

# Linux BootCamp

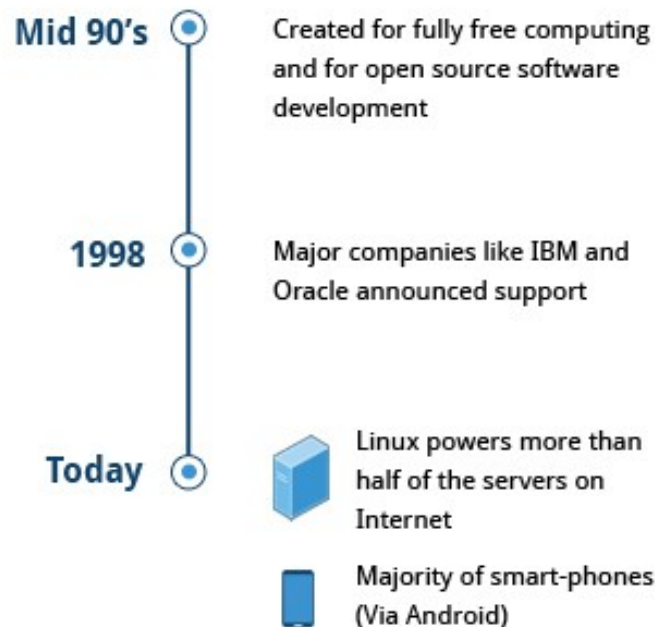
Muthuraj Thangavel

# About Myself

- Dev and sysadmin for [cs.annauniv.edu](http://cs.annauniv.edu)
- Backup and Disaster data recovery volunteer –  
Centre for research, Anna University
- Linux User
- Systems Engineer Recruit, Directi ([media.net](http://media.net))

# Introduction & History

- Linus Torvalds was a student in Helsinki, Finland
- In 1991, he started a project: writing his own operating system kernel. This soon became known as the Linux kernel.
- In 1992, Linux was re-licensed using the General Public License (GPL) by GNU

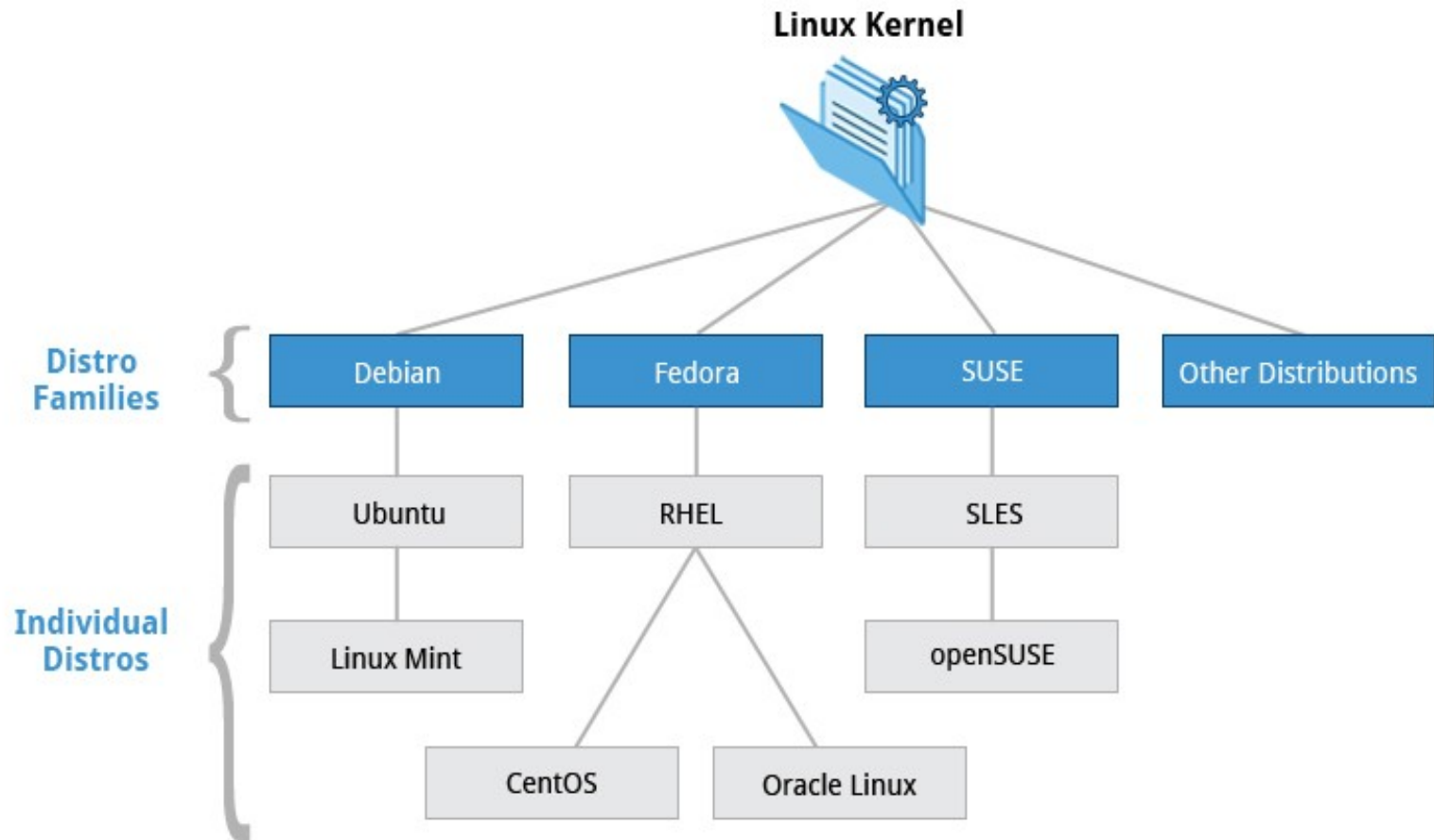
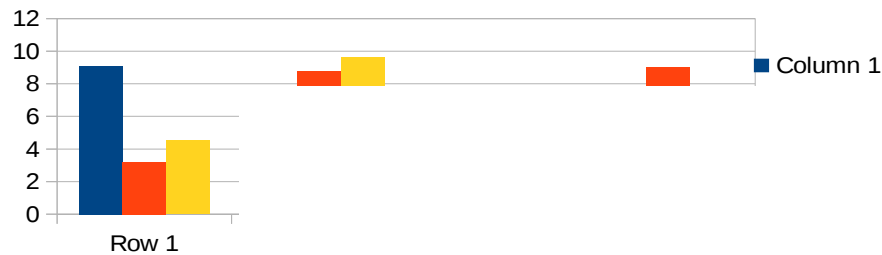




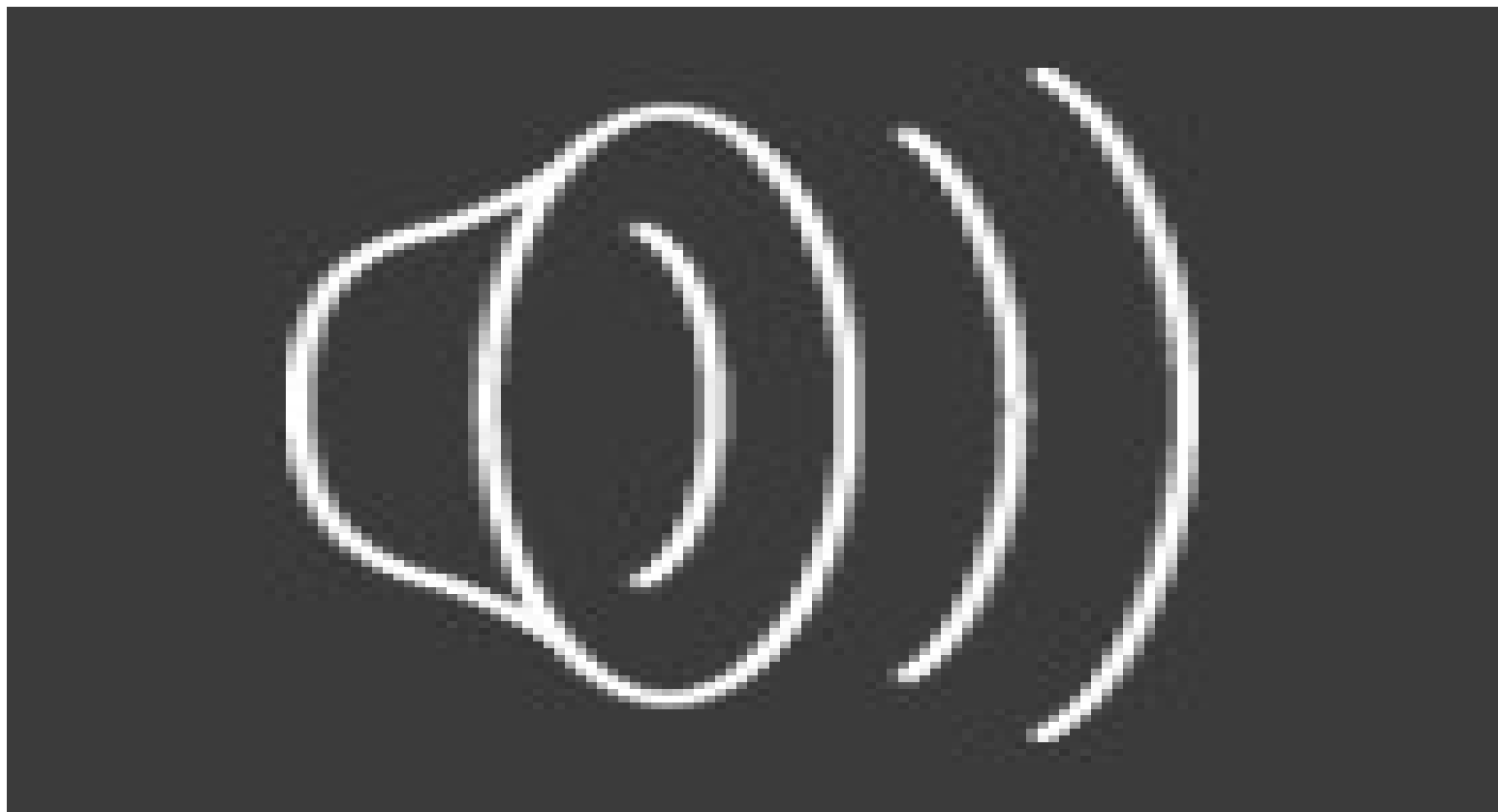
Microsoft ❤️ Linux



# Linux Distribution families



# Linux Terminology

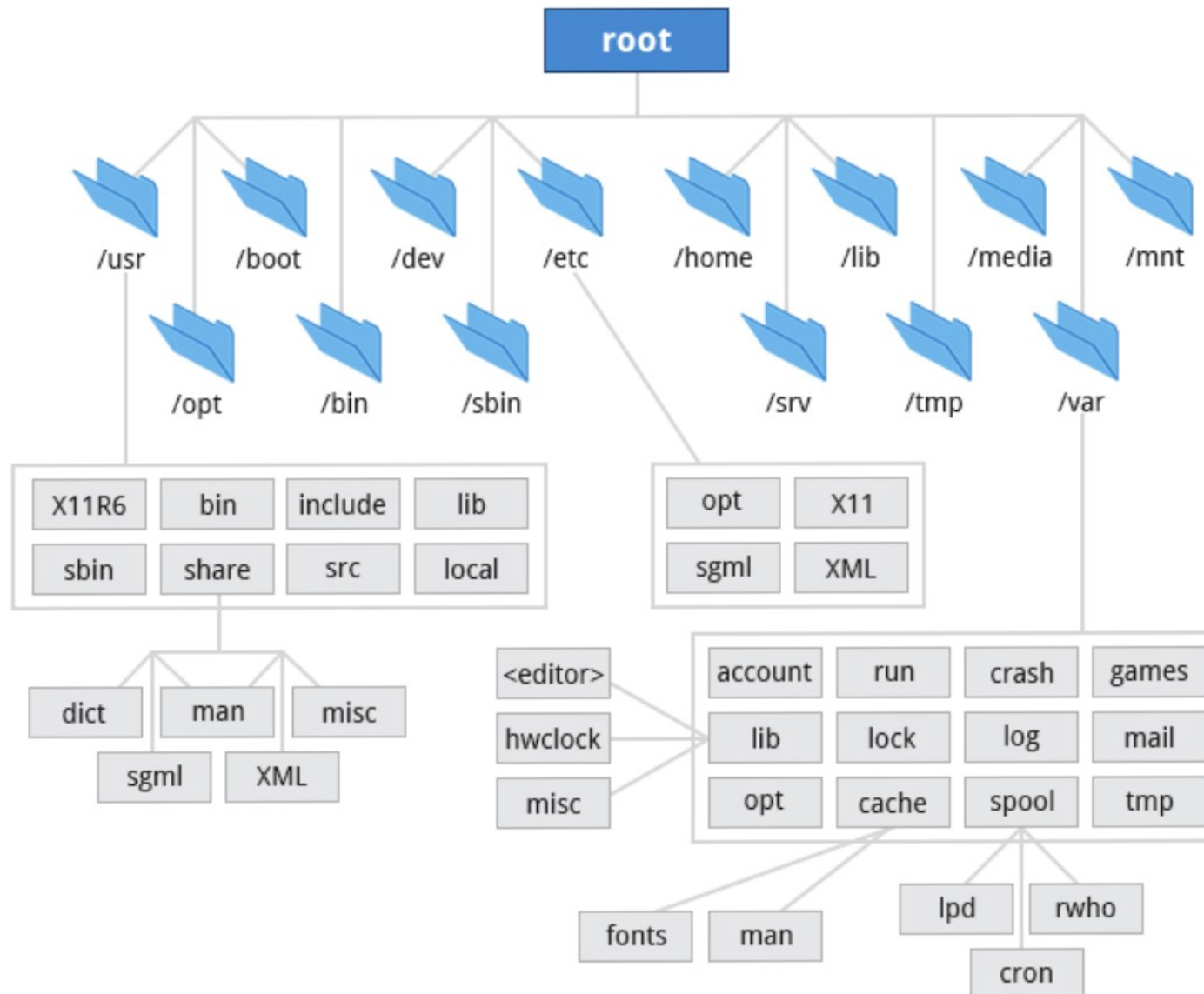




# Linux Filesystem

- Linux systems store their important files according to a standard layout called the Filesystem Hierarchy Standard, or FHS.
- Conventional disk filesystems: **ext2, ext3, ext4, XFS, Btrfs, JFS, NTFS, etc.**
- Flash storage filesystems: **ubifs, JFFS2, YAFFS, etc.**
- Special purpose filesystems: **procfs, sysfs, tmpfs, debugfs, etc.**
- Linux uses the ‘/’ character to separate paths (unlike Windows, which uses ‘\’), and does not have drive letters.
- New drives are mounted as directories in the single filesystem, often under /media
- All Linux filesystem names are case-sensitive

# Filesystem hierarchy

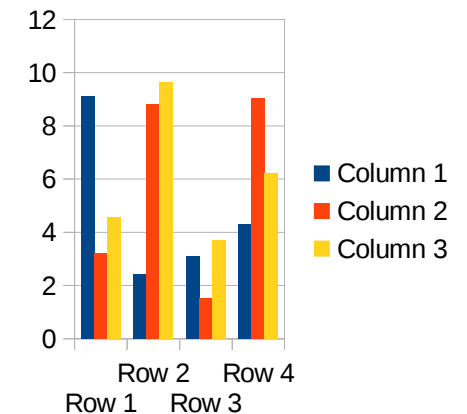
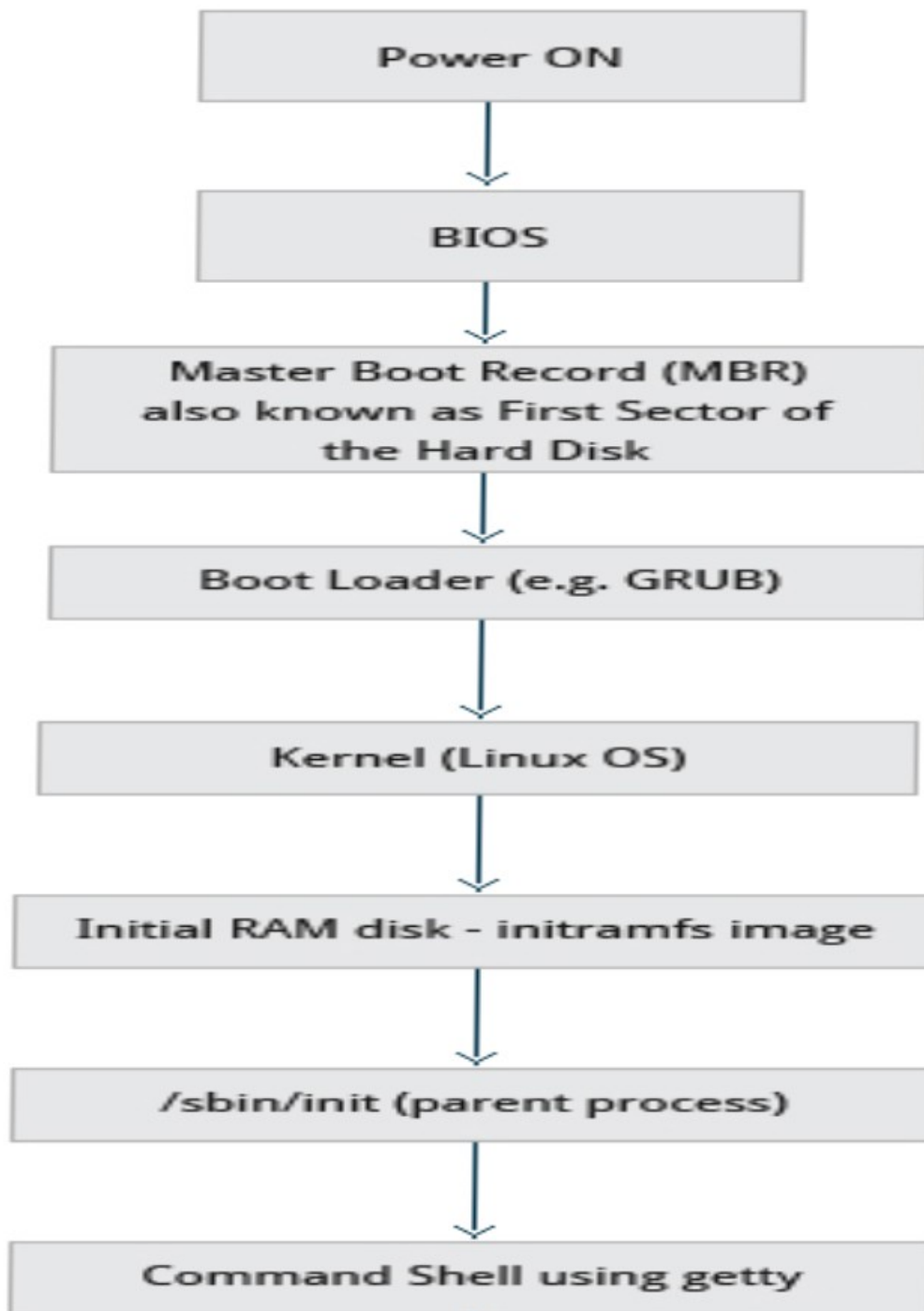


# Comparison of windows and linux filesystem

	Windows	Linux
Partition	Disk1	/dev/sda1
Filesystem type	NTFS/FAT32	EXT3/EXT4/XFS...
Mounting Parameters	DriveLetter	MountPoint
Base Folder where OS is stored	C drive	/

# Boot Process

- The Linux boot process is the procedure for initializing the system.
- It consists of everything that happens from when the computer power is first switched on until the user interface is fully operational.
- A good understanding of the steps in the boot process will help you with troubleshooting problems as well as with tailoring the computer's performance to your needs.





# GUI

- Common tasks using GUI
- Settings
- Menu
- Shortcuts
- Default Applications





# Desktop Environments - KDE

The screenshot displays the KDE desktop environment with a blue background. In the top-left corner, a 'wallpapers' window shows a gallery of desktop backgrounds. The top-center features a Konqueror web browser window displaying the openDesktop.org website, which lists various KDE applications and their ratings. On the bottom-left, a 'theme kde' help window provides information on how to change the desktop theme. The bottom-right corner contains a chat window with a list of messages from users like mtux, skreesh2, sheytan, and pano. The system tray at the bottom shows the KDE logo, several application icons, and a clock displaying 00:18.

Home > wallpapers

Air.jpg Atra Dot - Blue 1920x1200.jpg Atra Dot.jpg

Blue Curl.jpg Code Poets Dream.jpg Emotion.jpg

Exotic\_Imagination\_by\_MagicalVibe... Exotic saturnNov.jpg Fantasy\_Is\_celsojun...

Let's build the Desktop of the future - openDesktop.org - Konqueror

File Edit View Go Bookmarks Tools Settings Window Help

http://opendesktop.org/

openDesktop.org: Applications Artwork Linux Distributions Documents

openDesktop.org

Apps Artwork Groups Knowledge Base Events Forum People Jobs Register Login

view add content my content

CONTENT

All Office Multimedia Graphic Network Printing Text Editors Education Business Telephony Games Development Administration Scientific Security Utilities PDA Software Server Software Accessibility Embedded Libraries

latest alphabetical highest rated most downloaded

DNetStats 1.1.0 KDE Networking Tool 81% good Aug 1 2009 UPDATED more

JPass - Password Manager 0.1.6 Beta Java Security 50% good Aug 1 2009 FRESH more

Metis Dictionary 0.3.5 Java Education 50% good Aug 1 2009 FRESH more

JAG 0.2.5 Qt Arcade Game 27 comments 63% good Aug 1 2009 UPDATED more

JSame 0.1.1 50% good FRESH more

My openDesktop.org

my profile my messages my friends my groups my contents

STATUS MESSAGE

update

all messages

social desktop

Make the coolest Social Desktop App and win a

theme kde

How do I theme KDE 4?

Question: How do I theme KDE 4? Is there a difference between Plasmoid- and Widget-theme?

Answer: There are 2 possibilities how to change your desktop appearance. Firstly there are plasma themes. They only theme plasmoids and the panel. New plasma themes can be downloaded by using the "New Theme" button directly from the "Desktop Settings" dialogue or can be downloaded from [url]http://www.kde-look.org/[url]. If you would like to create your own plasma theme, follow this link for information on how to get started.

one item, page 1/1

KDE uBlog

jospoortv...

mtux 2 hours ago from choqoK

@iranbaan: هويت يك شهيد ديگر مشخص شد: مصطفی کيارستمي در نمازجمعه 26 تير به شهادت رسيد

mtux 2 hours ago from choqoK

ابن اميد هم كه هميشه وسط صحبت قطع مينه! (... (نميدونم ناراحت مينه كابل رو ميكشه! يا

skreesh2 2 hours ago from choqoK

@sheytan I'm guessing you just jumped to !kde 4.3 then. Or is this 4.4 ?

sheytan 2 hours ago from choqoK

Finally !kde sysetettings looks nice. And the new !plasma dialogs animation is cool too :D

pano 2 hours ago from choqoK

(6/10) KDE.news The Business Of Free

(6/10) nu.nl - Algemeen Canal Parade druk bezocht

(6/10) Z24 laatste nieuws Alimentatie steeds later betaald

Let's build the Desktop of the future

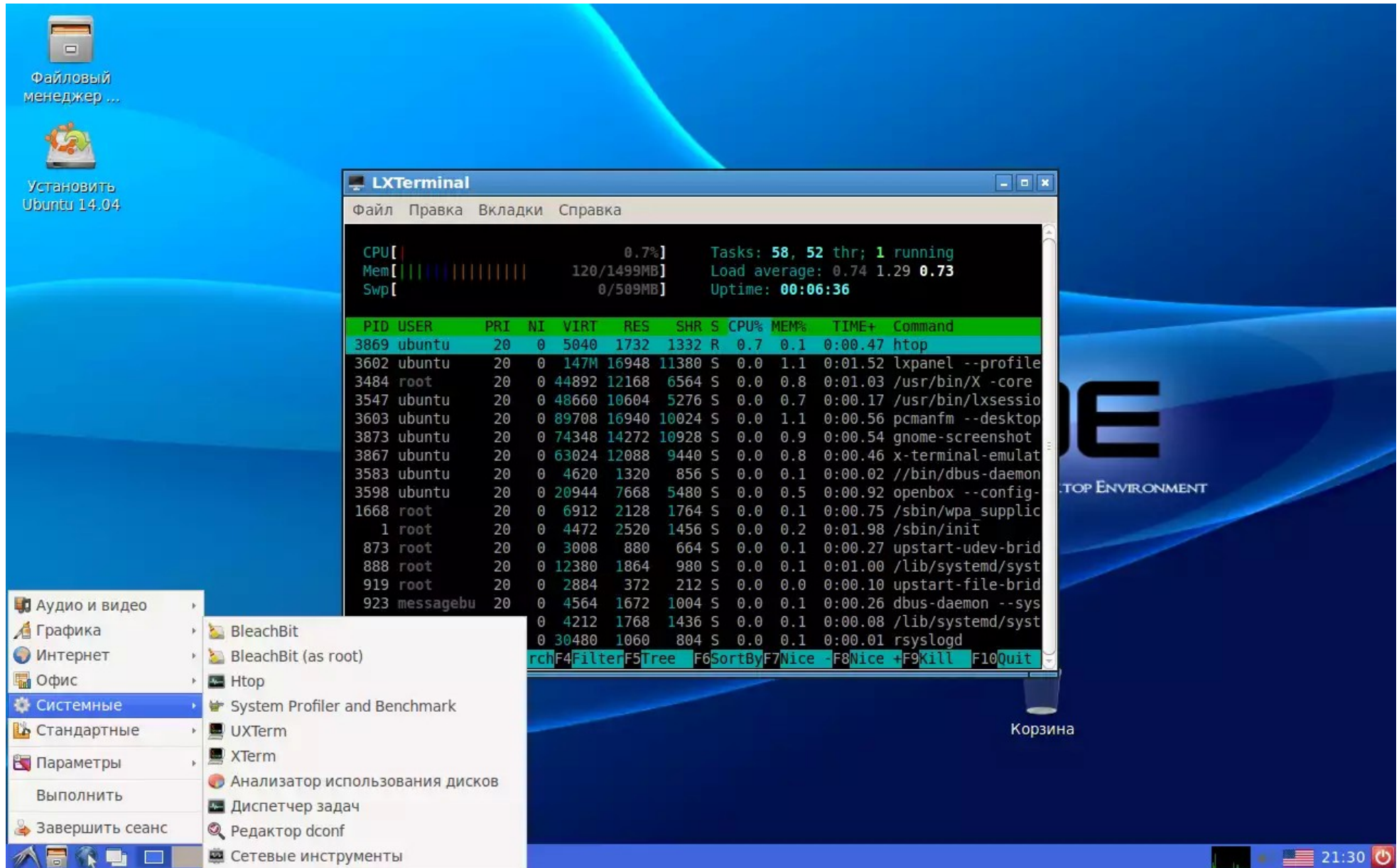
00:18

# Desktop Environments - GNOME





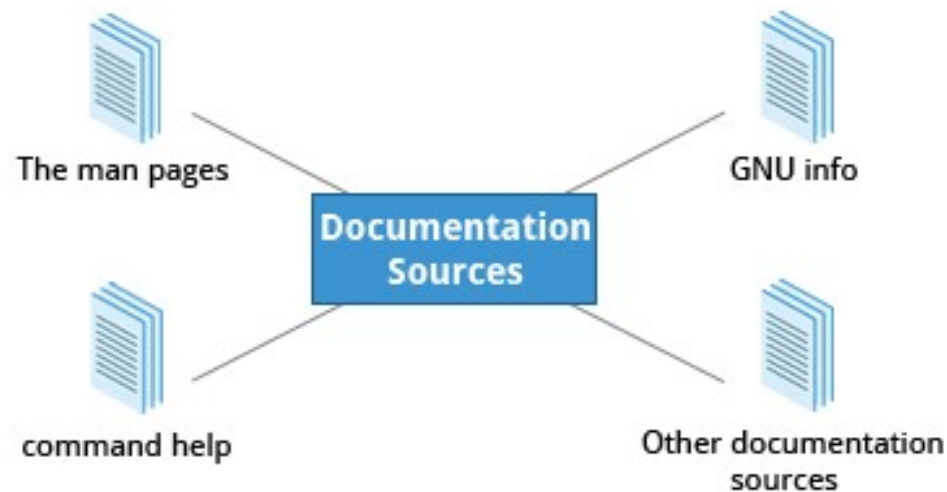
# Desktop Environments - LXDE



# Installing software via GUI

- Using Ubuntu software centre
- Synaptic Package manager

# Finding information - Linux Documentation



- Important Linux documentation sources include:
  - The man pages (short for manual pages)
  - GNU Info
  - The help command and --help option

# man

- Typing `man` with a topic name as an argument retrieves the information stored in the topic's man pages.
- Some Linux distributions require every installed program to have a corresponding man page, which explains the depth of coverage. (Note: `man` is actually an abbreviation for manual.)
- The man pages structure were first introduced in the early UNIX versions of the early 1970s.

# Try it out !!!

- man man
- man apropos



# Command Line Operations

- There is a saying, "graphical user interfaces make easy tasks easier, while command line interfaces make difficult tasks possible."
- Linux relies heavily on the abundance of command line tools.
- The command line interface provides the following advantages:
  - No GUI overhead.
  - Virtually every task can be accomplished using the command line.
  - You can script tasks and series of procedures.
  - You can log on remotely to networked machines anywhere on the Internet.
  - You can initiate graphical apps directly from the command line.

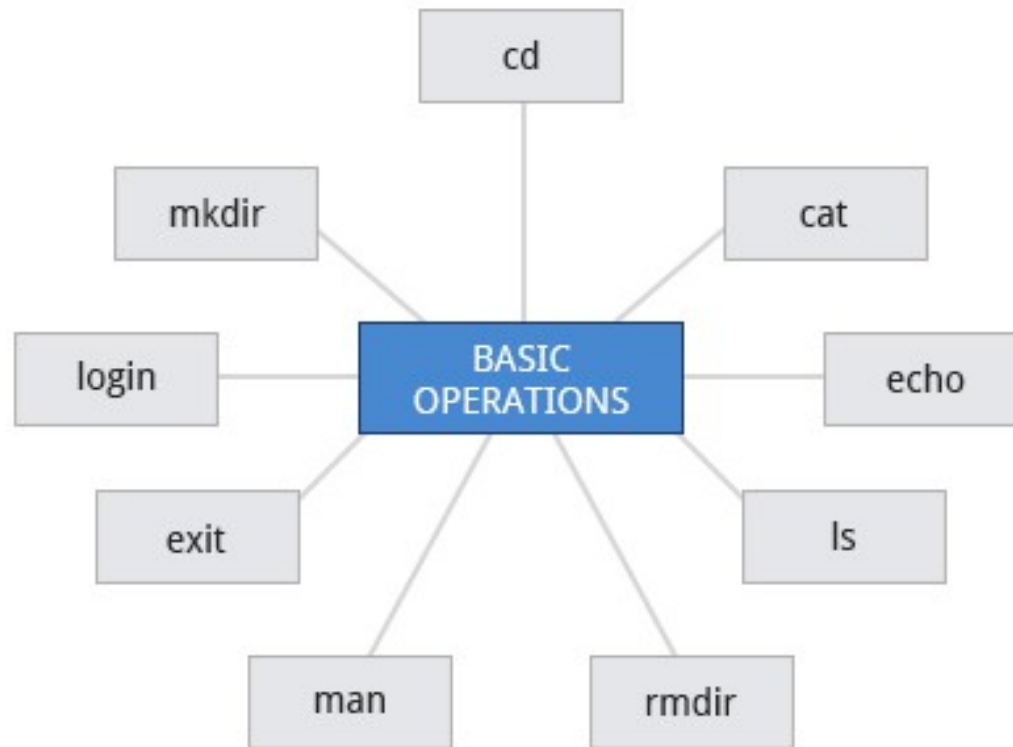
- Most input lines entered at the shell prompt have three basic elements:
  - Command
  - Options
  - Arguments

- A terminal emulator program emulates (simulates) a stand alone terminal within a window on the desktop. By this we mean it behaves essentially as if you were logging into the machine at a pure text terminal with no running graphical interface. Most terminal emulator programs support multiple terminal sessions by opening additional tabs or windows.

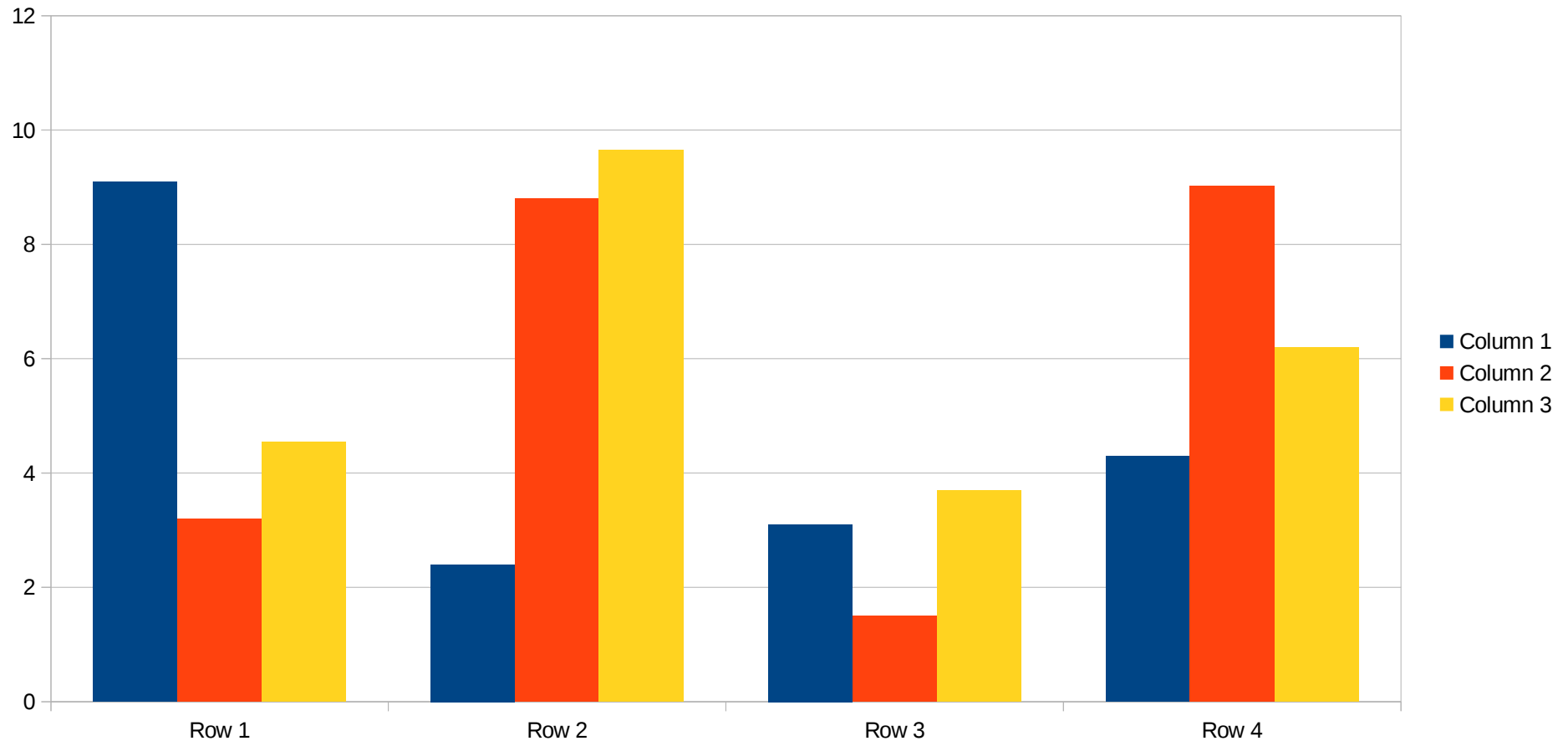
- xterm
- rxvt
- konsole
- terminator

- The customizable nature of Linux allows you to drop (temporarily or permanently) the X Window graphical interface, or to start it up after the system has been running.
- Certain Linux distributions distinguish versions of the install media between desktop (with X) and server (usually without X); Linux production servers are usually installed without X and even if it is installed, usually do not launch it during system start up.
- Removing X from a production server can be very helpful in maintaining a lean system which can be easier to support and keep secure.

# Basic Operations



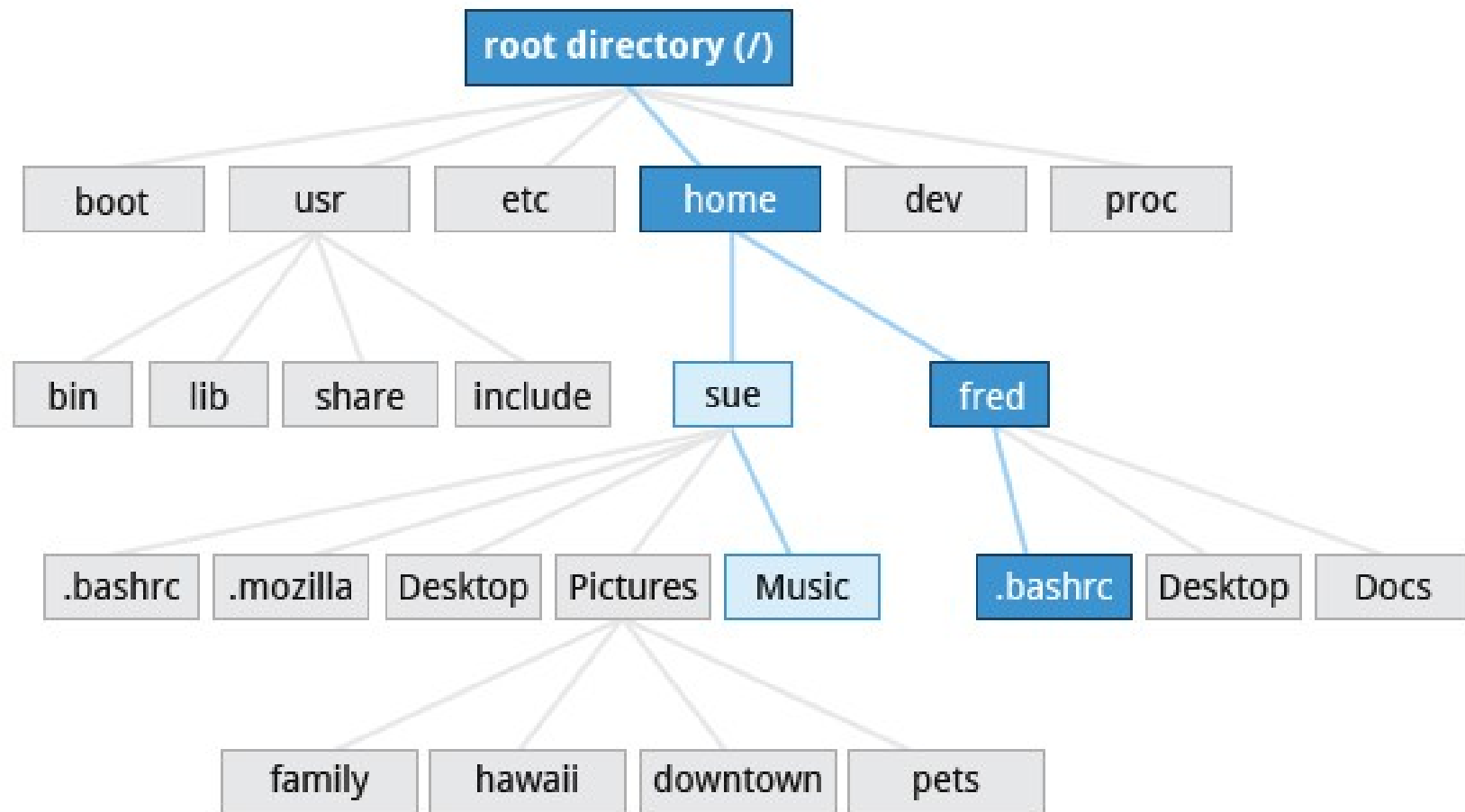
# Directory navigation



# Understanding Absolute and Relative Paths

- **Absolute pathname:** An absolute pathname begins with the root directory and follows the tree, branch by branch, until it reaches the desired directory or file. Absolute paths always start with /.
- **Relative pathname:** A relative pathname starts from the present working directory. Relative paths never start with /.





1. In the above example, we use the relative path method to list the files under Music from your current working directory (sue)  
\$ ls ../sue/Music

2. In the above example, we use the Absolute pathname method to edit the .bashrc file:  
\$ gedit /home/fred/.bashrc

- Absolute pathname method: `$ cd /usr/bin`
- Relative pathname method: `$ cd ../../usr/bin`

# Exploring the Filesystem

- `cd /` Changes your current directory to the root (/) directory (or path you supply)
- `ls` List the contents of the present working directory
- `ls -a` List all files including hidden files and directories (those whose name start with . )
- `tree` Displays a tree view of the filesystem

# Hard and soft links

- Symbolic link is the linux equivalent of Windows' "Shortcuts"
- Symbolic links take no extra space on the filesystem (unless their names are very long). They are extremely convenient as they can easily be modified to point to different places. An easy way to create a shortcut from your home directory to long pathnames is to create a symbolic link.

- $\ln$
- $\ln -s$

Command	Usage
<b>cat</b>	Used for viewing files that are not very long; it does not provide any scroll-back.
<b>tac</b>	Used to look at a file backwards, starting with the last line.
<b>less</b>	Used to view larger files because it is a paging program; it pauses at each screenful of text, provides scroll-back capabilities, and lets you search and navigate within the file. Note: Use <code>/</code> to search for a pattern in the forward direction and <code>?</code> for a pattern in the backward direction.
<b>tail</b>	Used to print the last 10 lines of a file by default. You can change the number of lines by doing <code>-n 15</code> or just <code>-15</code> if you wanted to look at the last 15 lines instead of the default.
<b>head</b>	The opposite of <b>tail</b> ; by default it prints the first 10 lines of a file.

# Creating a Directory and touch

- `$ mkdir`
- `$ touch`

Command	Usage
<b>cat</b>	Used for viewing files that are not very long; it does not provide any scroll-back.
<b>tac</b>	Used to look at a file backwards, starting with the last line.
<b>less</b>	Used to view larger files because it is a paging program; it pauses at each screenful of text, provides scroll-back capabilities, and lets you search and navigate within the file. Note: Use <code>/</code> to search for a pattern in the forward direction and <code>?</code> for a pattern in the backward direction.
<b>tail</b>	Used to print the last 10 lines of a file by default. You can change the number of lines by doing <code>-n 15</code> or just <code>-15</code> if you wanted to look at the last 15 lines instead of the default.
<b>head</b>	The opposite of <b>tail</b> ; by default it prints the first 10 lines of a file.



Command	Usage
<code>mv</code>	Rename a directory
<code>rmdir</code>	Remove an empty directory
<code>rm -rf</code>	Forcefully remove a directory recursively

- When commands are executed, by default there are three standard file streams (or descriptors) always open for use: standard input (standard in or stdin), standard output (standard out or stdout) and standard error (or stderr).
- Usually, stdin is your keyboard, stdout and stderr are printed on your terminal; often stderr is redirected to an error logging file.

- In Linux, all open files are represented internally by what are called file descriptors. Simply put, these are represented by numbers starting at zero. stdin is file descriptor 0, stdout is file descriptor 1, and stderr is file descriptor 2. Typically, if other files are opened in addition to these three, which are opened by default, they will start at file descriptor 3 and increase from there.

- I/O Redirection
  - Through the command shell we can redirect the three standard filestreams so that we can get input from either a file or another command instead of from our keyboard, and we can write output and errors to files or send them as input for subsequent commands.
- 
- `$ do_something < input-file`
  - `$ do_something > output-file`

# Pipes

- The UNIX/Linux philosophy is to have many simple and short programs (or commands) cooperate together to produce quite complex results, rather than have one complex program with many possible options and modes of operation. In order to accomplish this, extensive use of pipes is made; you can pipe the output of one command or program into another as its input.

- `$ command1 | command2 | command3`

# locate

- `$ locate passwd | wc`

You can search for a filename containing specific characters using **wildcards**.

Wildcard	Result
<code>?</code>	Matches any single character
<code>*</code>	Matches any string of characters
<code>[set]</code>	Matches any character in the set of characters, for example <code>[adf]</code> will match any occurrence of "a", "d", or "f"
<code>[!set]</code>	Matches any character not in the set of characters

To search for files using the `?` wildcard, replace each unknown **character** with `?`, e.g. if you know only the first 2 letters are 'ba' of a 3-letter filename with an extension of `.out`, type `ls ba?.out`.

To search for files using the `*` wildcard, replace the unknown **string** with `*`, e.g. if you remember only that the extension was `.out`, type `ls *.out`

# find

- find is extremely useful and often-used utility program in the daily life of a Linux system administrator. It recurses down the filesystem tree from any particular directory (or set of directories) and locates files that match specified conditions. The default pathname is always the present working directory.



- Searching for files and directories named "gcc":
  - `$ find /usr -name gcc`
- Searching only for directories named "gcc":
  - `$ find /usr -type d -name gcc`
- Searching only for regular files named "test1":
  - `$ find /usr -type f -name test1`

# find and exec

- Another good use of find is being able to run commands on the files that match your search criteria. The -exec option is used for this purpose.
- To find and remove all files that end with .swp:
- `$ find -name "*.swp" -exec rm {} ';' ;`
- The {} (squiggly brackets) is a place holder that will be filled with all the file names that result from the find expression, and the preceding command will be run on each one individually.
- Note that you have to end the command with either ';' (including the single-quotes) or \; Both forms are fine.









# Installing Software

- The Advanced Packaging Tool (apt) is the underlying package management system that manages software on Debian-based systems. While it forms the backend for graphical package managers, such as the Ubuntu Software Center and synaptic, its native user interface is at the command line, with programs that include apt-get and apt-cache.

- Yellowdog Updater Modified (yum) is an open-source command-line package-management utility for RPM-compatible Linux systems, basically what we have called the Fedora family. yum has both command line and graphical user interfaces.
- zypper is a package management system for openSUSE that is based on RPM. zypper also allows you to manage repositories from the command line. zypper is fairly straightforward to use and resembles yum quite closely.





**Linux  
Package Manager**



**Debian Family  
System**



**apt-get**



**High Level  
Package Manager**



**SUSE Family  
System**



**zypper**



**Fedora Family  
System**



**yum**

sudo