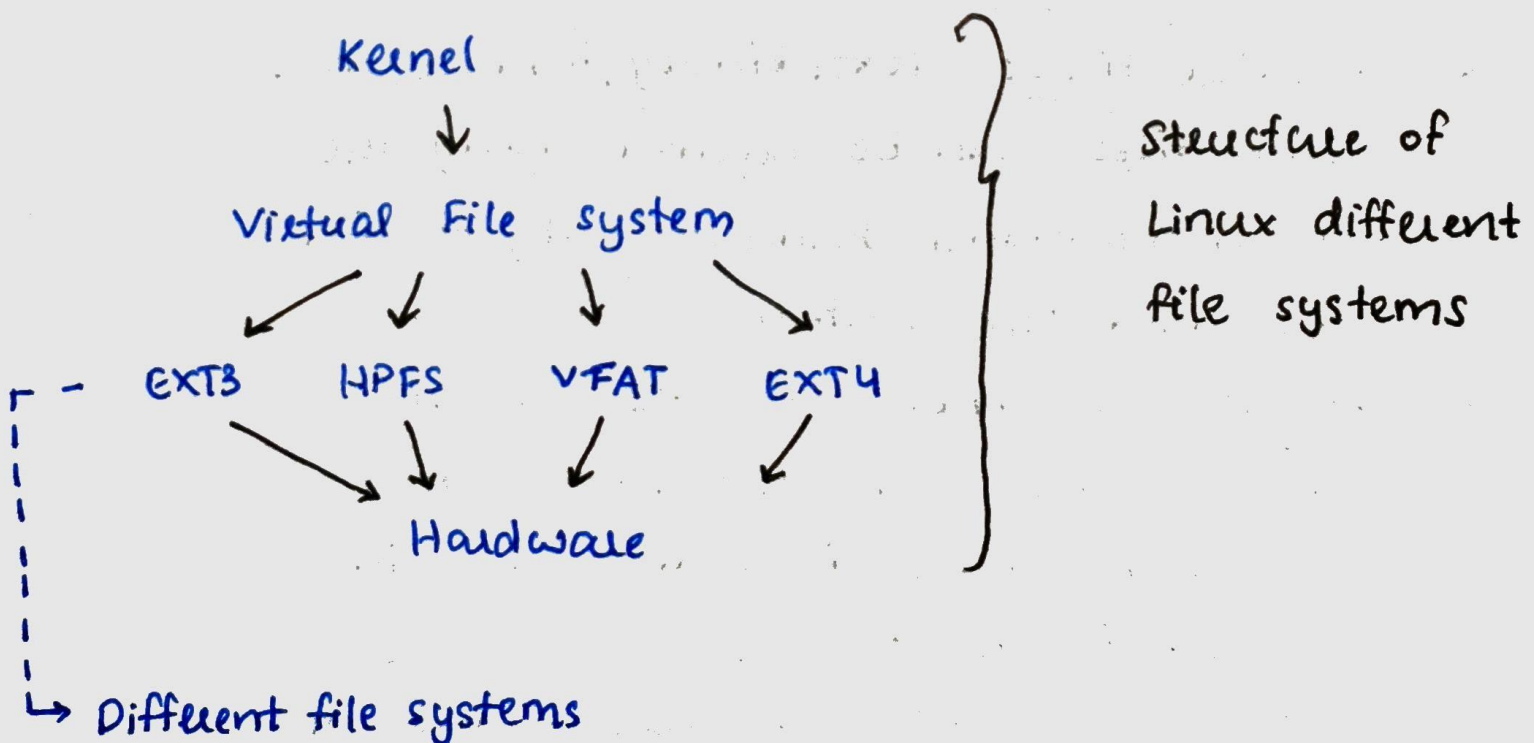
`ls` → function/command for listing all elements inside a given directory.

Different file system in Linux :

Structured collection of files on a disk drive or partition. A general purpose computer system needs to store data systematically to access files later.

Linux file system contains:

- Root Directory
- A specific data storage format (EXT3, EXT4, XFS, etc...)
- A partition for logical volume having a particular file system



V File - locking in Linux programming

File locking is a mutual exclusion mechanism to ensure a file can be read/written by multiple processes in a safe way. This mechanism restricts access to a file among multiple processes and it allows only one process to access the file in a specific time, thus avoiding the interceding update problem. This also prevents reading of the file while it is being modified or deleted.

Most operating systems support the concept of record locking, which means that individual records within any given file may be locked, thereby increasing the number of concurrent update processes. Database maintenance uses file locking, whereby it can serialize access to the entire physical file underlying a database.

Although this does prevent any other processes from accessing the file, it can be more efficient than to individually lock a large number of regions in the file by removing the overhead of acquiring and releasing each lock.

There are 2 types of file locking

i) **Advisory Locking:**

Not an enforced locking scheme, as it will work only if the participating processes are co-operating by explicitly acquired locks. Otherwise Advisory locks will be ignored if a process is not aware of locks at all.

ii) **Mandatory Locking:**

Mandatory Locking does not require any co-operation between the participating processes. Once a mandatory lock is activated on a file, the operating system prevents other processes from reading or writing the file.

- All locks support blocking and non-blocking operations
- Locks are allowed only on files and not directories
- Locks are automatically removed when the process exits or terminates.
- It's guaranteed that if a lock is acquired, the process acquiring the lock is still alive

These are few of the characteristics of File locking.