

APPLIED OPERATING SYSTEM LABORATORY



FEU ALABANG



FEU DILIMAN



FEU TECH

Technology Driven by Innovation

MODULE 4

LINUX FILE SYSTEM, FILENAMES AND WILDCARDS



FEU ALABANG



FEU DILIMAN



FEU TECH

Technology Driven by Innovation

OBJECTIVES

Upon completion of this module, the student will be able to:

- Understand LINUX file system/structure, partition, filename format, absolute and relative paths
- Create different file types and display directory content using wildcards in LINUX commands



FEU ALABANG



FEU DILIMAN



FEU TECH

Technology Driven by Innovation

TOPIC OUTLINE

- **Linux File System**
 - Directory Structure / Layout
 - Absolute and Relative Path
 - Filename Convention and File Types
- **How to create different files**
 - Empty file
 - Ordinary/Regular file
 - Hidden files
- **Others**
 - Autocomplete
 - Wildcards



FEU ALABANG



FEU DILIMAN



FEU TECH

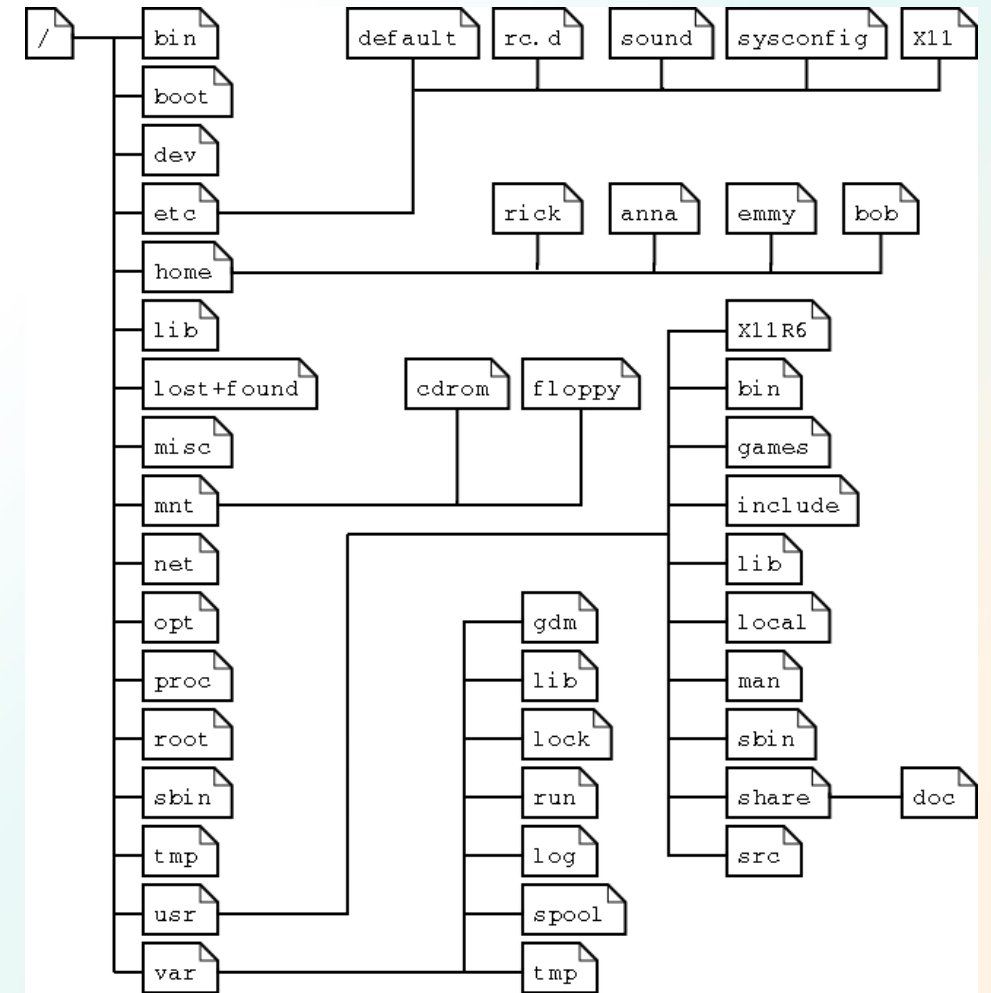
Technology Driven by Innovation

LINUX FILE SYSTEM

Linux File System Layout

The tree of the file system starts at the trunk or **forward slash (/)**. This directory, containing all underlying directories and files, is also called the **root directory** or "**the root**" of the file system.

```
vetcha@DESKTOP-U1V5H04:~$ cd /
vetcha@DESKTOP-U1V5H04:/$ ls
bin  dev  home  lib  lib64  media  opt  root  sbin  srv  tmp  var
boot  etc  init  lib32  libx32  mnt  proc  run  snap  sys  usr
```



LINUX FILE SYSTEM

Partition

This is a logical division of a hard disk created so that you can have different operating system on the same hard disk, or to create the impression of having separate hard drives for file management, multiple users or other purposes.

File System

This refers to the way in which files are named and where they are placed logically for storage and retrieval.



FEU ALABANG



FEU DILIMAN



FEU TECH

Technology Driven by Innovation

LINUX FILE SYSTEM

A **path** is a unique location to a file or a folder in a file system of an OS.

Absolute Path

The highest directory in the Linux directory tree is the /. An absolute path is defined as specifying the location of a file or directory from the root directory(/).

To write an absolute path-name:

- Start at the root directory (/) and work down
- Write a slash (/) after every directory name.

```
vetcha@DESKTOP-U1V5HO4: ~  
vetcha@DESKTOP-U1V5HO4:~$ pwd  
/home/vetcha
```



FEU ALABANG



FEU DILIMAN



FEU TECH

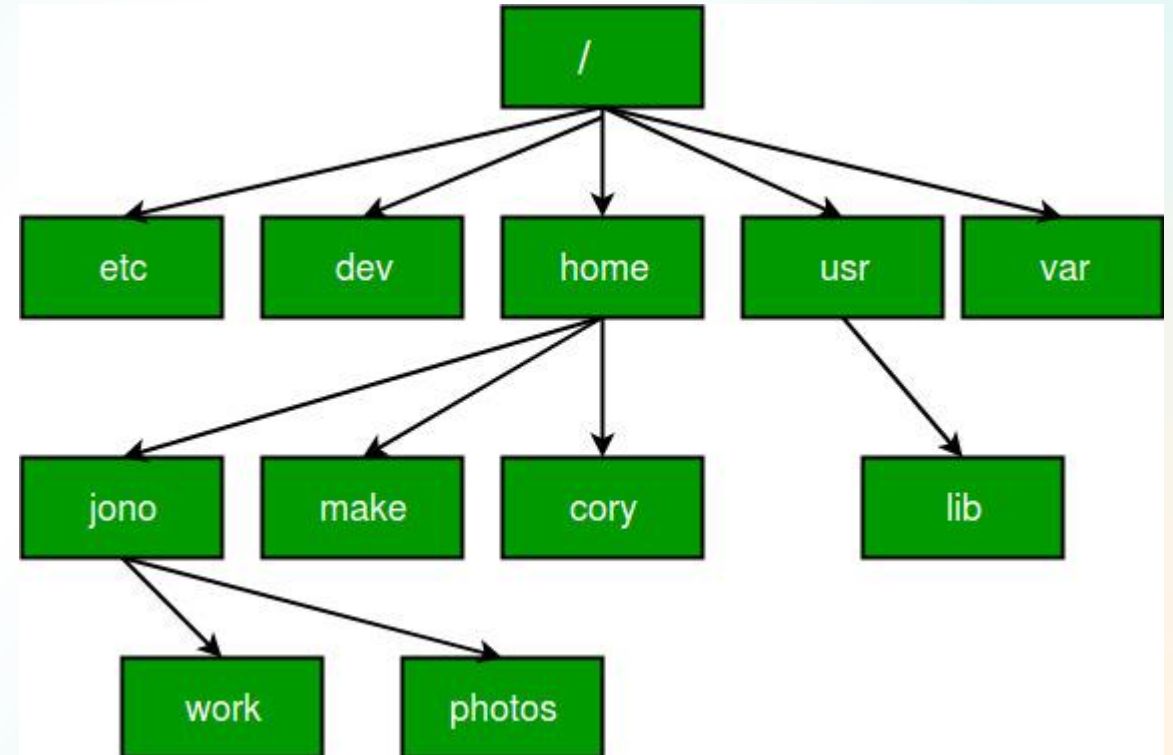
Technology Driven by Innovation

LINUX FILE SYSTEM

Relative Path

It is defined as the path related to the present working directory. It starts at your current directory and never starts with a / .

```
vetcha@DESKTOP-U1V5H04:~$ ls
A12      'Folder 1'  empty.txt  fileA
B        a+b        empty1     fileB
B12      a-b        empty2     fileC
C        a.txt      file       first
C13      b.txt      file1      hello.txt
FOLDER   c.txt      file2      hi.txt
FOLDER2  elif       file3      his.txt
vetcha@DESKTOP-U1V5H04:~$ cd FOLDER
vetcha@DESKTOP-U1V5H04:~/FOLDER$ cd /usr/lib
vetcha@DESKTOP-U1V5H04:/usr/lib$
```



LINUX FILE SYSTEM

Filename Convention

- All file names are **case sensitive**. So filenames **sam.txt Sam.txt SAM.txt** all are three different files.
- You can use upper and lowercase letters, numbers, “.” (dot), and “_” (underscore) symbols.
- You can use other special characters such as blank space except / (root directory), but they are hard to use and it is better to avoid them.
- Most modern Linux and UNIX limit filename to 255 characters (255 bytes). However, some older version of UNIX system limits filenames to 14 characters only.

```
vetcha@DESKTOP-U1V5HO4: ~  
vetcha@DESKTOP-U1V5HO4:~$ ls  
A12  C13  a-b  b.txt  file  file2  fileA  fileC  
B12  a+b  a.txt  c.txt  file1  file3  fileB  ls
```

LINUX FILE SYSTEM

File Types

Seven different types of Linux file types and **ls** command identifiers:

- **-** : regular file
- **d** : directory
- **c** : character device file
- **b** : block device file
- **s** : local socket file
- **p** : named pipe
- **l** : symbolic link

vetcha@DESKTOP-U1V5HO4: ~/FOLDER

```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ mkfifo pipe
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls
Folder1 hello.txt pipe regfile1 regfile2
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l
total 0
drwxrwxrwx 1 vetcha vetcha 512 Jul 24 22:21 Folder1
-rw-rw-rw- 1 vetcha vetcha  6 Jul 24 21:30 hello.txt
prw-rw-rw- 1 vetcha vetcha  0 Jul 24 22:21 pipe
-rw-rw-rw- 1 vetcha vetcha 15 Jul 24 20:58 regfile1
-rw-rw-rw- 1 vetcha vetcha  0 Jul 24 21:12 regfile2
```

vetcha@DESKTOP-U1V5HO4: /

```
vetcha@DESKTOP-U1V5HO4:~$ cd /
vetcha@DESKTOP-U1V5HO4:/$ ls -l
total 580
lrwxrwxrwx 1 root root 7 Apr 23 14:40 bin -> usr/bin
drwxr-xr-x 1 root root 512 Apr 23 14:49 boot
drwxr-xr-x 1 root root 512 Jul 23 19:29 dev
drwxr-xr-x 1 root root 512 Jul 23 19:29 etc
drwxr-xr-x 1 root root 512 Jul 23 19:29 home
-rwxr-xr-x 1 root root 591344 Jan 1 1970 init
lrwxrwxrwx 1 root root 7 Apr 23 14:40 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Apr 23 14:40 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 Apr 23 14:40 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 Apr 23 14:40 libx32 -> usr/libx32
drwxr-xr-x 1 root root 512 Apr 23 14:40 media
```



FEU ALABANG



FEU DILIMAN



FEU TECH

Technology Driven by Innovation

FILE TYPE

file command

- It is used to determine the type of a file.

Syntax: **file** <filename>

```
vetcha@DESKTOP-U1V5HO4: ~
```

```
vetcha@DESKTOP-U1V5HO4:~$ file numbers
numbers: ASCII text
vetcha@DESKTOP-U1V5HO4:~$
```

```
vetcha@DESKTOP-U1V5HO4: ~
```

```
vetcha@DESKTOP-U1V5HO4:~$ file *
A12:      empty
B12:      empty
C13:      empty
FOLDER:   directory
a+b:      empty
a-b:      empty
a.txt:    empty
b.txt:    empty
c.txt:    empty
empty1:   empty
empty2:   empty
file:     empty
file1:    empty
file2:    empty
file3:    empty
fileA:    empty
fileB:    empty
fileC:    empty
hello.txt: ASCII text
hi.txt:   ASCII text
his.txt:  ASCII text
ls:       empty
my folder: directory
names:    ASCII text
numbers:  ASCII text
regfile1: ASCII text
regfile3: empty
sample.txt: ASCII text
vetcha@DESKTOP-U1V5HO4:~$
```



FEU ALABANG



FEU DILIMAN



FEU TECH

Technology Driven by Innovation

HOW TO CREATE DIFFERENT FILES

Regular file

- **cat**, **touch**, and **echo** commands can be used to create regular or ordinary files.
- It governs all different files such as text files, images, binary files, shared libraries, etc.

```
vetcha@DESKTOP-U1V5HO4: ~/FOLDER
```

```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ cat > regfile1
Regular File 1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls
regfile1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ cat regfile1
Regular File 1
```

```
vetcha@DESKTOP-U1V5HO4: ~/FOLDER
```

```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls
regfile1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ touch regfile2
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls
regfile1  regfile2
```

```
vetcha@DESKTOP-U1V5HO4: ~/FOLDER
```

```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ echo "hello" > hello.txt
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls
hello.txt  regfile1  regfile2
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ cat hello.txt
hello
vetcha@DESKTOP-U1V5HO4:~/FOLDER$
```



FEU ALABANG



FEU DILIMAN



FEU TECH

Technology Driven by Innovation

HOW TO CREATE DIFFERENT FILES

Empty file

- The **touch** command is the easiest way to create new, empty files.

Hidden file

- You can recognize the **hidden** files and folders with the **dot (.)** before their names.
- Use **ls -a** to display all the files, including the hidden ones

```
vetcha@DESKTOP-U1V5H04: ~  
vetcha@DESKTOP-U1V5H04:~$ ls  
A12  C13  a-b   b.txt  file   file2  fileA  fileC  'my folder'  
B12  a+b  a.txt  c.txt  file1  file3  fileB  ls  
vetcha@DESKTOP-U1V5H04:~$ touch empty1 empty2  
vetcha@DESKTOP-U1V5H04:~$ ls  
A12  C13  a-b   b.txt  empty1  file   file2  fileA  fileC  'my folder'  
B12  a+b  a.txt  c.txt  empty2  file1  file3  fileB  ls  
vetcha@DESKTOP-U1V5H04:~$ cat empty1  
vetcha@DESKTOP-U1V5H04:~$ cat empty2  
vetcha@DESKTOP-U1V5H04:~$
```

```
vetcha@DESKTOP-U1V5H04:~/FOLDER$ touch .h1  
vetcha@DESKTOP-U1V5H04:~/FOLDER$ ls -a  
.  .. .h1 hello.txt regfile1 regfile2  
vetcha@DESKTOP-U1V5H04:~/FOLDER$ ls  
hello.txt regfile1 regfile2
```



OTHERS

Autocomplete

Autocomplete works on files and directories. Just press **Tab** key to complete the directory or file that you are looking for.

Wildcards

A wildcard is a symbol or a set of symbols that stands in for other characters. It can be used to substitute for any other character or characters in a string.

* - represents zero or more characters

? - represents a single character

[] - represents a range of characters

```
vetcha@DESKTOP-U1V5HO4: ~  
vetcha@DESKTOP-U1V5HO4:~$ ls  
a.txt b.txt c.txt file file1 file2 file3  
vetcha@DESKTOP-U1V5HO4:~$ ls file?  
file1 file2 file3  
vetcha@DESKTOP-U1V5HO4:~$ ls file*  
file file1 file2 file3
```

```
vetcha@DESKTOP-U1V5HO4: ~  
vetcha@DESKTOP-U1V5HO4:~$ ls ?.*  
a.txt b.txt c.txt
```

```
vetcha@DESKTOP-U1V5HO4: ~  
vetcha@DESKTOP-U1V5HO4:~$ ls file[1-3]  
file1 file2 file3
```

```
vetcha@DESKTOP-U1V5HO4: ~  
vetcha@DESKTOP-U1V5HO4:~$ ls file[1-3A]  
file1 file2 file3 fileA  
vetcha@DESKTOP-U1V5HO4:~$ ls file[1-3A-C]  
file1 file2 file3 fileA fileB fileC
```



FEU ALABANG



FEU DILIMAN



FEU TECH

Technology Driven by Innovation

REFERENCES

- Sobell, M., et al. (2017). A Practical Guide to Linux Commands, Editors, and Shell Programming, 4th Ed. Addison-Wesley Professional
- Cobbaut, P. (2016). Mastering Linux- Networking
- Blum, R., (2015). Linux Command Line and Shell Scripting Bible



FEU ALABANG



FEU DILIMAN



FEU TECH

Technology Driven by Innovation