**Linux**

**PHYSICAL LINK**

Physical links can only be created for files and not for directories, the PHYSICAL LINK is the name given to a file in another location which contains the path or information that is put to it.

**The symbolic** links have the same function as a physical but the difference is that it does not contain the data of the file, instead it points to that folder or to the registry where the data is located. In other words it is like a shortcut in Windows that when I click it takes me to where the content I need, it would be like a kind of hyperlink of a web page. If the destination file is deleted the symbolic link will be unusable, because it will not be able to refer to the file it was pointing to, since it is not there.

**Commands:**

To see system reference manual -> **man**

Logical Link: **pwd -L**

Physical Link: **pwd -P**

print name of current/working directory : **pwd**

version information : **- -version**

To get Help: **–help**

list directory contents: **man -ls**

Change Directory: **cd**

Current Directory: **cd.**

Get Back to **root** Directory: **cd**

Get back to **Previous** Directory: **cd - or cd ..**

To go in Home: **cd /home**

To go in Home’s Documents: **cd /home/appscodepc/Documents/**

**Or cd Documents**

How to make a file Hidden: **.File\_name**

How to see all directory List (Hidden also): **ls -a**

Make a Fresh Terminal: **clear**

**The "ls -l"**

displays the **contents of the current directory** in a **long listing format,** one per line

To see in more human readable format: **ls -lh**

**drwxrwxr-x 2 appscodepc appscodepc 4.0K জানু 1 12:28**

**Directory, owner, group, others**

Permission no. of link root root Size Date Time

**owner group**

**permissions(r-read, x-execute, w-write)**

**root root stands for the owner of the file and the group owner**

**Next is the file size,**

**after which the date the file was created,**

**the time when it was last modified and the files name.**

**man mkdir -** make directories

A tree of Directories: **mkdir -p Linux/Ubuntu/Evan**

**man rmdir -** remove empty directories

Remove an empty Directory: **rmdir File\_name**

Remove Full Directory (not empty): **rmdir -p Linux/Ubuntu/Evan**

**man file** — determine the file type

See the type of the file: **file file\_name**

**man touch** - change file timestamps

Update the access and modification times of each FILE to the current time.

If file does not exist, create a empty file,

If a file exists, update the modified Time.

Create single file name f1.txt: **touch f1.txt**

Create multiple file name f2.txt & f3.txt: **touch f2.txt f3.txt**

**man rm -** remove files or directories

Remove Single File: **rm File\_Name**

**-i** prompt before every removal

Remove multiple file: **rm -i f2.txt f3.txt -> yes ->yes**

**-r, -R, --recursive -> remove directories and their contents recursively**

Remove the Test Directories: **rm -r Test/**

**man cp -** copy files and directories

Copy in Same Directory: **cp File\_name Copied\_File\_name**

**cp f1.txt f1\_cp.txt**

Copy in Specific Directory: **cp f1.txt /home/appscodepc/Documents/**

See the list of files in a Directory: **ls /home/appscodepc/Documents/**

Multiple Copy: **cp -r test testcp**

In the test there are 2 files. Both files are copied in testcp.

**man mv** - move (rename) files

Rename f1 with f2: **mv f1.txt f2.txt**

Move f2 in another directory: **mv f2.txt test/f3.txt**

**man head -** output the first part of files

**Print the first 10 lines**  of each FILE to standard output.

With more than one FILE, precede each with a header giving the file name.

Print First 10 lines: **head file\_name**

Print First 5 lines: **head -5 file\_name**

**man tail -** output the last part of files

Last 10 lines

**man cat** - concatenate files and print on the standard output

Print the full file information

Just see one full file: **cat f3.txt**

Concat 2 files: **cat f1.txt f3.txt**

**echo -** display a line of text

“The” will be displayed in text1 file: **echo The >text1.txt**

**echo big >text2.txt**

**echo apple >text3.txt**

**cat text1.txt text2.txt text3.txt -> The big apple**

Merge all 3 files in a single file: **cat text1.txt text2.txt text3.txt >all.txt**

**cat all.txt -> The big apple [ Same Output ]**

**This will ask input cat > file1.txt**

**My name is Evan**

**and ..**

**cat file1.txt -> will show the file.**

Go to as **root user: sudo su**

file1 will be copied in file2: **cat file1.txt > file2.txt**

**man more -** file perusal filter for crt viewing

more is a filter for paging through text one screenful at a time. This version is especially primitive. Users should realize that less(1) provides more(1) emulation plus extensive enhancements.

**more file\_name**

**less** - opposite of more

Less is a program similar to more(1), but it **has many more features**. Less does not have to read the entire input file before starting, so with large input files it starts up faster than text editors like vi(1). Less uses termcap (or terminfo on some systems), so it can run on a variety of terminals. There is even limited support for hardcopy terminals. (On a hardcopy terminal, lines which should be printed at the top of the screen are prefixed with a caret.)

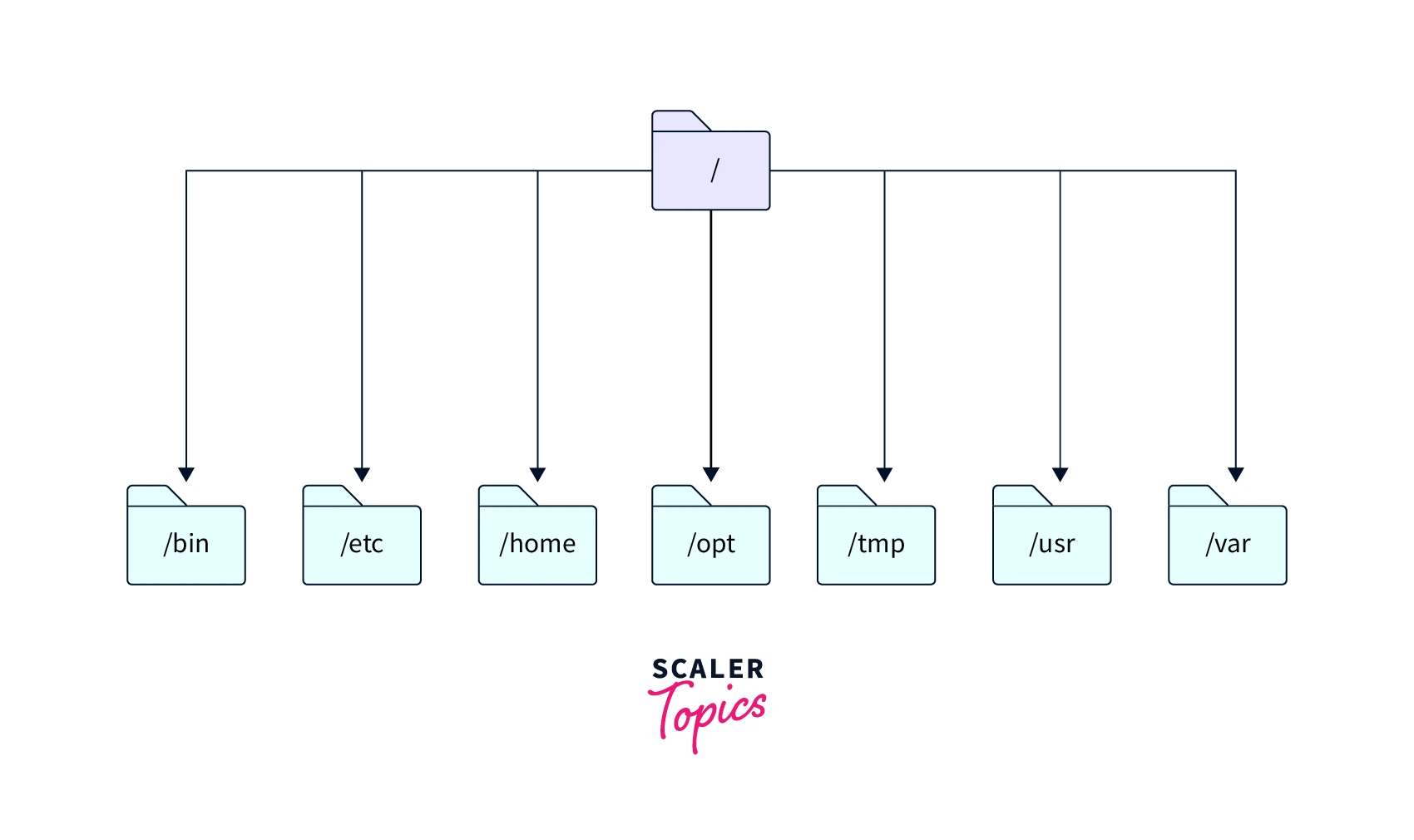
**less file\_name**

**Linux File System**

**Directories, also known as folders, are an integral part of the Linux file system.**

**A directory can be thought of as a virtual container that holds files and other directories within it.**

**One of the primary functions of directories in Linux is to provide a hierarchical structure for organizing files. This structure starts at the root directory (/), and branches out into subdirectories as needed.**

****

* **/ (root directory):  
  The root directory is the top-level directory in the Linux file system. All other directories and files are contained within the root directory.**
* **/bin:  
  The /bin stands for binaries. This directory contains essential command-line tools and programs that are required for basic system administration tasks.**
* **/boot:  
  The /boot directory contains the boot loader files and kernel images needed to start the system.**
* **/dev:  
  The /dev directory contains device files that represent hardware devices and virtual devices such as terminals, printers, and disks.**
* **/etc:  
  The /etc directory contains system configuration files that are used by various applications and services on the system.**
* **/home:  
  The /home directory contains the home directories of users on the system. Each user has their own subdirectory within /home where they can store their personal files and settings.**
* **/lib:  
  The /lib directory contains shared library files that are needed by various programs on the system.**
* **/media:  
  The /media directory is used to mount removable media such as CDs, DVDs, and USB drives.**
* **/mnt:  
  The /mnt directory is used to mount file systems temporarily, such as network file systems or disk images.**
* **/opt:  
  The /opt directory is used to store additional software packages that are not part of the core system.**
* **/proc:  
  The /proc directory is a virtual file system that provides information about running processes and system resources.**
* **/run:  
  The /run directory contains temporary files that are created by system services and daemons.**
* **/sbin:  
  The /sbin directory contains system binaries and administrative tools that are required for system maintenance.**
* **/srv:  
  The /srv directory is used to store data for services provided by the system.**
* **/sys:  
  The /sys directory is a virtual file system that provides information about the system's hardware and devices.**
* **/tmp:  
  The /tmp directory contains temporary files that are created by applications and services running on the system.**
* **/usr:  
  The /usr directory contains user-level programs, libraries, documentation, and shared data files.**
* **/var:  
  The /var directory contains variable data files that change frequently, such as log files and system databases.**

**Linux Package Manager:**

**Sudo -> super user do**

**sudo, sudoedit —** execute a command as another user

sudo allows a permitted user to execute a command as the superuser or another user, as specified by the security policy.

**sudo apt update**

**sudo apt upgrade**

**sudo apt install Software\_name**

**Text Editor:**

nano is a small and friendly editor. It copies the look and feel of Pico, but is free software, and implements several features that Pico lacks, such as: opening multiple files, scrolling per line, undo/redo, syntax coloring, line numbering, and soft-wrapping over‐ long lines.

**What is the dpkg command?**

Essentially, the man page describes it like this: “dpkg is a tool to install, build, remove and manage Debian packages.” We use the dpkg command to interact with packages on our system.

Search a software: **sudo apt search software\_Name**

**Nano:**

**Exit-> ctrl+X ctrl+C Enter.**

**Vim:**

**To insert -> i**

**To save changes-> :wq**

**Not save-> :q!**