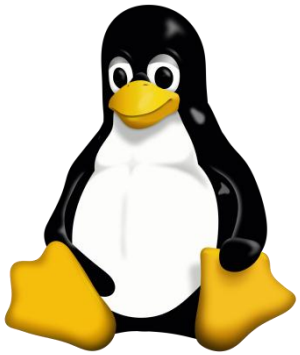
Linux Fundamentals

VERSION 3



Part 4

Eng Ali Mohammad. Bani Bakkar

Filters

Eng Ali Mohammad. Bani Bakkar
Email : alli_m_alqadri@hotmail.com

grep

- The **grep** is to filter lines of text containing (or not containing) a certain string.

$ cat tennis.txt

Amelie Mauresmo, Fra

Kim Clijsters, BEL

Justine Henin, Bel

Serena Williams, usa

Venus Williams, USA

$ cat tennis.txt | grep Williams

Serena Williams, usa

Venus Williams, USA

Example

```
$ echo "one Two three">>ali3.txt
$ echo "one two three">>ali3.txt
$ echo "three four five">>ali3.txt
$ grep two ali3.txt
one two three
$ cat ali3.txt | grep two
one two three

$ cat ali3.txt | xargs echo
one Two three one two three three four five
```

- Cat filename | grep –i “flag” (security tasks)
- Cat filename | grep –i “root\|password\|login” (security tasks)

Grep cont.

- You can write this without the cat.

$ grep Williams tennis.txt
Serena Williams, usa
Venus Williams, USA

- **grep -i** filters in a case insensitive way.

$ grep ali tennis.txt
Justine Henin, Bel
$ grep -i ali tennis.txt
filters
183
Kim Clijsters, BEL
Justine Henin, Bel

- **grep -v** outputs lines not matching the string.

$ grep -v Fra tennis.txt
Kim Clijsters, BEL
Justine Henin, ali
Serena Williams, usa
Venus Williams, USA

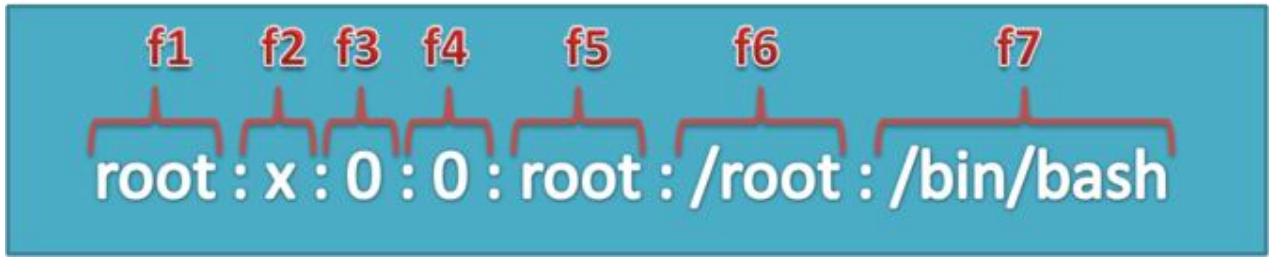
cut

- The **cut** filter can select columns from files, depending on a delimiter or a count of bytes.
- Use **cut** to filter for the username and userid in the **/etc/passwd** file.
- It uses the colon as a delimiter, and selects fields 1 and 3.

```
$ cut -d: -f1,3 /etc/passwd | tail -4
Figo:510
Pfaff:511
Harry:516
Hermione:517
```

- When using a space as the delimiter for **cut**, you have to quote the space.

```
$ cut -d" " -f1 tennis.txt
Amelie
Kim
Justine
Serena
Venus
```



tr

- You can translate characters with **tr**.

```
$ cat tennis.txt | tr 'e' 'E'
AmEliE MaurEsmo, Fra
Kim ClijstErs, BEL
JustinE HEnin, BEl
SErEna Williams, usa
VEnus Williams, USA
```

- Here we set all letters to uppercase by defining two ranges.

```
$ cat tennis.txt | tr 'a-z' 'A-Z'
AMELIE MAURESMO, FRA
KIM CLIJSTERS, BEL
JUSTINE HENIN, BEL
SERENA WILLIAMS, USA
VENUS WILLIAMS, USA
```

wc

- Counting words, lines and characters is easy with **wc**.

$ wc tennis.txt

5 15 100 tennis.txt

$ wc -l tennis.txt

5 tennis.txt

$ wc -w tennis.txt

15 tennis.txt

$ wc -c tennis.txt

100 tennis.txt

uniq

- With **uniq** you can remove duplicates from a **sorted list**.

```
$ cat music.txt
Queen
Brel
Queen
Abba
$ sort music.txt
Abba
Brel
Queen
Queen
$ sort music.txt |uniq
Abba
Brel
Queen
```

- **uniq** can also count occurrences with the **-c** option.

```
$ sort music.txt |uniq -c
1 Abba
1 Brel
2 Queen
```

od

- uses **od** to show the contents of the file in hexadecimal bytes

$ cat > text.txt

abcdefg

1234567

$ od -x text.txt

0000000 61 62 63 64 65 66 67 0a 31 32 33 34 35 36 37 0a

0000020

- The same file can also be displayed in octal bytes.

$ od -b text.txt

0000000 141 142 143 144 145 146 147 012 061 062 063 064 065 066 067 0120000020

pipe examples

who | wc

- How many users are logged on to this system ?

$ who
root tty1 Jul 25 10:50
paul pts/0 Jul 25 09:29 (laika)
Harry pts/1 Jul 25 12:26 (barry)
paul pts/2 Jul 25 12:26 (pasha)
$ who | wc -l
4
Display a sorted list of logged on users.
$ who | cut -d' ' -f1 | sort
Harry
paul
paul
root

pipe examples

- Display a sorted list of logged on users, but every user only once .

```
$ who | cut -d' ' -f1 | sort | uniq
Harry
paul
root
```

grep | cut

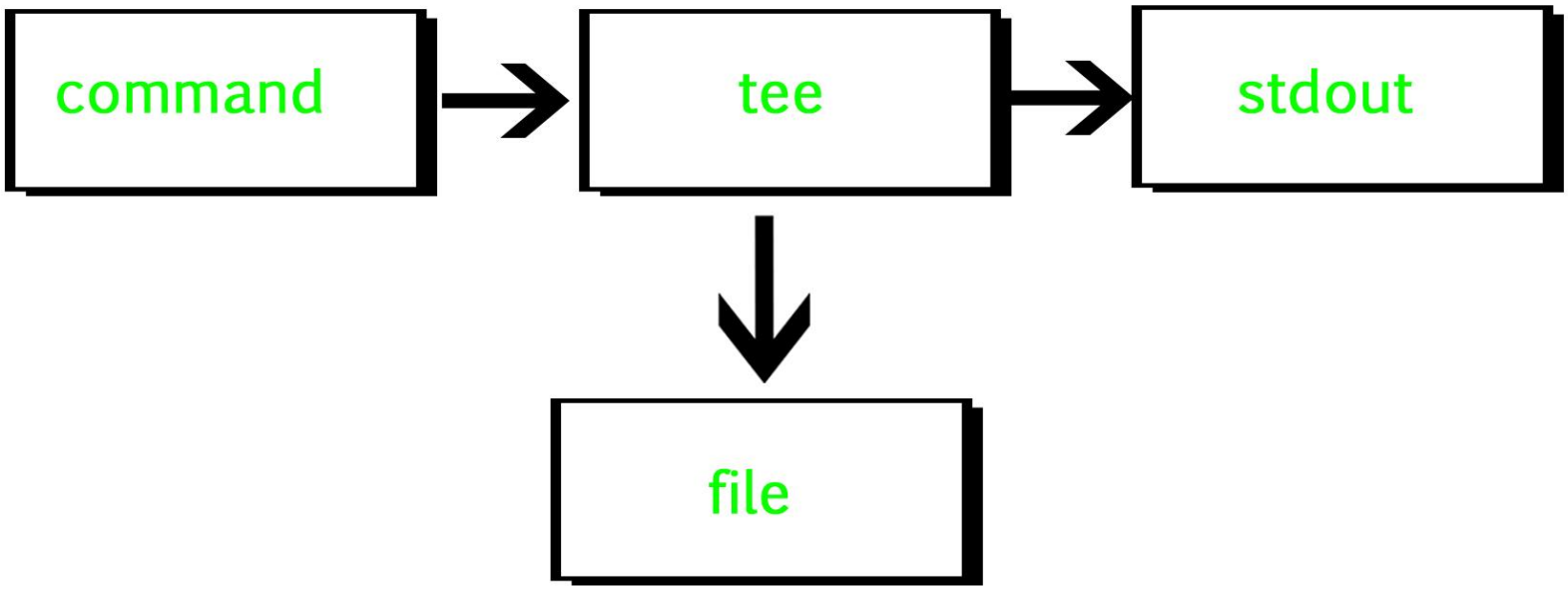
- Display a list of all bash **user accounts** on this computer.

```
$ grep bash /etc/passwd
root:x:0:0:root:/root:/bin/bash
paul:x:1000:1000:paul,,,:/home/paul:/bin/bash
serena:x:1001:1001::/home/serena:/bin/bash
$ grep bash /etc/passwd | cut -d: -f1
root
paul
Serena
```

display the users of /etc/

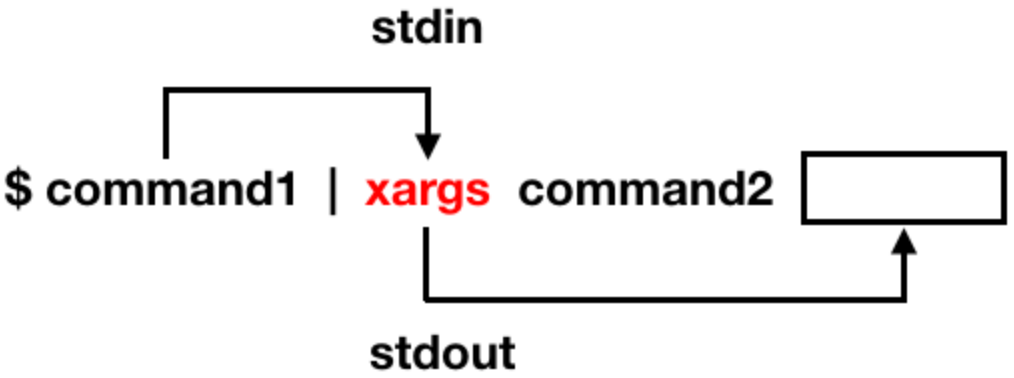
```
cut -d: -f1 /etc/passwd | column
```

Tee



Tee , xarg

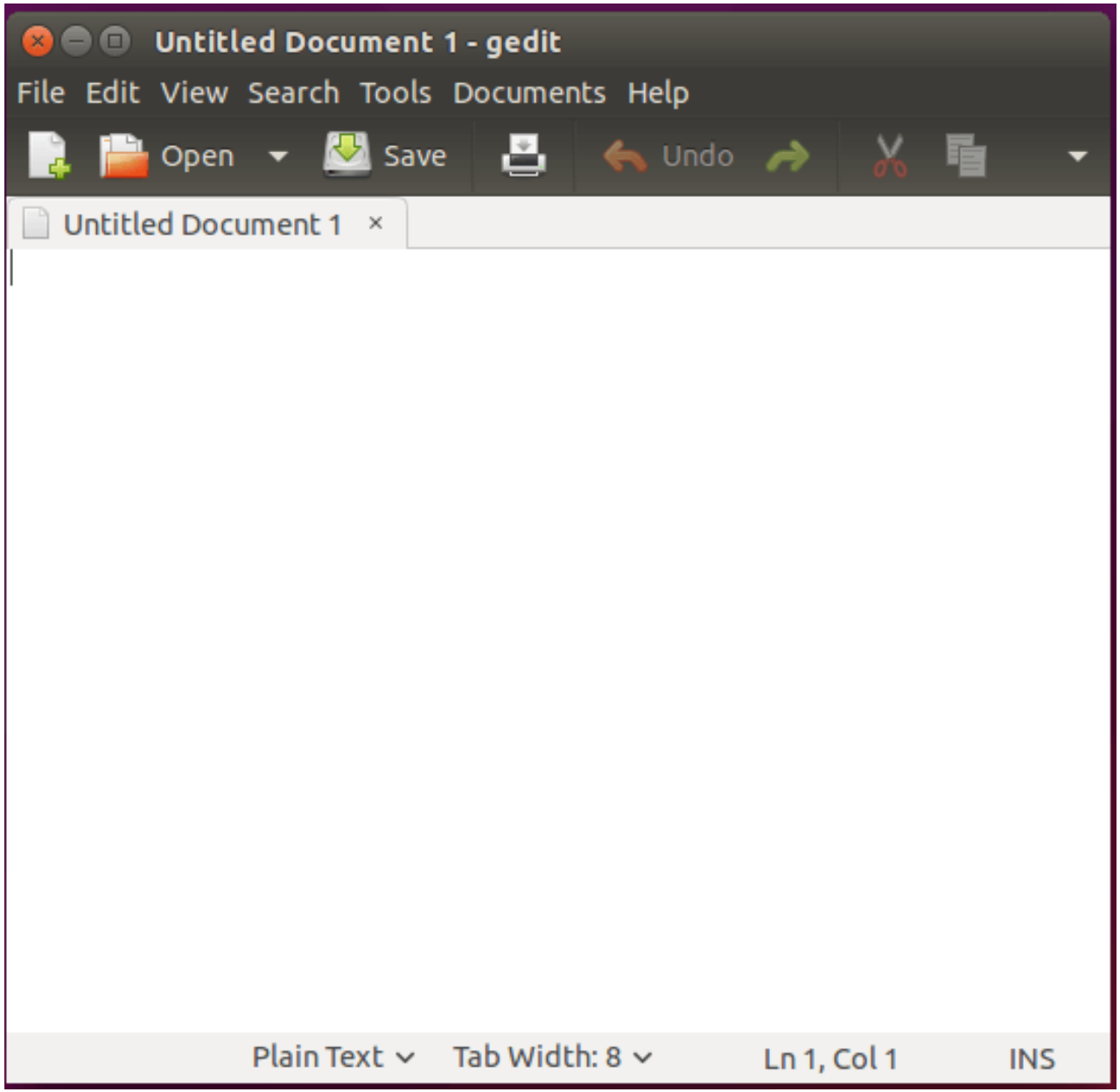
- wc -l file1.txt| tee file2.txt
- echo 'one two three' | xargs mkdir ls one two three



Linux Text Editors

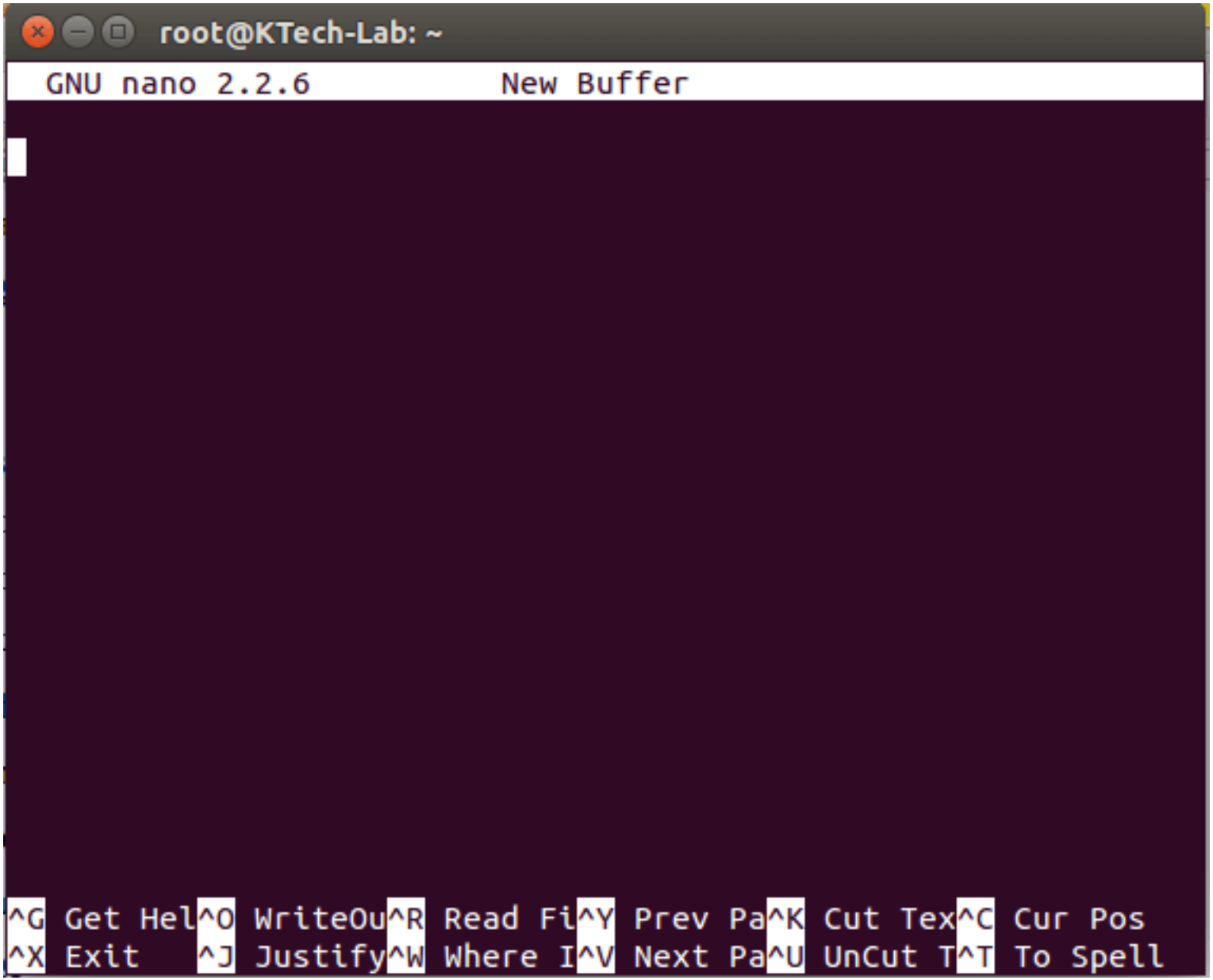
Gedit

- This is a general purpose GUI based text editor and is installed by default text editor on Gnome desktop environment. It is simple to use, highly pluggable and a powerful editor with the following features:
- Support for UTF-8
- Use of configurable font size and colors
- Highly customizable syntax highlighting
- Undo and redo functionalities
- Reverting of files
- Remote editing of files
- Search and replace text
- Clipboard support functionalities and many more


Untitled Document 1 - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
Untitled Document 1
Plain Text
Tab Width: 8
Ln 1, Col 1
INS

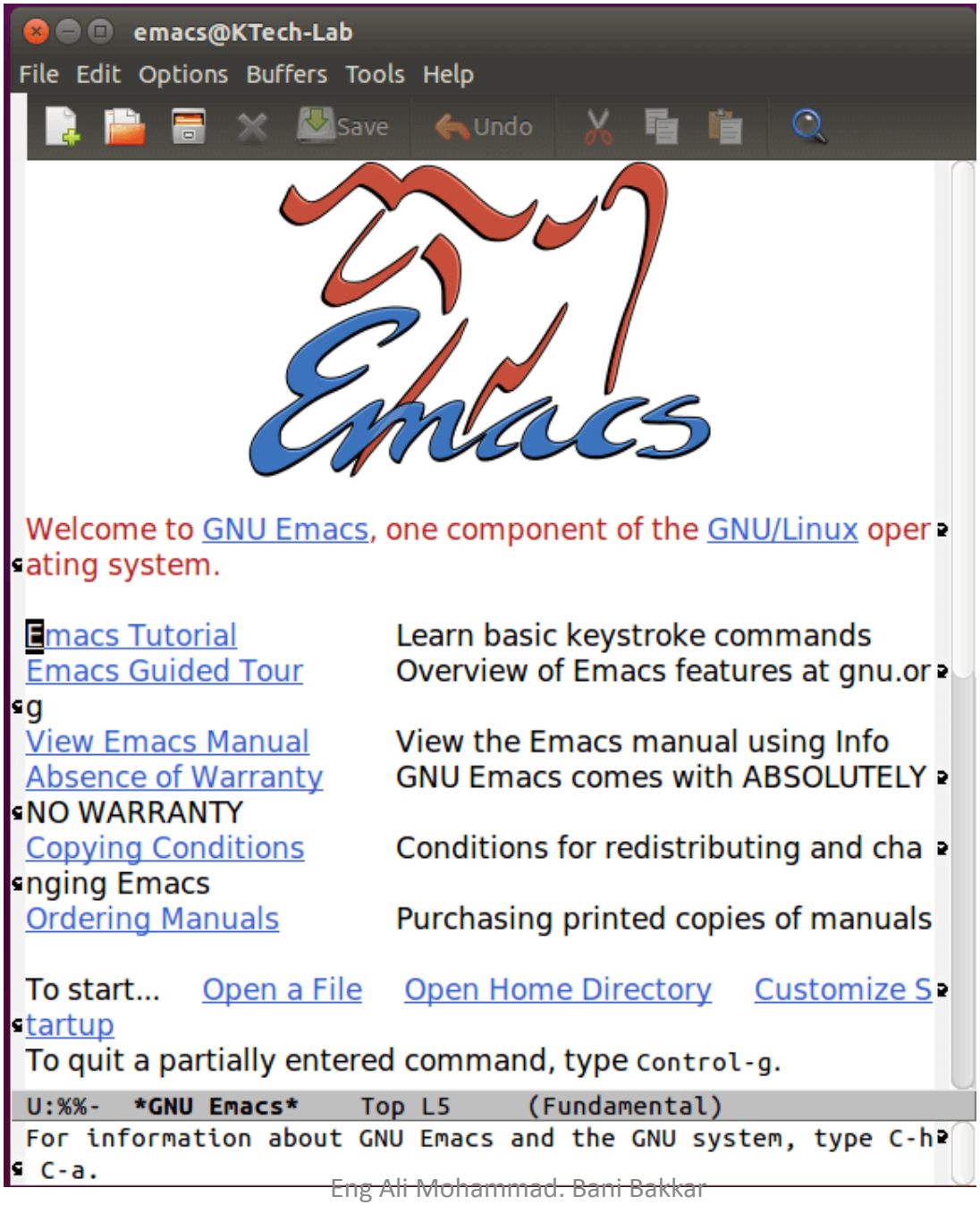
Nano Editor

- Nano is an easy to use text editor especially for both new and advanced Linux users. It enhances usability by providing customizable key binding.
- Nano has the following features:
- Highly customizable key bindings
- Syntax highlighting
- Undo and redo options
- Full line display on the standard output
- Pager support to read form standard input


root@KTech-Lab: ~
GNU nano 2.2.6 New Buffer
^G Get Hel^O WriteOu^R Read Fi^Y Prev Pa^K Cut Tex^C Cur Pos
^X Exit ^J Justify^W Where I^V Next Pa^U UnCut T^T To Spell

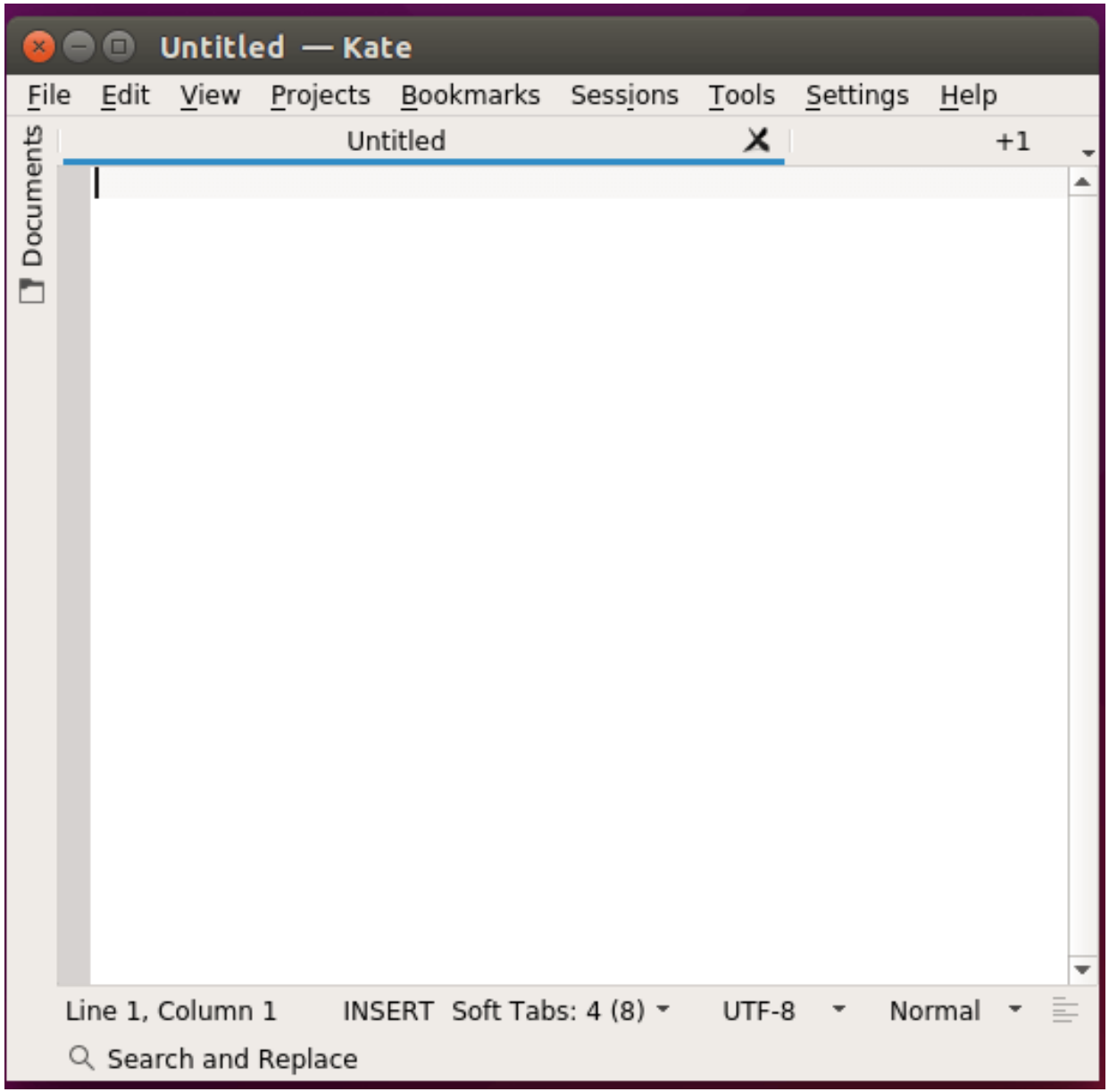
GNU Emacs

- This is a highly extensible and customizable text editor that also offers interpretation of the Lisp programming language at its core. Different extensions can be added to support text editing functionalities.
- Emacs has the following features:
- User documentation and tutorials
- Syntax highlighting using colors even for plain text.
- Unicode supports many natural languages.
- Various extension including mail and news, debugger interface, calender and many more


emacs@KTech-Lab
File Edit Options Buffers Tools Help
Save
Undo
Welcome to GNU Emacs, one component of the GNU/Linux operating system.
Emacs Tutorial
Learn basic keystroke commands
Emacs Guided Tour
Overview of Emacs features at gnu.org
View Emacs Manual
View the Emacs manual using Info
Absence of Warranty
GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions
Conditions for redistributing and changing Emacs
Ordering Manuals
Purchasing printed copies of manuals
To start... Open a File Open Home Directory Customize Startup
To quit a partially entered command, type Control-g.
U:%%- *GNU Emacs* Top L5 (Fundamental)
For information about GNU Emacs and the GNU system, type C-h C-a.

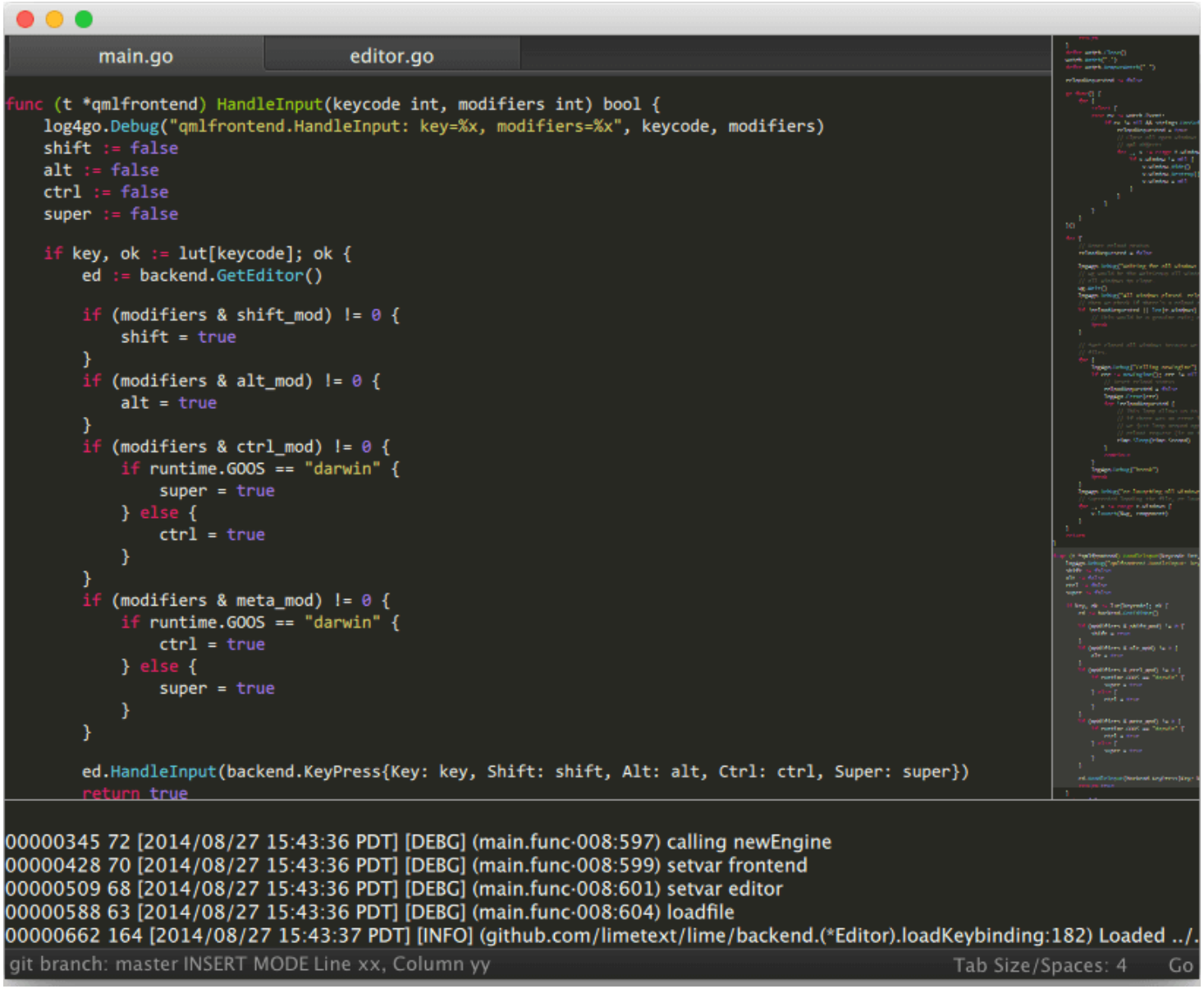
Kate/Kwrite

- Kate is a feature rich and highly pluggable text editor that comes with KDesktop Environment (KDE). The Kate project aims at development of two main products that is: KatePart and Kate.
- KatePart is an advanced text editor component included in many KDE applications which may require users to edit text whereas Kate is an multiple document interface(MDI) text editor.
- The following are some of its general features:
- Extensible through scripting
- Encoding support such as unicode mode
- Text rendering in bi-directional mode
- Line ending support with auto detection functionalities
- Also remote file editing and many other features including advanced editor features, applications features, programming features, text highlighting features, backup features and search and replace features.


Untitled — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
Documents
Untitled
+1
Line 1, Column 1 INSERT Soft Tabs: 4 (8) UTF-8 Normal
Search and Replace

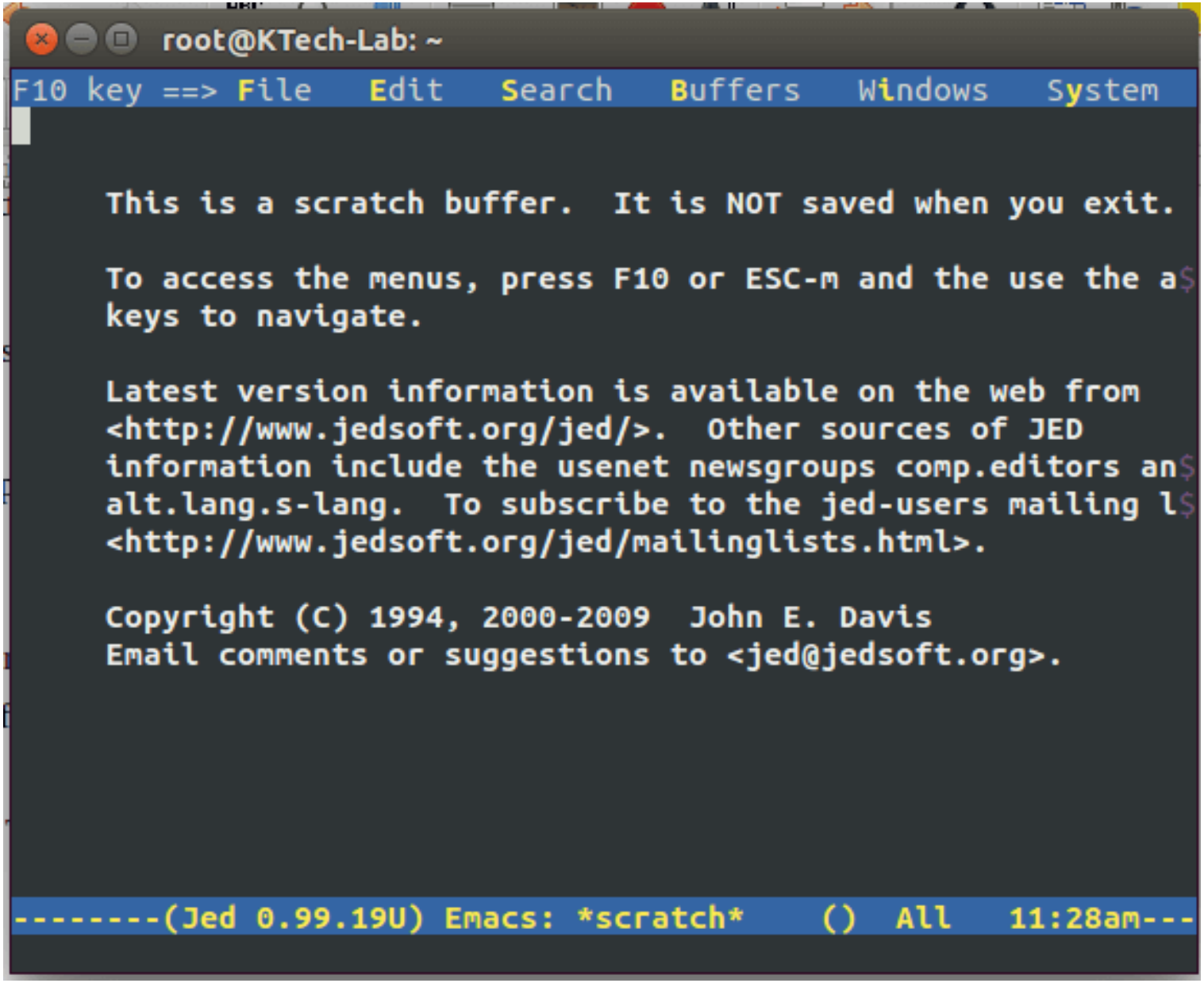
Lime Text

- This is a powerful IDE-like text editor which is free and open-source successor of popular Sublime Text. It has a few frontends such as command-line interface that you can use with the pluggable backend.


main.go
editor.go
func (t *qmlfrontend) HandleInput(keycode int, modifiers int) bool {
log4go.Debug("qmlfrontend.HandleInput: key=%x, modifiers=%x", keycode, modifiers)
shift := false
alt := false
ctrl := false
super := false
if key, ok := lut[keycode]; ok {
ed := backend.GetEditor()
if (modifiers & shift_mod) != 0 {
shift = true
}
if (modifiers & alt_mod) != 0 {
alt = true
}
if (modifiers & ctrl_mod) != 0 {
if runtime.GOOS == "darwin" {
super = true
} else {
ctrl = true
}
}
if (modifiers & meta_mod) != 0 {
if runtime.GOOS == "darwin" {
ctrl = true
} else {
super = true
}
}
ed.HandleInput(backend.KeyPress{Key: key, Shift: shift, Alt: alt, Ctrl: ctrl, Super: super})
return true
00000345 72 [2014/08/27 15:43:36 PDT] [DEBG] (main.func·008:597) calling newEngine
00000428 70 [2014/08/27 15:43:36 PDT] [DEBG] (main.func·008:599) setvar frontend
00000509 68 [2014/08/27 15:43:36 PDT] [DEBG] (main.func·008:601) setvar editor
00000588 63 [2014/08/27 15:43:36 PDT] [DEBG] (main.func·008:604) loadfile
00000662 164 [2014/08/27 15:43:37 PDT] [INFO] (github.com/limetext/lime/backend.(*Editor).loadKeybinding:182) Loaded ../.
git branch: master INSERT MODE Line xx, Column yy
Tab Size/Spaces: 4
Go

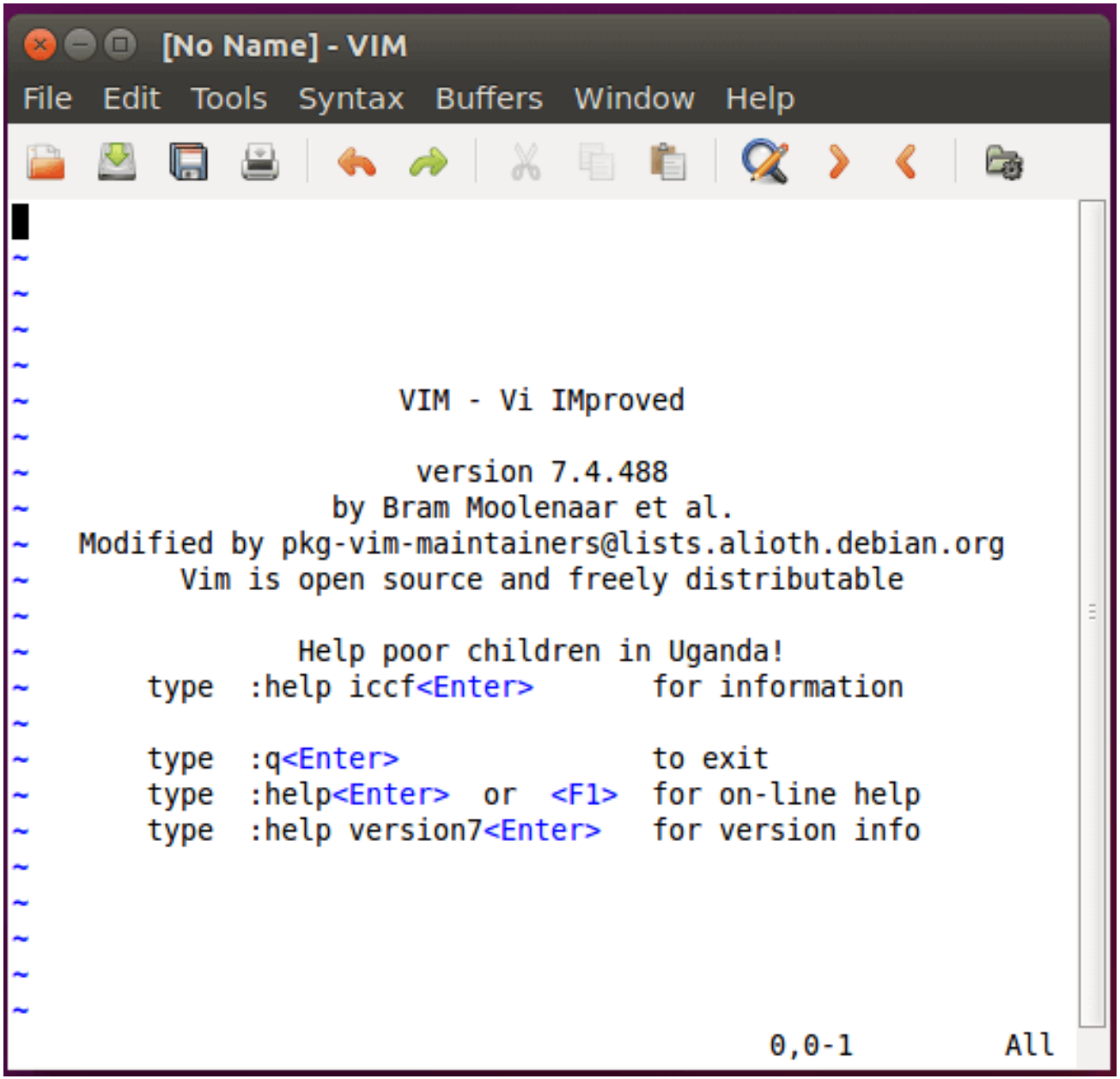
Jed Editor

- This is also another command line editor with support for GUI like features such as dropdown menus. It is developed purposely for software development and one of its important features is support of unicode mode.


root@KTech-Lab: ~
F10 key ==> File Edit Search Buffers Windows System
This is a scratch buffer. It is NOT saved when you exit.
To access the menus, press F10 or ESC-m and the use the a$
keys to navigate.
Latest version information is available on the web from
<http://www.jedsoft.org/jed/>. Other sources of JED
information include the usenet newsgroups comp.editors an$
alt.lang.s-lang. To subscribe to the jed-users mailing l$
<http://www.jedsoft.org/jed/mailinglists.html>.
Copyright (C) 1994, 2000-2009 John E. Davis
Email comments or suggestions to <jed@jedsoft.org>.
--------(Jed 0.99.19U) Emacs: *scratch* () All 11:28am---

gVim Editor

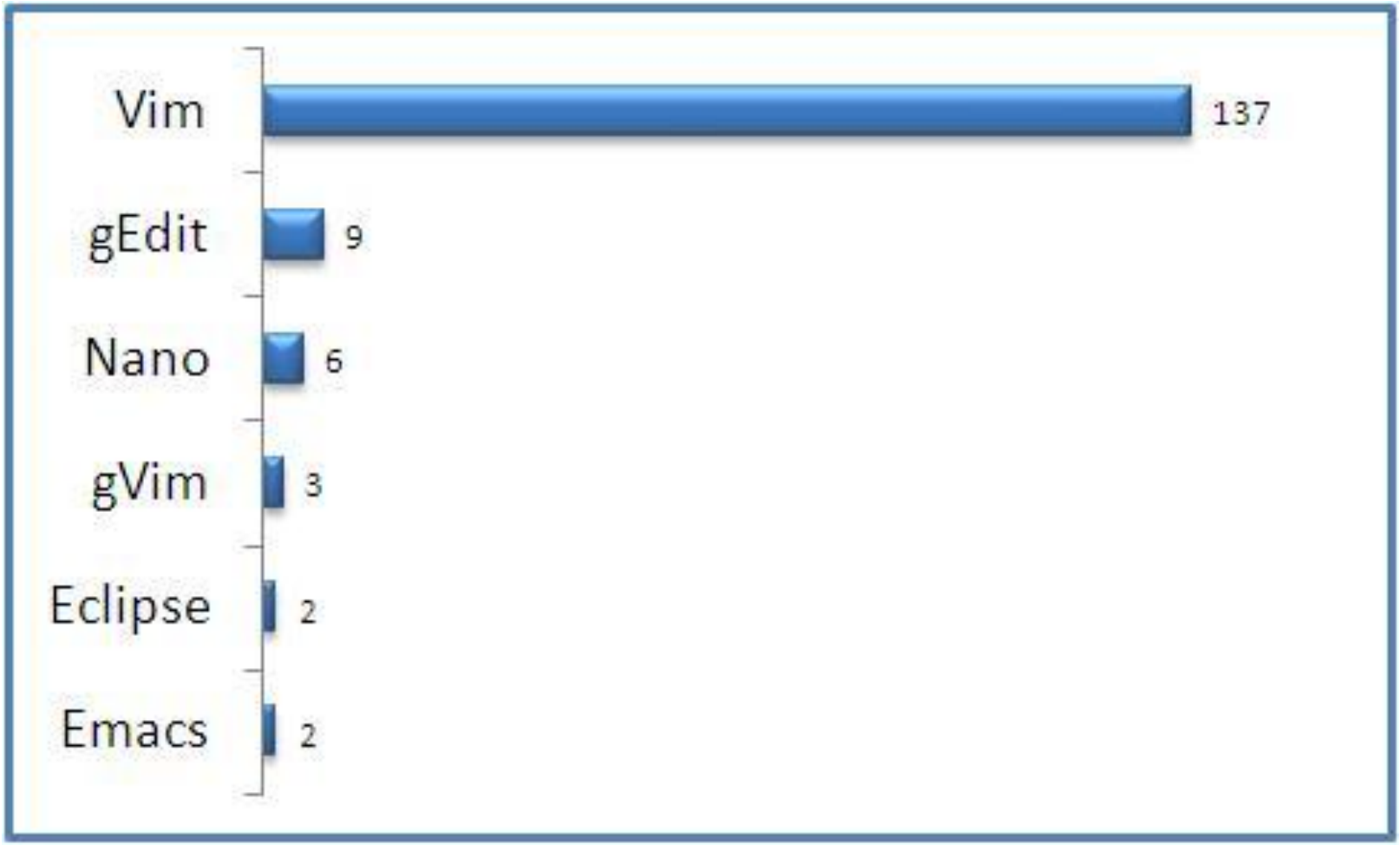
- It is a GUI version of the popular Vim editor and it has similar functionalities as the command line Vim.


[No Name] - VIM
File Edit Tools Syntax Buffers Window Help
VIM - Vi IMproved
version 7.4.488
by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.alioth.debian.org
Vim is open source and freely distributable
Help poor children in Uganda!
type :help iccf<Enter> for information
type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version7<Enter> for version info
0,0-1 All

also

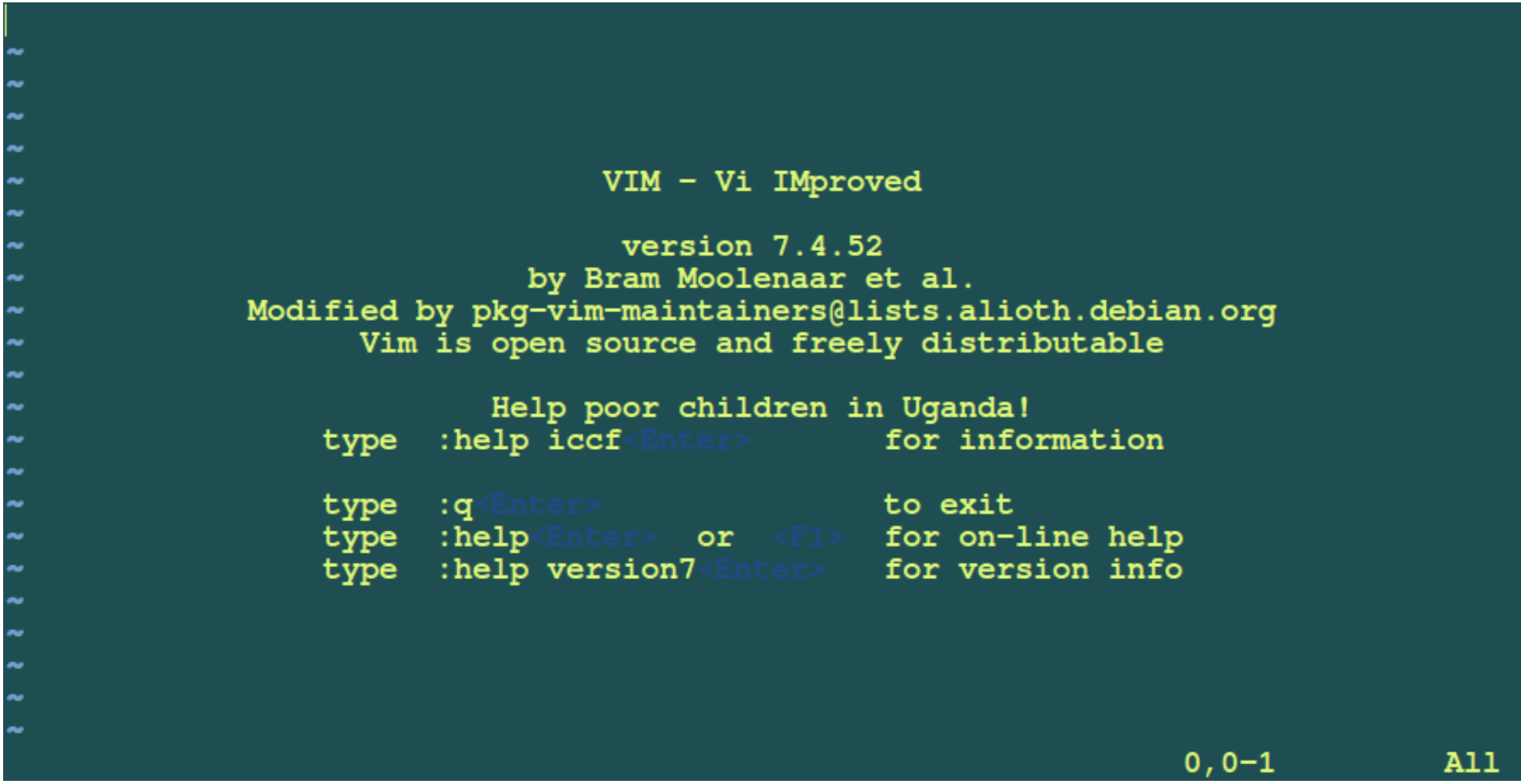
- Geany Editor
- Leaf Pad
- Bluefish
- Atom
- VSCode
- Light Table
- Medit Text Editor
- Neovim – Vim-based Text Editor
- etc

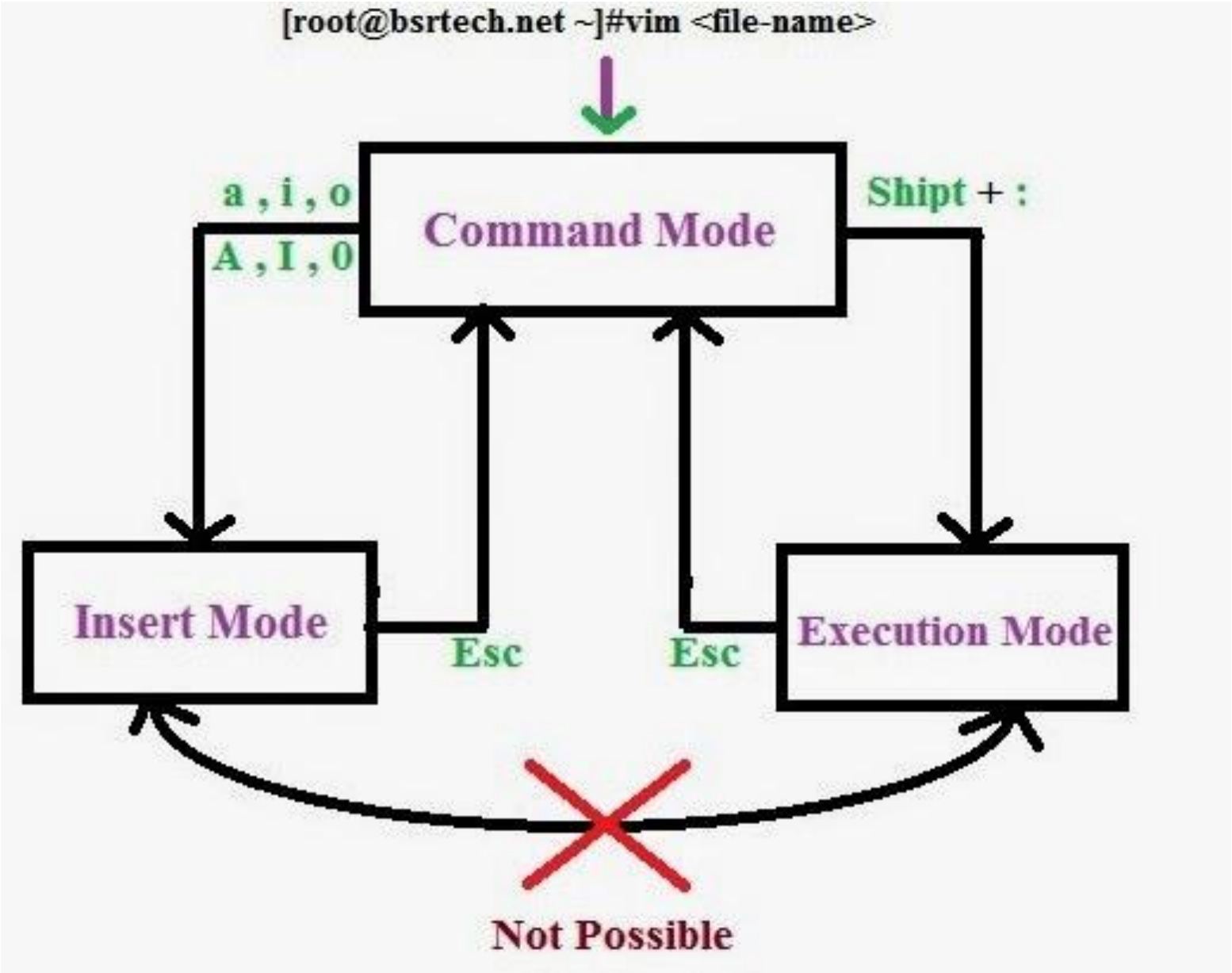
Favorite Linux Text Editor Voting Results



Vim

- Vim is a powerful text editor used in CLI (command line interface).
- Linux uses a lot of configuration files, you'll often need to edit them and vim is a great tool to do so.
- Alternatives to vim are the commandline editors nano and joe.


VIM - Vi IMproved
version 7.4.52
by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.alioth.debian.org
Vim is open source and freely distributable
Help poor children in Uganda!
type :help iccf<Enter> for information
type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version7<Enter> for version info
0,0-1 All


[root@bsrtech.net ~]#vim <file-name>
a , i , o
A , I , 0
Command Mode
Shipt + :
Insert Mode
Esc
Esc
Execution Mode
Not Possible

Vim Command Reference

- save: **:w**
  save and exit: **:wq**
  exit: **:q**
  force: **!** (example **:w! :q!**)
  documents.
  copy: **y**
  copy a line: **yy**
  paste: **p**
  cut: **d**
  cut a line: **dd**
- **Set number**

Vim Command Reference

- **command action**
- x delete the character below the cursor
- X delete the character before the cursor
- r replace the character below the cursor
- p paste after the cursor (here the last deleted character)
- xp switch two characters

Introduction to Users management

User types

- Root user
- Service user
- Normal user

What is the importance for every type?

commands

- **whoami**

The **whoami** command tells you your username.

- **who**

The **who** command will give you information about who is logged on the system.

- **who am i**

With **who am i** the **who** command will display only the line pointing to your current session.

- **w**

The **w** command shows you who is logged on and what they are doing.

- **id**

The **id** command will give you your user id, primary group id, and a list of the groups that you belong to.

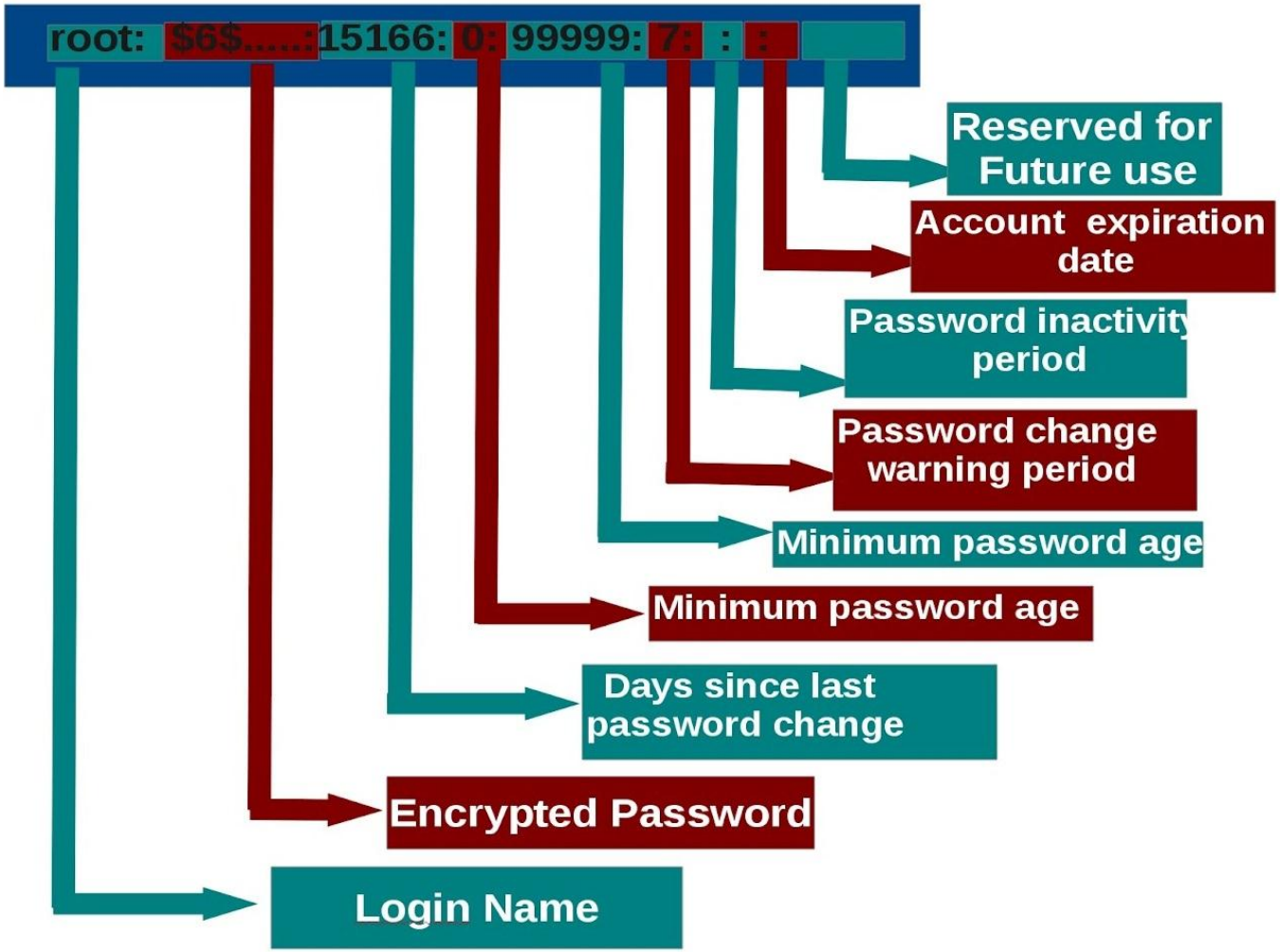
Whereis

/etc/passwd

```
root:x:0:0:root:/root:/bin/bash
 1   2 3 4  5     6       7

 1.root: username
 2.x: password (saved in /etc/shadow in encrypted form)
 3.0: UID (0 is for root)
 4.0: GID (0 is for root)
 5.root: comments
 6./root: Home directory
 7./bin/bash: Login Shell
```

/etc/shadow


root: 6.....:15166: 0: 99999: 7: : :
Reserved for Future use
Account expiration date
Password inactivity period
Password change warning period
Minimum password age
Minimum password age
Days since last password change
Encrypted Password
Login Name

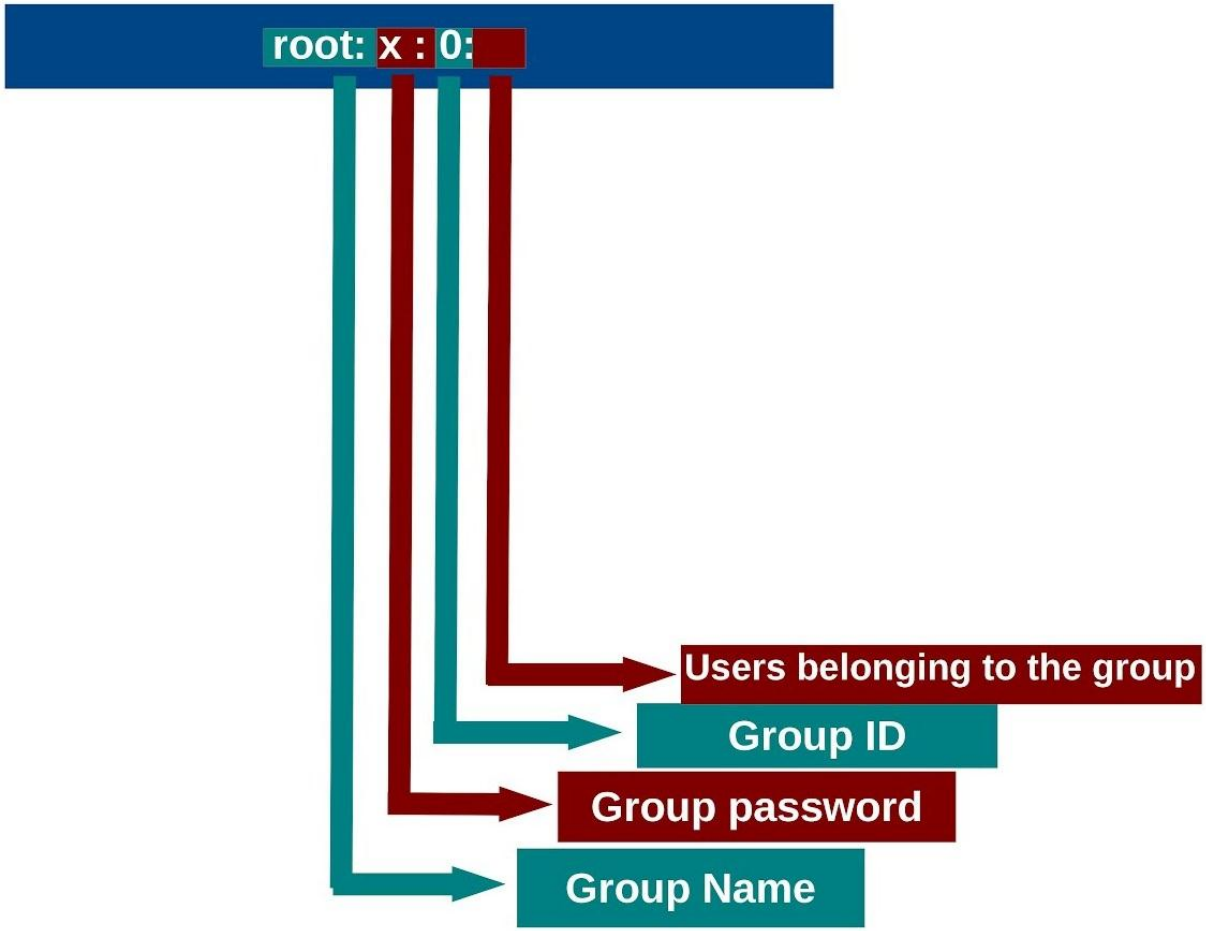
su to another user

- The **su** command allows a user to run a shell as another user.
- **su - $username**
- By default, the **su** command maintains the same shell environment.

Useradd userdel

- Useradd / adduser
- -s : shell
- -p :password
- -u :user id
- -c : gecos
- -e :expiration date
- -md : home directory
- Use usermod to change the attributes of the user after creation
- You can delete the user with **userdel**. The -r option of userdel will also remove the home directory.

/etc/group


root: x : 0:
Users belonging to the group
Group ID
Group password
Group Name

- # useradd ali
- # passwd ali
- # useradd -G admin -u 1005 -s /usr/sbin/nologin ali
- # useradd -c "ahmed " -e 2016-12-31 -s /bin/csh ahmed
- To verify:
- # id
- # id ali
- # id -u ali (UID for ali)
- # tail -n 1 /etc/passwd
- username:password:UID:GID:GECOS:/home/dir:shell
- note:
- # tail -n 1 /etc/shadow
- name:password:1astchange:minage:maxage:warning:inactive:expire:b1ank
- The number 1 indicates an MD5 hash. The number 6 appears when a SHA-512 hash is used.
- # authconfig --passalgo=<descrypt|bigcrypt|md5|sha256|sha512> (to change the hashing type)
- note:
- !! indicates that the user has no password

Modify users:

- #usermod -L ali (lock the user)
- #usermod -U ali (unlock the user)
- #usermod -G sales ali (overwrite secondary group)
- #usermod -aG admin ali (append to secondary group)
- or:
- #vim /etc/group
- To verify:
- #id ali

Delete users:

- #userdel ali
- #userdel -r test (removes home directory)
- Note:
- useradd command assigns new users the first free UID number available in the range starting from UID 1000 or above.
- [root@master ~]#useradd ali
- #ls /home/ -l
- drwx------. 5 ali ali 4096 Jun 21 21:36 ali
- [root@master ~]#userdel ali
- [root@master ~]#useradd ahmed
- #ls /home/ -l
- drwx------. 5 ahmed ahmed 74 Jun 23 05:00 ali
- drwx------. 3 ahmed ahmed 74 Jun 23 02:55 ahmed

Groupadd

- -g :primary group id
- -G : secondary group id
- -aG: add secondary group
- $useradd ali –G sales

- TO ADD A USER TO A SECONDARY GROUP USE:

usermod -a -G examplegroup exampleusername

- UID ranges:
- UID 0 is always assigned to the superuser account, root.
- UID 1-200 is a range of "system users" assigned statically to system processes by Red Hat.
- UID 201-999 is a range of "system users" used by system processes that do not own files on the file system.
- UID 1000+ is the range available for assignment to regular users.

- To change the default:
- #vim /etc/login.defs
- ==========
- Password aging:
- #chage -l ali (list info about the user)
- #chage -E 2017-1-1 ali (expire the user on the specified address)
- #chage -m 1 ali (set minimum number of days before password change)
- #chage -M 120 ali (set maximim number of days before password change)
- #passwd -x 90 ali (the password will expire after 90 days)
- ==========
- Create a default file or directoy in the user's home directory:
- #touch /etc/skel/new_file
- ==========
- GUI tool to manage users and groups:
- #yum install system-config-users
- #system-config-users

- Primary group is the user Private Group (UPG).
- # groupadd sales
- # groupadd -g 1005 admin

- To verify:
- # id
- # id ali
- # grep sales /etc/group
- groupname:password:GID:list,of,users,in,this,group

- # groupmod -g 2000 admin
- # groupadd old
- # groupmod -n new old (rename a group)

run a program as another user

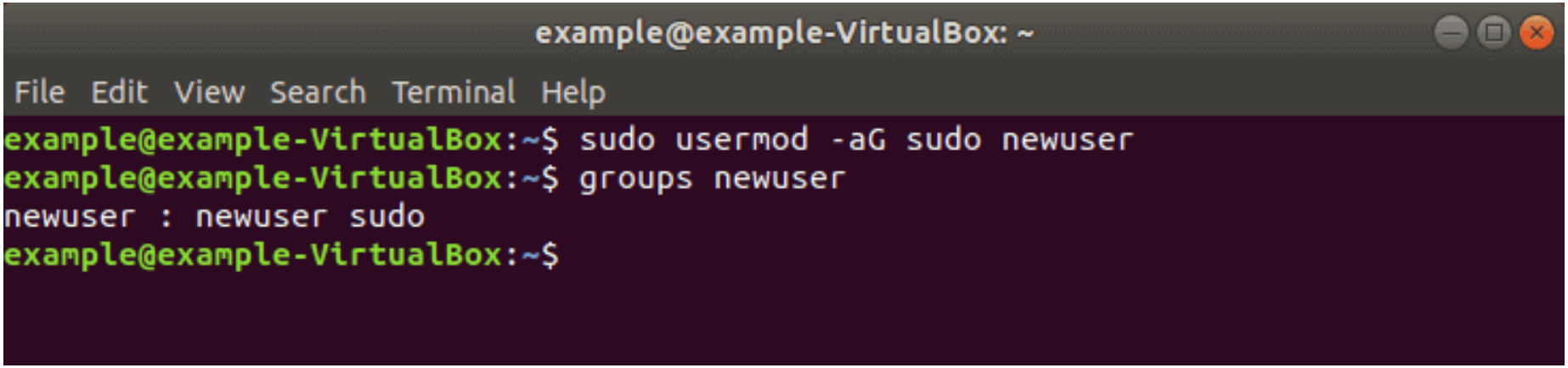
- The sudo program allows a user to start a program with the credentials of another user.

Before this works, the system administrator has to set up the **/etc/sudoers** file. This can be useful to delegate administrative tasks to nother user (without giving the root password).

- Running commands as root with sudo:
- # vim /etc/sudoers
- ali ALL=(ALL) ALL
- %sales ALL=(ALL) ALL

- $ sudo passwd ahmed
- $ sudo passwd -l ahmed
- To verify:
- # tail -f /var/log/secure

```
ubuntu@ubuntu:~$ sudo adduser ayman
Adding user `ayman' ...
Adding new group `ayman' (1000) ...
Adding new user `ayman' (1000) with group `ayman' ...
Creating home directory `/home/ayman' ...
Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for ayman
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
ubuntu@ubuntu:~$ su ayman
Password:
ayman@ubuntu:/home/ubuntu$ sudo adduser rami
[sudo] password for ayman:
ayman is not in the sudoers file.
ayman@ubuntu:/home/ubuntu$
```


example@example-VirtualBox: ~
File Edit View Search Terminal Help
example@example-VirtualBox:~$ sudo usermod -aG sudo newuser
example@example-VirtualBox:~$ groups newuser
newuser : newuser sudo
example@example-VirtualBox:~$

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
jdoe    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
```

/etc/login.defs

- The **/etc/login.defs** file contains some default settings for user passwords like password aging and length settings.

File Security

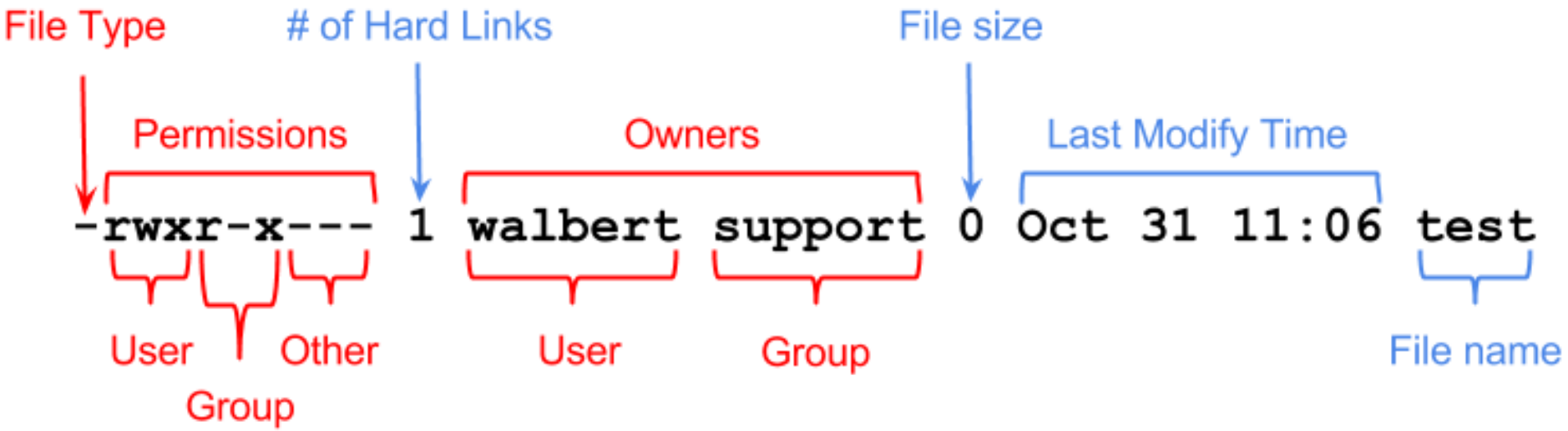
Standard File Permissions

Ownership of Linux files

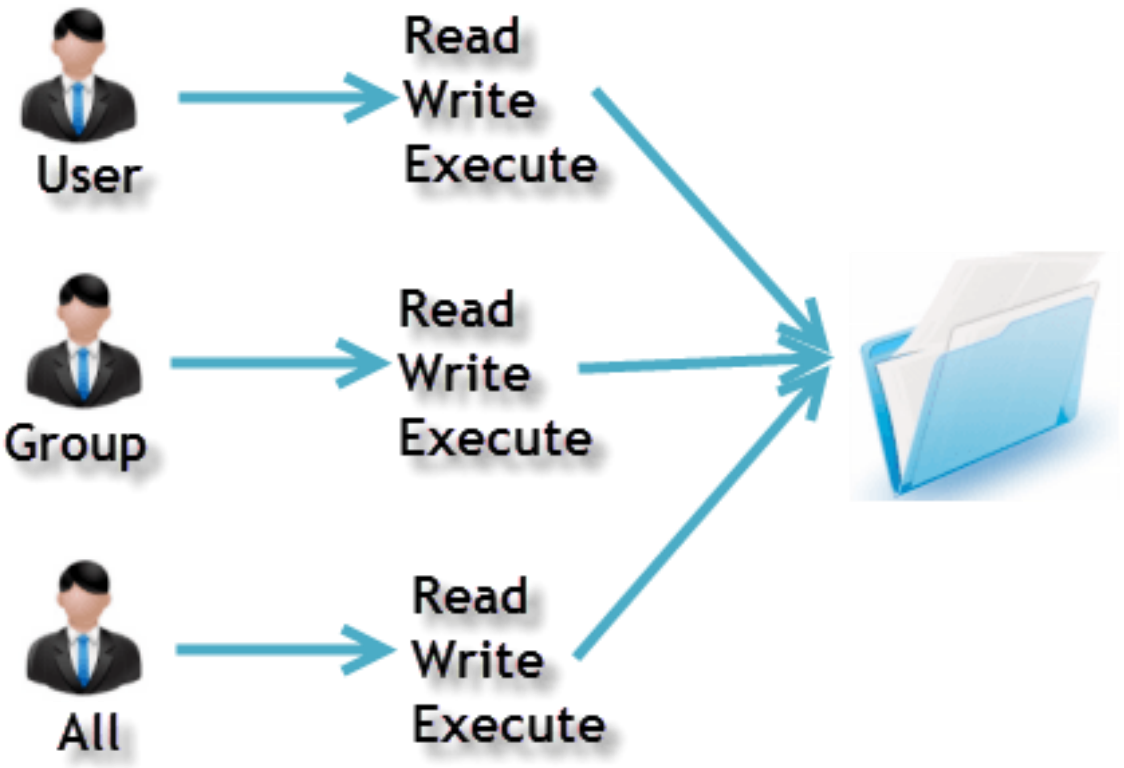
- Every file has a **user owner** and a **group owner**
- Every file and directory on your Unix/Linux system is assigned 3 types of owner.
- **User**
- A user is the owner of the file. By default, the person who created a file becomes its owner. Hence, a user is also sometimes called an owner.
- **Group**
- A user- group can contain multiple users. All users belonging to a group will have the same access permissions to the file.
- **Other**
- Any other user who has access to a file. This person has neither created the file, nor he belongs to a usergroup who could own the file. Practically, it means everybody else.

Permissions

- **Read:** This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.
- **Write:** The write permission gives you the authority to modify the contents of a file. The write permission on a directory gives you the authority to add, remove and rename files stored in the directory.
- **Execute:** you cannot run a program unless the execute permission is set.
- A user must have **execute** access to the **bin** directory in order to execute the **ls** or the **cd** command.
- Use Ls –l /home/ -d


File Type
# of Hard Links
File size
Permissions
Owners
Last Modify Time
-rwxr-x--- 1 walbert support 0 Oct 31 11:06 test
User
Other
Group
User
Group
File name

Owners assigned Permission On Every File and Directory



setting permissions (chmod)

- Permissions can be changed with **chmod**. The first example gives the user owner execute permissions.
- This example removes the group owners read permission.

$ chmod g-r permissions.txt

$ ls -l permissions.txt

- This example removes the others read permission.

$ chmod o-r permissions.txt

- This example gives all of them the write permission.

$ chmod a+w permissions.txt

setting permissions (chmod) cont.

- You don't even have to type the a.

$ chmod +x permissions.txt

- You can also set explicit permissions.

$ chmod u=rw permissions.txt

- make any kind of combination.
- $ chmod u=rw,g=rw,o=r permissions.txt
- Even fishy combinations are accepted by chmod.

$ chmod u=rwx,ug+rw,o=r permissions.txt

setting octal permissions

drwxrwxrwx

d = Directory
r = Read
w = Write
x = Execute

7	rwx	111
6	rw-	110
5	r-x	101
4	r--	100
3	-wx	011
2	-w-	010
1	--x	001
0	---	000

chmod 777

rwx | rwx | rwx

Owner | Group | Others

setting octal permissions cont.

- This makes **777** equal to rwxrwxrwx and by the same logic, 654 mean rw-r-xr-- . The **chmod**
- command will accept these numbers.

$ chmod 777 permissions.txt

$ chmod 664 permissions.txt

$ chmod 750 permissions.txt

umask

- **What is umask?**
  umask, :**U**ser file creation **mask** which is used for determining the default permission for a new file creation. umask command is a shell built-in.

- the default file creation value is 666, the default directory creation value is 777
- Say, the umask value is 022(normal user). 002(root user)
- Assume we create a file say "file1". The permissions given for this file will be the result coming from the substraction of the umask from the default value :
-
  Default: 666
  umask : 022

  Result : 644

  644 is the permission to be given on the file "file1“.

Umask cont.

- **How to find out the umask value?**
  $ umask 0022 The option -S gives in more readable format.
  $ umask -S u=rwx,g=rx,o=rx This means umask, at the max, allows all permissions for the user, read and execute alone for the group and others.
  How to set the umask value?
  $ umask 033
- **How to set this umask permanently for a user?**
  To set this value permanently for a user, it has to be put in the appropriate etc/profile file AND etc/bashrc which depends on the default shell of the user.

  Who can set the umask value?
  It can be set by the root user which will be applicable across the system. Also, a given user can override the umask value by having his own setting in his/her profile file.

chgrp
and chown

- **chgrp**
- You can change the group owner of a file using the **chgrp** command.

chgrp newgroup file

Chown

- You can change the user owner of a file using the **chown** command.

chown newuser File

- You can also use **chown** to change both the user owner and the group owner.

chown groupowner:userowner File

list of special files

- - normal file
- d directory
- l symbolic link
- p named pipe
- b block device
- c character device
- s socket