

Using Linux

System and Network administration

Anupa Shyamlal

University of Colombo School of Computing

Linux Installation

Basic guide to install Linux based OS

Linux Versions

- Linux isn't a complete operating system — it's just a kernel.
- Linux kernel combined with other free software creates the complete package.
- Some of the famous distributions:
 - Ubuntu : Well-known Linux distribution. Based on Debian.
 - Mint : Built on top of Ubuntu. Included media codecs and proprietary softwares.
 - Debian : Since 1993. Moving much more slowly than distributions like Ubuntu.
 - CentOS / Red Hat : A commercial Linux distribution. Intended for servers.
 - Arc : Designed to be flexible, lightweight, minimal. Highly configurable.
 - Puppy Linux : Small, lightweight operating system

Ref: https://en.wikipedia.org/wiki/List_of_Linux_distributions

Ubuntu Desktop vs Server

- Ubuntu Server has inbuilt standard set of packages most used on servers, like,
 - web server
 - email server
 - file server
 - samba server...etc.
 - No GUI environment by default (no KDE, no Gnome, no X11 services*)
- The Ubuntu Desktop assumes you have a video card and a monitor, and it installs X11, a desktop environment, and a variety of user applications like a Office-like application suite, a web browser, a music player, etc.

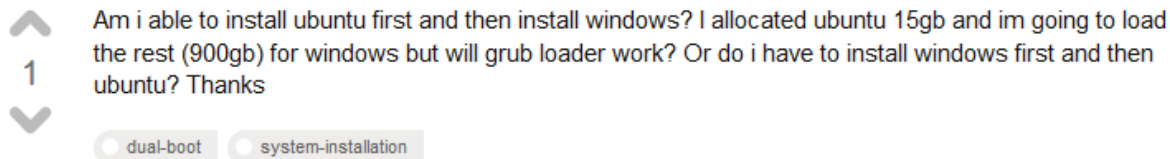
* https://en.wikipedia.org/wiki/X_Window_System

Installation

- Now it is easy as installing a common software
- Installing Ubuntu
 - <http://www.linuxtechi.com/install-ubuntu-16-04-with-screenshots/>
- Installing Ubuntu Server (bit different, includes networking stuff)
 - <http://www.wikihow.com/Install-Ubuntu-Server>

Buzz words

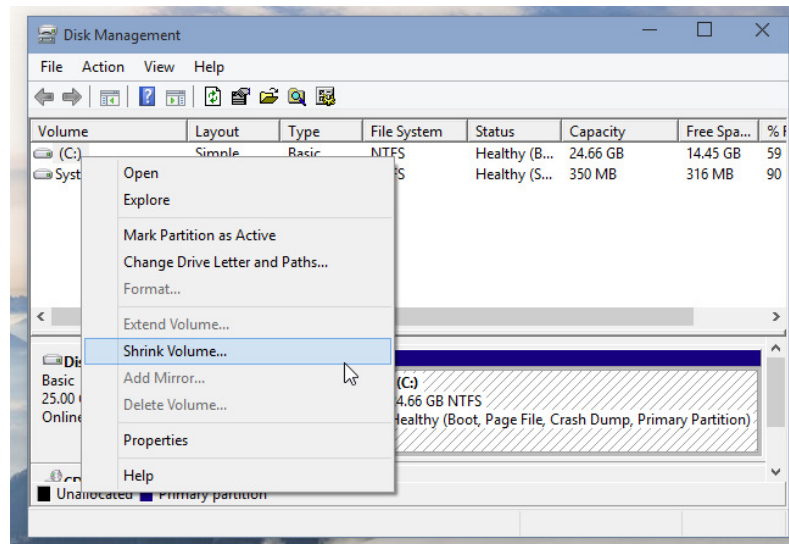
- Grub (Grand Unified Bootloader)
 - A Linux bootloader.
 - New version: Grub2
 - Capable of loading a variety of free and proprietary operating systems.
 - If you have multiple kernel images installed on your system, you can choose which one to be executed via the GRUB.



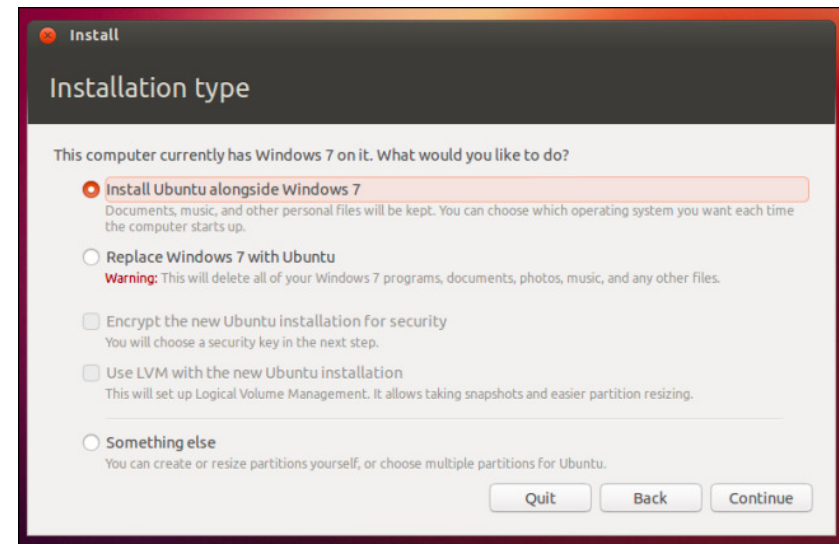
When you install Windows OS, the Windows MBR(Master Boot Record) will wipe out, or rather over-write, every other boot loader available in the system.

Buzz Words

- Dual Partition
 - **Install Windows first:** If you install Linux second, it can set up its boot loader properly to happily co-exist with Windows. If you install Windows second, it will ignore Linux, and you'll have to go through some trouble to get your Linux boot loader working again.
 - **Make space for Linux:** If not available, need to resize Windows partitions to make room for Linux. Software: DiskPart, Minitool Partition Wizard
 - **Install Linux Second:** Boot from a bootable drive making sure you select the option that installs it alongside Windows — don't tell it to wipe your hard drive.



Making space



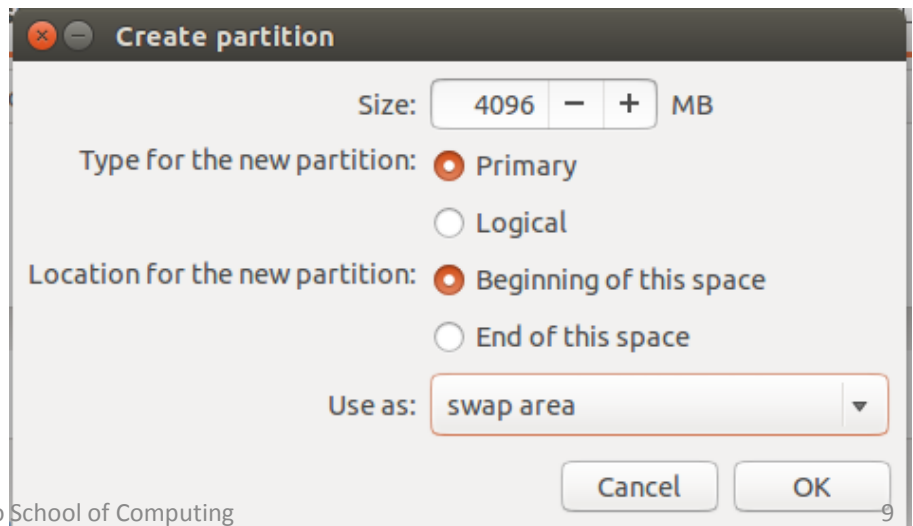
Don't tell it to wipe your hard drive unless you have nothing to loose.

Reference: <https://help.ubuntu.com/community/WindowsDualBoot>
University of Colombo School of Computing

Buzz Words

- Swap Partition

- Linux divides its physical RAM into chunks of memory called pages.
- Swapping is the process whereby a page of memory is copied to the preconfigured space on the hard disk, called swap space, to free up that page of memory.
- When the system requires more memory than is physically available, the kernel swaps out less used pages and gives memory to the current application (process) that needs the memory immediately



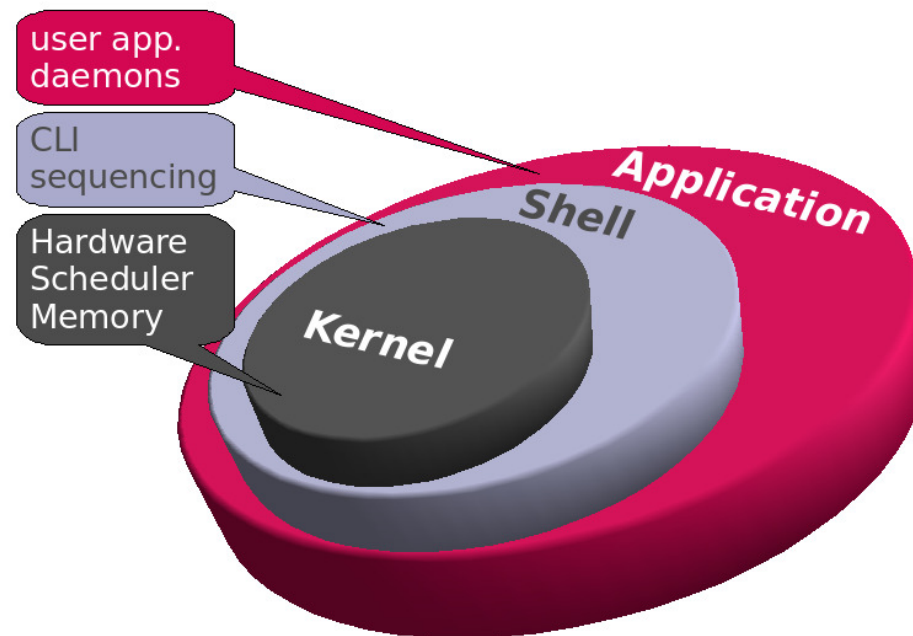
Virtual Machines

- Still afraid of loosing data after Linux installation? Try this.
- Emulation of a computer system.
- Important topics:
 - Guest additions
(<https://www.howtogeek.com/howto/2845/install-guest-additions-to-windows-and-linux-vms-in-virtualbox/>)
 - Shared folders
 - Drag n Drop
 - Shared Clipboard
 - Networking
(https://blogs.oracle.com/fatbloke/entry/networking_in_virtualbox1)
 - Network Address Translation (NAT)
 - Bridged networking
 - Internal networking
 - Host-only networking
 - NAT with Port-forwarding

Linux Commands

Basic guide to Linux shell commands

Shell



Shells



- The interactive interface between user and operating system.
- Layer of programming that understands and executes the commands, a user enters. (command interpreter.)
- UNIX provides several shells,
 - Bourne shell – sh
 - Korn shell – ksh
 - C shell – csh
 - Bash – bash
 - Enhanced C shell (a freeware shell derived from the C shell) – tcsh
 - Z shell (a freeware shell derived from the Korn shell) – zsh
- Linux uses the freeware Bash shell as its default command interpreter (compatible with Bourne shell, created & distributed by the GNU project)

Why CLI?

- Memorization and familiarity needed
- Only need to utilize a keyboard to navigate the interface, often resulting in faster performance.
- Lot less of the computer's system resources needed than a GUI.
- Allows a single action to be taken to achieve results over large numbers of discrete tasks.
- Scriptability
 - Eg: Delete all files in the current directory and all directories beneath that which start with an r and end in .txt: `$ rm **/r*.txt`
- Automating tasks : cron and bash scripts

More: <http://westminster.lib.co.us/linux/cli-vs-gui.html>

Logged-in name (`whoami`): howtogeek
Computer-name (`hostname`): Ubuntu
Current location (`pwd`) : ~/Desktop

Prompt

```
howtogeek@ubuntu: ~/Desktop$
```

```
howtogeek@ubuntu:~$ ls  
Desktop    examples.desktop  pidgin    timer.sh  
Documents  Music             Public    Ubuntu One  
Downloads  Pictures          Templates Videos  
howtogeek@ubuntu:~$ cd Desktop  
howtogeek@ubuntu:~/Desktop$
```

Commands

Output

Syntax

`command [-argument] [-argument] [--argument] [file]`

Example:

- `ls` : List files in current directory
- `ls -l` : Lists files in “long” format
- `ls -l --color` : As above, with colored output

Arguments can be a file name:

- `cat sample.txt` : Show contents of a file
- `cat -n sample.txt` : Show contents of a file, with line numbers

Some other characters : `>`, `>>`, `|`, `,`, `;` (later)

Basic Linux Commands

- Reference : linux-training.be/linuxfun.pdf
- Content
 - Working with directories
 - File contents
 - Text Editors
 - Sudo
 - Piping and redirection
 - Filters
 - Other

Working with Directories

- pwd
- cd
 - Home directory: ~
 - Parent Directory: ..
- cp – copy files
- mv – move files
- rm – remove files
- rmdir – remove directories
- touch – make a new, empty file
- mkdir – make a new, empty directory
- file : outputs type of the file
- locate : search for files anywhere in the file system

Relative path vs Absolute path

- 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. (But windows??)

- Absolute path

The specifying the location of a file or directory from the root directory(/).

A complete path from start of actual filesystem from / directory.

`/var/ftp/pub`

`/boot/grub/grub.conf`

- Relative path

Path related to the present working directory(pwd).

Suppose pwd is `/var/log` and need to change directory to `/var/log/kernel`.

`$cd kernel` (Using relative path)

`$cd /var/log/kernel` (Using absolute path)

Exercise

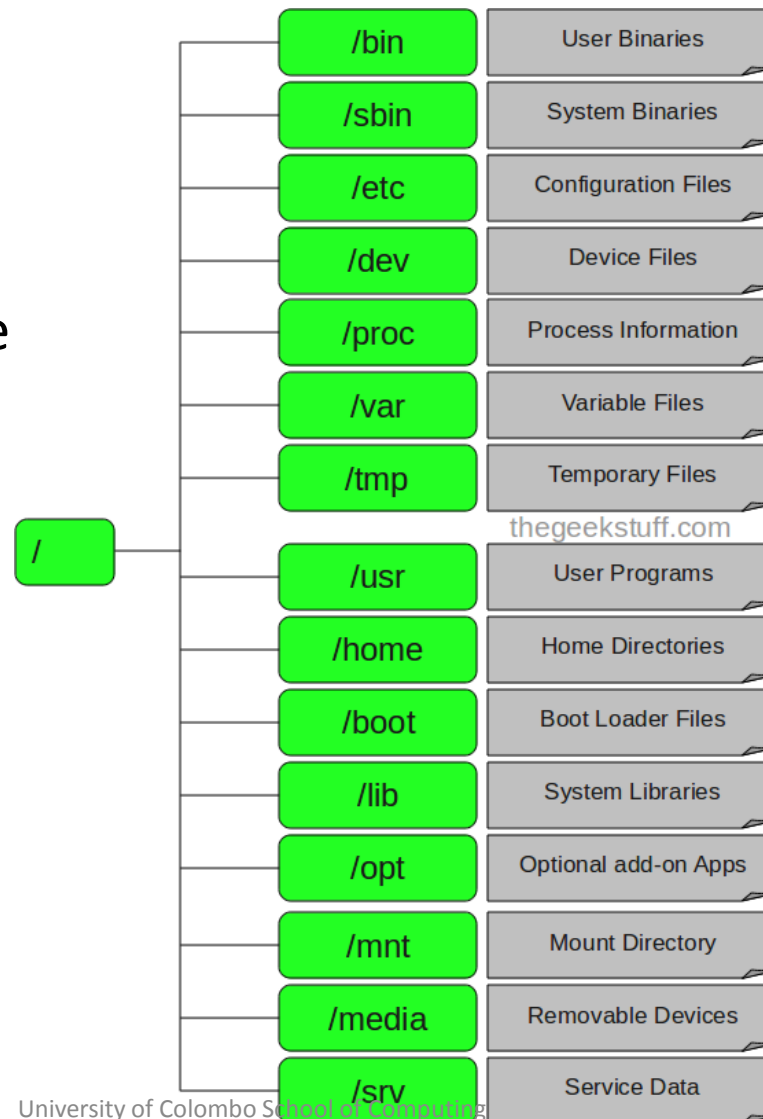
- Assume that your current working directory is '/tmp' and your home directory is '/home/amal'. Which of the below commands will copy all the content of '/tmp/test/' to a 'test' subdirectory of your home directory?
 - `cp -r test/* /home/amal`
 - `cp -r ./test ~`
 - `cp -r ~/test .`
 - `cp -r /tmp/test /home/amal/test`

Exercise

- Assume you are in home folder.
- Go to `/usr/local/lib`
- Now find the root (the ultimate destination)
- What are the directories under root.
- Visualize the directory structure in Linux.
- Go inside bin directory.
- Can you find familiar names?

Exercise

- Linux directory structure

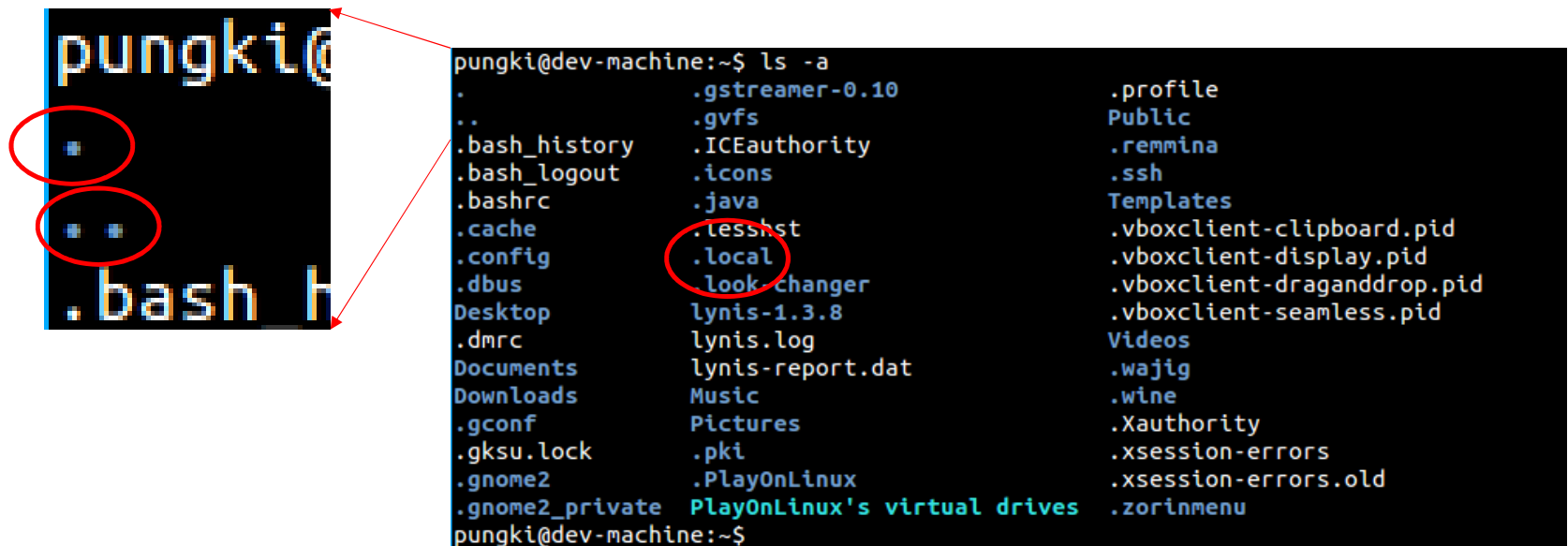


```

root@musang:~# cd /bin/
root@musang:/bin# ls
Mail@      dirname*      id*           ntfs-3g*     sha224sum*   umount*
[*]        dmesg*        install*      ntfs-3g.probe* sha256sum*   uname*
arch*      dnsdomainname@ ipmask*       ntfsclust*   sha384sum*   uncompress@
ash*       domainname@   join*         ntfscluster* sha512sum*   unexpand*
awk@       du*           kill*         ntfsicmp*    shred*        uniq*
base64*    echo*         killall*      ntfsfix*     shuf*         unlink*
basename*  ed@           ksh*          ntfsinfo*    sleep*        users*
bash*      egrep@        link*         ntfsls*      sln@          usleep*
bunzip2@   env*          ln*           od*           sort*         vdir*
bzipcat@   expand*        loadkeys*     paste*        split*        wc*
bzip2*     expr*         login*        pathchk*      stat*         which*
bzip2recover* factor*        logname*      ping*          stty*         who*
cat*       false*        ls*           ping6*         su*           whoami*
chgrp*     fgrep@        lsmod@        pinky*         sulogin@      yes*
chmod*     fmt*          mail@         pr*            sum*          ypdomainname@
chown*     fold*         md5sum*       printenv*     sync*         zcat*
chroot*    free*         mkdir*        printf*        tac*          zcmp*
cksum*     ftp*          mkiifd*       ps*            tail*         zdiff*
comm*      fusermount*   mkfs*         ptx*           tar*          zegrep*
compress@  gawk@         mount*        pwd*           tar-1.13*     zfgrep*
cp*        gawk-3.1.5*   more*         readlink*     tar-1.16.1@  zforce*
cpio*      getopt*       mt@           rksh@          tee*          zgrep*
csh@       ginstall@     mt-GNU*       rm*            telnet*       zless*
csplit*    grep*         mt-st*        rmdir*         test*         zmore*
cut*       groups*       mv*           rpm*           touch*        znew*
date*      gunzip*       nail@         rzcp*          true*         zsh*
dd*        gzexe*        netstat*      sed*           tr*           zsh-4.3.6@
df*        gzip*         nice*          seq*           tty*
dialog*    head*         nisdomainname@ setterm*       tsort*
dir*       hostid*       nl*           sh@            ttym*
dircolors* hostname*      nohup*        sha1sum*       ulockmgr_server*
root@musang:/bin# _

```


- Current directory, Parent directory, Hidden directory/file



The diagram illustrates the mapping between terminal symbols and files in a directory listing. On the left, a terminal window shows the prompt 'pungki@' and the command '.bash H'. Two red circles highlight the '.' and '..' symbols. Red arrows point from these circles to the corresponding entries in a directory listing on the right. The listing shows that '.' corresponds to the current directory, '..' to the parent directory, and '.local' to a hidden directory.

```
pungki@dev-machine:~$ ls -a
.          .gvfs          .profile
..         .gvfs          Public
.bash_history .ICEauthority .remmina
.bash_logout .icons         .ssh
.bashrc      .java          Templates
.cache       .lessht        .vboxclient-clipboard.pid
.config      .local         .vboxclient-display.pid
.dbus        look-changer   .vboxclient-draganddrop.pid
Desktop      lynis-1.3.8    .vboxclient-seamless.pid
.dmrc        lynis.log      Videos
Documents    lynis-report.dat .wajig
Downloads    Music          .wine
.gconf       Pictures       .Xauthority
.gksu.lock   .pki          .xsession-errors
.gnome2      .PlayOnLinux  .xsession-errors.old
.gnome2_private PlayOnLinux's virtual drives .zorinmenu
pungki@dev-machine:~$
```

Remember All?

- Use Bash built-in Help
 - Many commands have simple “help” screens that can be invoked with special command flags
 - These flags usually look like “-h” or “--help”
 - `grep -help`
- Online Manuals: “Man Pages”
 - `man ls`
 - `man man`
- If you cant remember the name of Linux command and you need to search for it
 - `man -k permission`
- Info Pages
 - Info pages are similar to man page, but instead of being displayed on one long scrolling screen, they are presented in shorter segments with links to other pieces of information.
 - `Info ls`

Exercise

- Which of the command are you likely to use to display hypertextual documentation on a command?
- Which Linux command can be used to create a new user account?

Tricks

- Shells support certain keystrokes for performing command-line editing
 - For example, Bash supports the left and right arrow keys, which move the cursor on the command line
 - Not all shells support command-line editing in the same manner
- Multiple Command Entry
 - You may type more than one command on the command line by separating each command with a semicolon(;)
 - When you press Enter, UNIX executes the commands in the order you entered them
- You can use the clear command to clear your screen; it has no options or arguments
- **You can access the command history with the up and down arrow keys with most shells**
- **Path Completion**

File Content

- File creation: touch
- View files: head, tail, cat, tac, more, less and strings

Exercise

- Locate the file called, “passwd” file.
- Open the complete file using `cat` command.
- Instead of displaying whole file, display the last 3 lines of the file.

Text editors

- nano
- vi
- vim

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

Esc
normal mode

~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
. goto mark	1	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= autoformat
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.
q record macro	w next word	e end word	r replace char	t 'till	y yank	u undo	i insert mode	o open below	p paste after	[misc]	misc
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	.	ex cmd line	" reg. spec	bol/ goto col
a append	s subst char	d delete	f find char	g extra cmds	h ←	j ↓	k ↑	l →	.	repeat t/T/t/F	' goto mk. bol	\ not used!
Z quit	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un-indent	> indent	?	find (rev.)		
Z extra cmds	X delete char	c change	V visual mode	b prev word	n next (find)	m set mark	.	reverse t/T/t/F	.	repeat cmd	/	find

motion	moves the cursor, or defines the range for an operator
command	direct action command, if red, it enters insert mode
operator	requires a motion afterwards, operates between cursor & destination
extra	special functions, requires extra input
q.	commands with a dot need a char argument afterwards
bol = beginning of line, eol = end of line, mk = mark, yank = copy	
words: quux(foo, bar, baz);	
WORDS: quux(foo, bar, baz);	

Main command line commands ('ex'):

:w (save), :q (quit), :q! (quit w/o saving)
:e f (open file f),
:%s/x/y/g (replace 'x' by 'y' filewide),
:h (help in vim), :new (new file in vim),

Other important commands:

CTRL-R: redo (vim),
CTRL-F/-B: page up/down,
CTRL-E/-Y: scroll line up/down,
CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*) (e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top, zb: bottom, zz: center
- (6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Exercise

- As a brief introduction to vi, go through the following: First, type **vi x**
- To put something in it, type the letter **i** (it stands for "insert-text mode"), and type the following (including hitting the Enter key at the end of each of the three lines):

*This is
My first
vi Text*

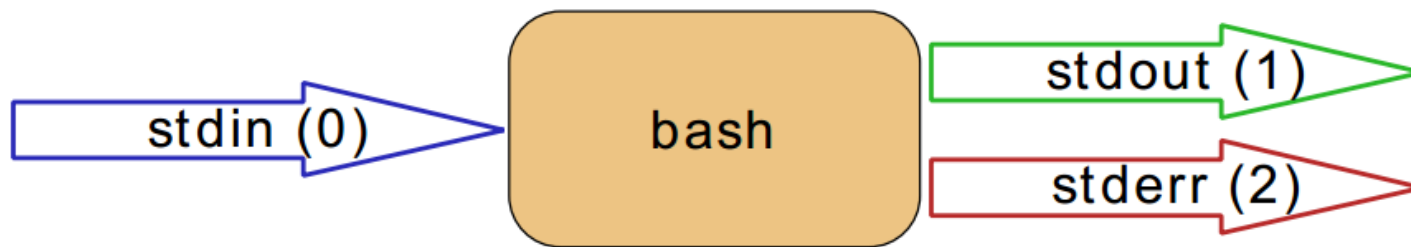
- Then hit the Escape key, to end insert-text-mode.
- Now save the file and exit vi, by typing **ZZ** (note the capitals).
- The key to learning vi is to keep in mind always the difference between insert-text mode and command mode.
- Type **more x** to display your file
- Now let's see how we can use vi again to modify that file. Type **vi x**
- We need to first move the cursor to the word "first". To do this, type **/fi** and hit the Enter key, which instructs vi to move the cursor to the first instance of **/fi** relative to the current cursor position.
- Hit **i**, and remove the word **first** then type **second** (note the space), and then hit Escape.
- Hit **d** twice which removes the line completely.
- Save and exit vi.

Sudo

- sudo allows a permitted user to execute a command as the superuser or another user, as specified in the *sudoers* file.
- By default, sudo requires that users authenticate themselves with a password.
- By default this is the user's password, not the root password itself.
- Examples:
 - `sudo apt-get install {package-name}`
 - `sudo su`

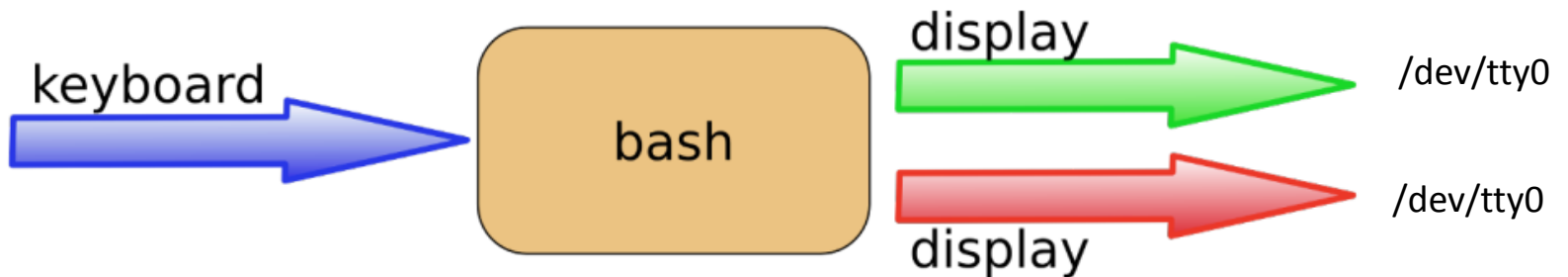
Piping and Redirection

- Every program we run on the command line automatically has three data streams connected to it.
 - STDIN (0): data fed into the program
 - STDOUT (1): data printed by the program, defaults to the terminal
 - STDERR (2): for error messages, also defaults to the terminal



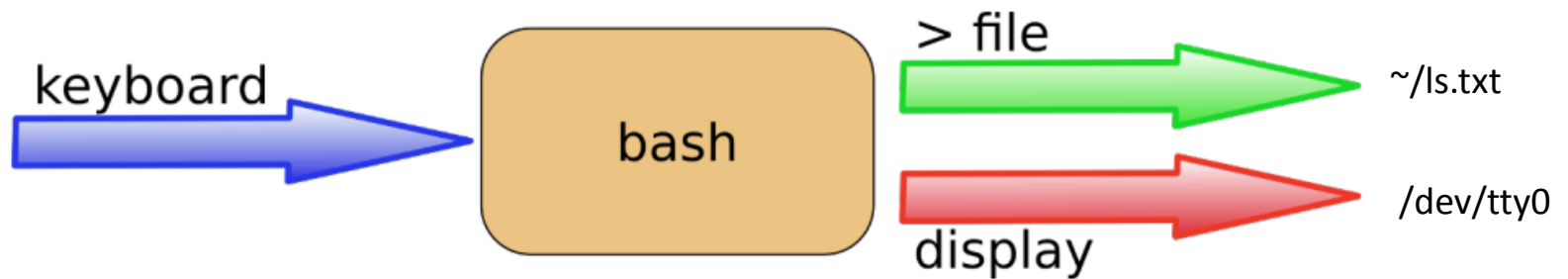
Piping and Redirection

- Every program we run on the command line automatically has three data streams connected to it.
 - STDIN (0): data fed into the program
 - STDOUT (1): data printed by the program, defaults to the terminal
 - STDERR (2): for error messages, also defaults to the terminal



Output redirection

- `>` and `>>`
- `ls -l > ~/ls.txt`

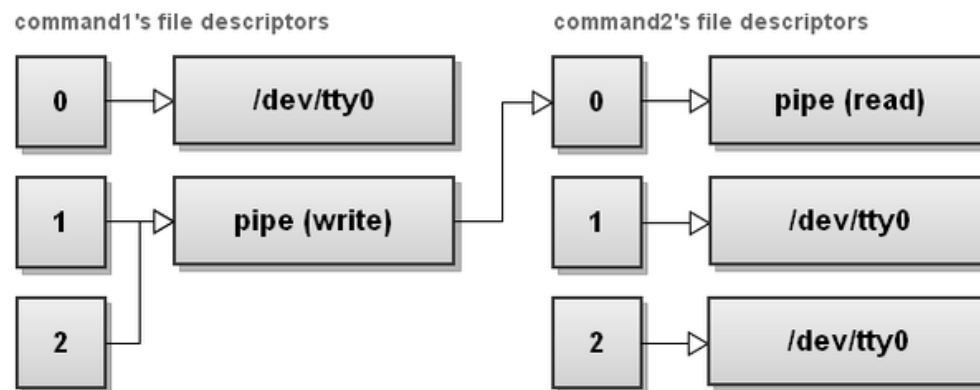


Piping

- Mechanism for sending data from one program to another.

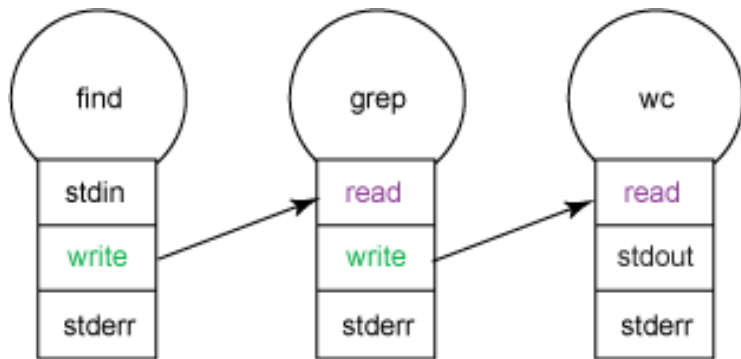
```
$ command1 | command2
```

- The pipe symbol (|) is used to indicate that the stdout of the first process should be the stdin of the second process:

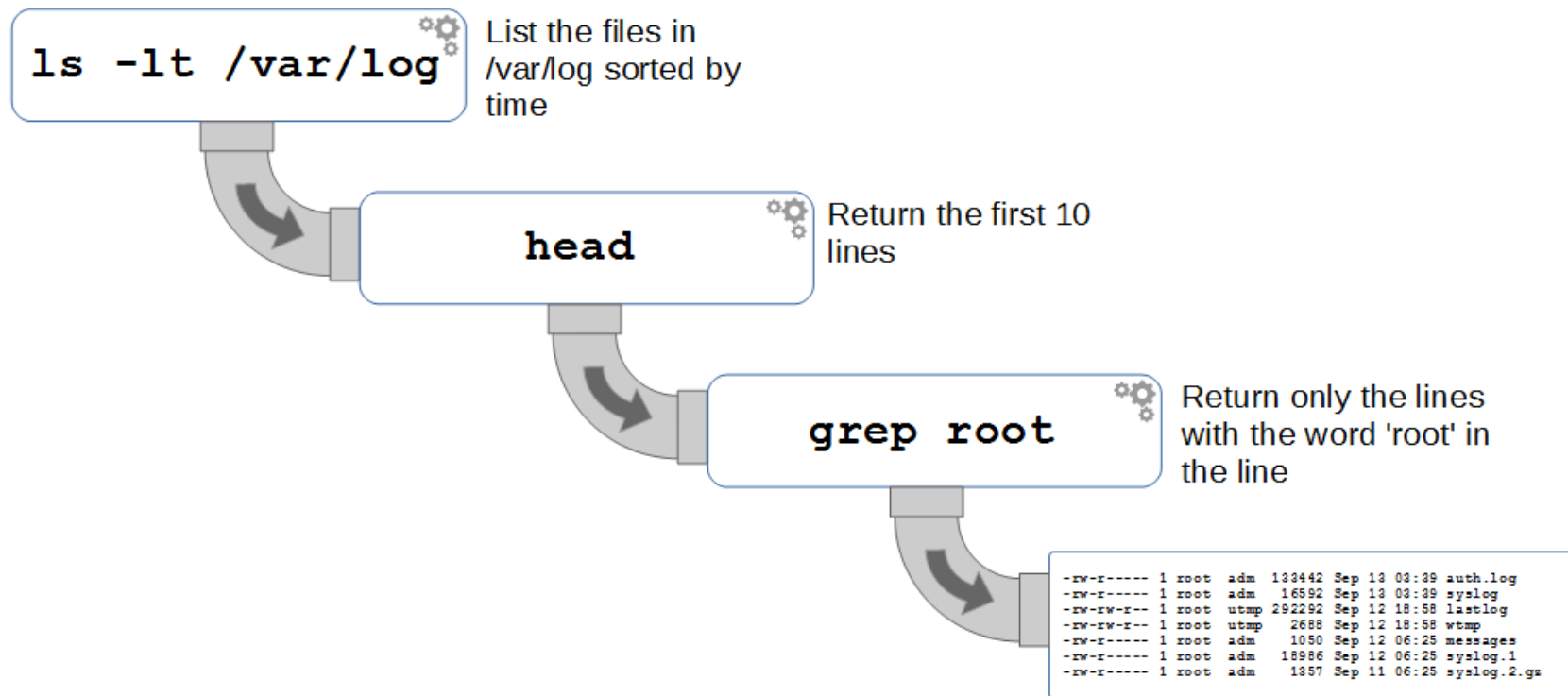


Example

```
find . -name "*.txt" | grep -i "abc" | wc
```



Example



Examples

- `wc -l barry.txt > myoutput`
- `ls >> myoutput`
- `wc -l < myoutput`
- `wc -l < barry.txt > myoutput`
- `ls | head -3`
- `ls | head -3 | tail -1`
- `ls | head -3 | tail -1 > myoutput`
- `echo -e "orange \npeach \ncherry" | sort > fruit`

Filters

cat*	Copies input direct to output.
head*	Shows beginning of a file (default 10 lines).
tail*	Shows end of a file (default 10 lines).
wc*	Counts characters, words and lines.
sort*	Sorts input lines.
grep*	Shows lines that match a regular expression.
tr	Translates or deletes specified character sets.
sed*	Stream editor.
uniq*	Discards all but one of successive identical lines.
awk	Highly-programmable field-processing.

Grep

- `grep -F` : grep evaluates PATTERN string as a "fixed string" — every character in your string is treated literally.
- `grep -E` : grep evaluates PATTERN string as an extended regular expression (ERE).
- `grep -r` : grep performs its search recursively.
- Arguments: `i`, `v`, `A`, `B`, `C`
- `cat /etc/passwd | grep "var" | grep -v "spool" | wc -l`
- `ifconfig | grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b"`

Sed

- The stream editor sed supports automated editing of text and is a more flexible filter than most.
- It streams its input (either stdin, or a specified input file) line by line, applies one or more editing operations, and writes the resulting lines to stdout.
 - `echo level5 | sed 's/5/42/'`
 - `echo "123 abc" | sed 's/[0-9]*/& &/'`
 - `sed 's/:.*//' /etc/passwd`

Special Characters

Character	Description	Example
	Redirect output of a command into another command	ls more
>	Redirect output of a command into a new file. If the file already exists, over-write it.	ls > myfiles.txt
>>	Redirect the output of a command onto the end of an existing file.	echo "Roshan 555" >> RNR.txt
;	Command separator. Allows you to execute multiple commands on a single line	cd /var/log; less messages
A few others: [], <, &&, &		

Character	Description	Example
\	Escape character	
/	Directory separator	/usr/src/linux
▪	Current directory. Can also “hide” files when it is the first character in a filename	
..	Parent directory	
~	User's home directory	
*	Represents 0 or more characters in a filename, or by itself, all files in a directory	pic*2002 can represent the files pic2002, picJanuary2002
?	Represents a single character in a filename.	hello?.txt can represent hello1.txt, helloz.txt, but not hello22.txt

Other Commands

- who, whoami
- uptime
- lsof
- tar
- ps
- scp
- wget

Find

- `find . -name "*.txt"`
- `find /home -iname tecmint.txt`
- `find / -type d -name Tecmint`
- `find . -type f -perm 0777 -print`
- `find / -perm /a=x`
- `find / -cmin -60`
- `find / -amin -60`
- `find / -size +50M -size -100M`

Questions?

Practicals

1. Working with directories and files and file content
2. Pipes, redirection and Linux filters practice