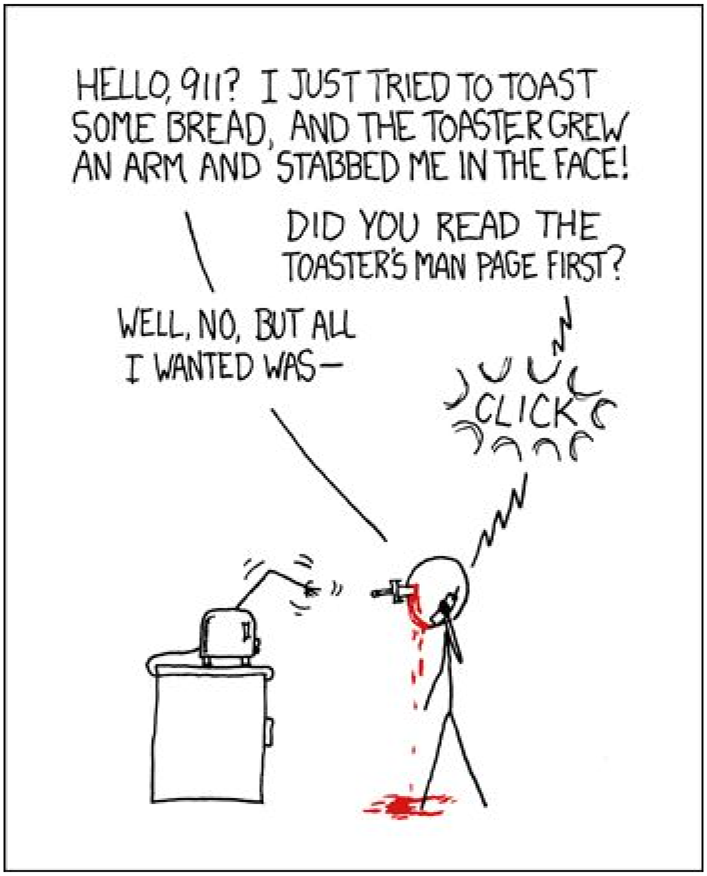
GNU/Linux Introduction - What’s behind the GUI?


HELLO, 911? I JUST TRIED TO TOAST SOME BREAD, AND THE TOASTER GREW AN ARM AND STABBED ME IN THE FACE!
DID YOU READ THE TOASTER'S MAN PAGE FIRST?
WELL, NO, BUT ALL I WANTED WAS—
CLICK

https://xkcd.com/293/

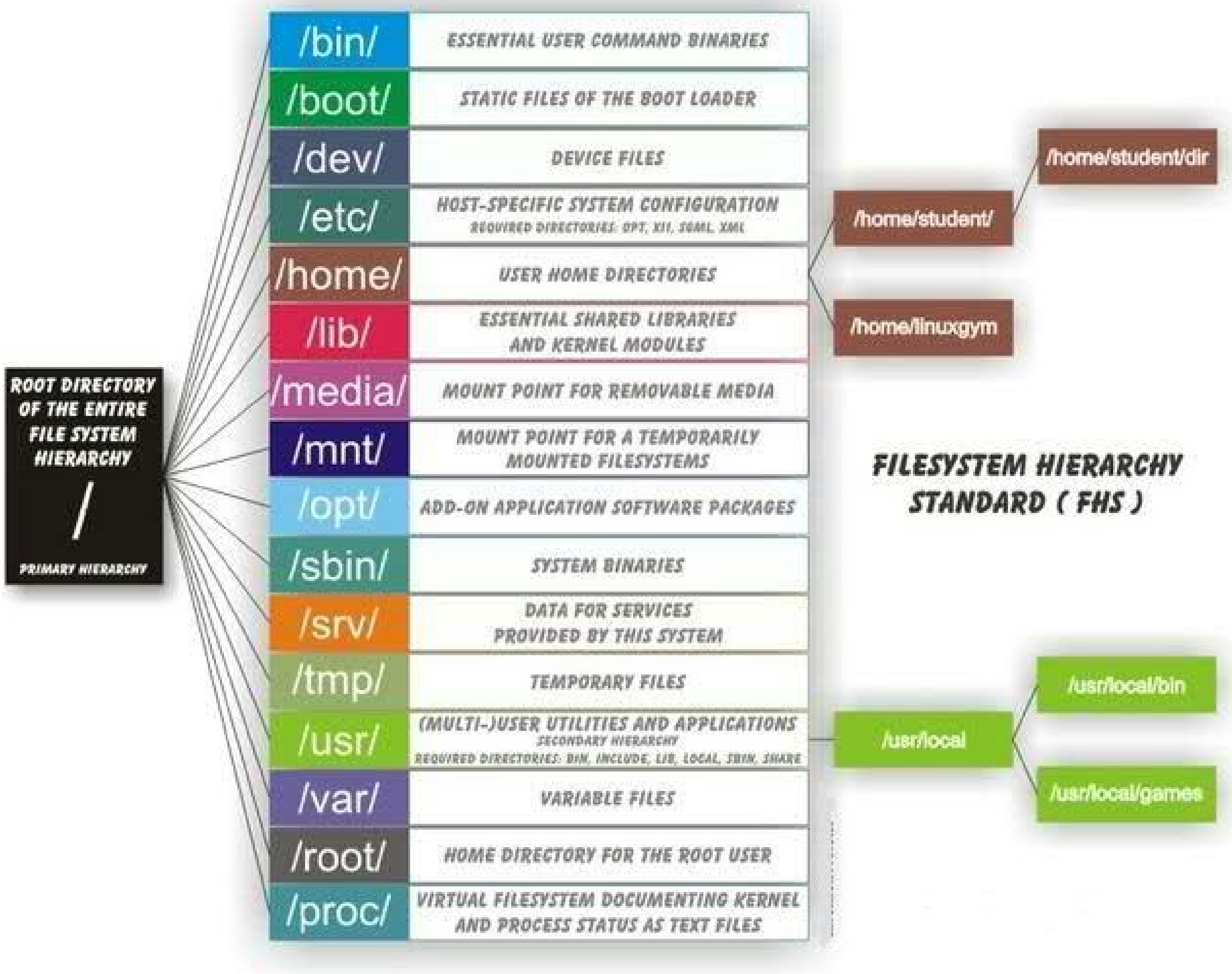
GNU/Linux - Linux is the kernel: the program in the system that allocates the machine's resources to the other programs that you run. Linux is normally used in combination with the GNU operating system: the whole system is basically GNU with Linux added, or GNU/Linux. All the so-called “Linux” distributions are really distributions of GNU/Linux. **http://www.gnu.org/gnu/linux-and-gnu.en.html**

GUI - In computing, a graphical user interface or **GUI**, sometimes pronounced /ˈɡuːi/ ("gooey") is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation. (https://en.wikipedia.org/wiki/Graphical_user_interface)

A **Unix shell** is a command-line interpreter or shell that provides a traditional user interface for the Unix operating system and for Unix-like systems. Users direct the

operation of the computer by entering commands as text for a command line interpreter to execute, or by creating text scripts of one or more such commands. Users typically interact with a Unix shell using a terminal emulator.
https://en.wikipedia.org/wiki/Unix_shell

Bash is a Unix shell and command language written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell.[3][4] Released in 1989,[5] it has been distributed widely as the shell for the GNU operating system and as a default shell on Linux and OS X. It has been ported to Microsoft Windows and distributed with Cygwin and MinGW, to DOS by the DJGPP project, to Novell NetWare and to Android via various terminal emulation applications. In the late 1990s, Bash was a minor player among multiple commonly used shells; at present Bash has overwhelming favor.
https://en.wikipedia.org/wiki/Bash_(Unix_shell)



/bin/	ESSENTIAL USER COMMAND BINARIES
/boot/	STATIC FILES OF THE BOOT LOADER
/dev/	DEVICE FILES
/etc/	HOST-SPECIFIC SYSTEM CONFIGURATION REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/	USER HOME DIRECTORIES
/lib/	ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/	MOUNT POINT FOR REMOVABLE MEDIA
/mnt/	MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/	ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/	SYSTEM BINARIES
/srv/	DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/	TEMPORARY FILES
/usr/	(MULTI-)USER UTILITIES AND APPLICATIONS SECONDARY HIERARCHY REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/	VARIABLE FILES
/root/	HOME DIRECTORY FOR THE ROOT USER
/proc/	VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES

Common Commands

It should be noted that most of the following commands have a number of options available in addition to what is listed here, for more information about them use...

man - A man page (short for manual page) is a form of online software documentation usually found on a Unix or Unix-like operating system. Topics covered include computer programs (including library and system calls), formal standards and conventions, and even abstract concepts.

man *name*

cd - The cd command is used to change the current directory (i.e., the directory in which the user is currently working) in Linux and other Unix-like operating systems.

cd [*OPTION*] [*DIRECTORY*]
eg. cd ..

ls - List information about the FILEs (the current directory by default).

ls [*OPTION*]... [*FILE*]...
eg. ls -al

pwd - The pwd is an acronym for print working directory. The pwd command is considered as one of the most frequently used commands on Linux, AIX, HP-UX, *BSD, and other UNIX like operating systems along with the ls, and cd commands

passwd - The passwd command changes passwords for user accounts. A normal user may only change the password for his/her own account, while the superuser may change the password for any account.

passwd [*options*] [*LOGIN*]
or just:
passwd

cp - a UNIX command for copying files and directories. The command has three principal modes of operation, expressed by the types of arguments presented to the program for copying a file to another file, one or more files to a directory, or for copying entire directories to another directory.

cp *sourcefile targetfile*

mv - move (or rename) files Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

mv *SOURCE DEST*
mv *SOURCE*... *DIRECTORY*

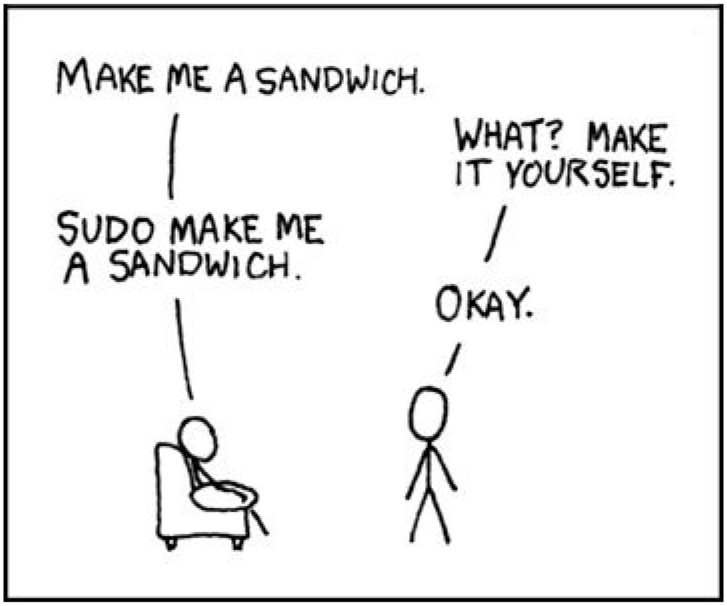
rm - remove files or directories

rm *file*

mkdir/rmdir - make directory / remove directory

mkdir [*OPTION*] *DIRECTORY*...

sudo - execute a command as another user (root by default)
allows a permitted user to execute a command as the superuser or another user, as specified in the sudoers file. The real and effective uid and gid are set to match those of the target user as specified in the passwd file (the group vector is also initialized when the target user is not root). By default, sudo requires that users authenticate themselves with a password (NOTE: by default this is the user's password, not the root password). Once a user has been authenticated, a timestamp is updated and the user may then use sudo without a password for a short period of time (5 minutes unless overridden in sudoers).



chmod - change file access permissions **chmod** changes the permissions of each given file according to *mode*, which can be either a symbolic representation of changes to make, or an octal number representing the bit pattern for the new permissions.
For example if you want to change the permissions of the file so that everybody has full access to it, you would enter:

chmod 777 *file*

The first 7 sets the permissions for the user, the second 7 sets the permissions for the group, and the third 7 sets the permissions for everybody else, who has access to this computer.

more about chmod: http://ubuntuforums.org/showthread.php?t=1252905

chown - Change the owner and/or group of each FILE to OWNER and/or GROUP.
chown [*OPTION*]... *OWNER*[*:*[*GROUP*]] *FILE*...

apt-get - The apt-get command is a powerful command-line tool used to work with Ubuntu's *Advanced Packaging Tool*(APT) performing such functions as installation of new software packages, upgrade of existing software packages, updating of the package list index, and even upgrading the entire Ubuntu system.
more info: http://linux.about.com/od/ubusrv_doc/a/ubusg11t01.htm

tail - Print the last 10 lines of each FILE to standard output. With more than one FILE, precede each with a header giving the file name. With no FILE, or when FILE is -, read standard input.

tail *file*

echo - display a line of text by printing STRING(s) to standard output.

echo [*OPTION*]... [*STRING*]...

cat - Concatenate FILE(s), or standard input, to standard output.

cat *file*

grep - searches the named input *FILE*s (or standard input if no files are named, or the file name **-** is given) for lines containing a match to the given *PATTERN*. By default, **grep** prints the matching lines.

```
grep [options] PATTERN [FILE...]
```

more - More is a filter for paging through text one screenful at a time. This version is especially primitive. Users should realize that less provides more emulation and extensive enhancements.

```
more file
```

less - a program similar to *more*, but which allows backward movement in the file as well as forward movement. Also, *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* .

```
less file
```

| - command pipe Pipes allow you to funnel the output from one command into another where it will be used as the input. In other words, the standard output from one program becomes the standard input for another.

```
ls -la /dev | more
```

*** asterisk, glob, wildcard -** Wildcard Matching

> A string is a wildcard pattern if it contains one of the characters '?', '*' or '['. Globbing is the operation that expands a wildcard pattern into the list of pathnames matching the pattern. Matching is defined by:
>
> A '?' (not between brackets) matches any single character.
>
> A '*' (not between brackets) matches any string, including the empty string.

> - send output to a file instead of the terminal (overwrites if file exists)

```
ls -al > results_of_ls.txt
```

>> - same as >, but appends to the end of the file

diff - analyzes two files and prints the lines that are different. Essentially, it outputs a set of instructions for *how to change one file in order to make it identical to the second file.*

diff *file1.txt file2.txt*

file - tests each argument in an attempt to classify it. There are three sets of tests, performed in this order: filesystem tests, magic number tests, and language tests. The *first* test that succeeds causes the file type to be printed.

file *FILE*

find - search for files in a directory hierarchy

find *[path...] [expression]*

locate - When used without any options, locate displays every *absolute pathname* for which the user has access permission that contains any of the names of files and/or directories that are provided to it as arguments

locate *[options] name(s)*

clear - clear the terminal screen

history - Manipulate the history list

history

history -cw

Directories and paths

../ = parent directory
./ = current directory

absolute and relative path locations
What is a path?

A path is a unique location to a file or a folder in a file system of an OS. A path to a file is a combination of / and alpha-numeric characters.

What is an absolute path?

An absolute path is defined as the specifying the location of a file or directory from the root directory(/). In other words we can say absolute path is a complete path from start of actual filesystem from / directory.

Some examples of absolute path:

/var/ftp/pub
/etc/samba.smb.conf
/boot/grub/grub.conf

If you see all these paths started from / directory which is a root directory for every Linux/Unix machines.

What is the relative path?

Relative path is defined as path related to the present working directory(pwd). Suppose I am located in /var/log and I want to change directory to /var/log/kernel. I can use relative path concept to change directory to kernel

changing directory to /var/log/kernel by using relative path concept.

```
pwd
/var/log
cd kernel
```

Note: If you observe there is no / before kernel which indicates it's a relative directory to present working directory.

Editors - vim, emacs, gedit, sublime

COMING SOON

More Info

GLUG - GNU/Linux & Unix Group http://www.gnulug.org/

Man pages - http://linux.die.net/man/

Unix & Linux Stack Exchange - http://unix.stackexchange.com/

The Linux Command Line: A Complete Introduction
http://smile.amazon.com/Linux-Command-Line-Complete-Introduction/dp/1593273894/

How Linux Works: What Every Superuser Should Know
http://smile.amazon.com/How-Linux-Works-Superuser-Should/dp/1593275676