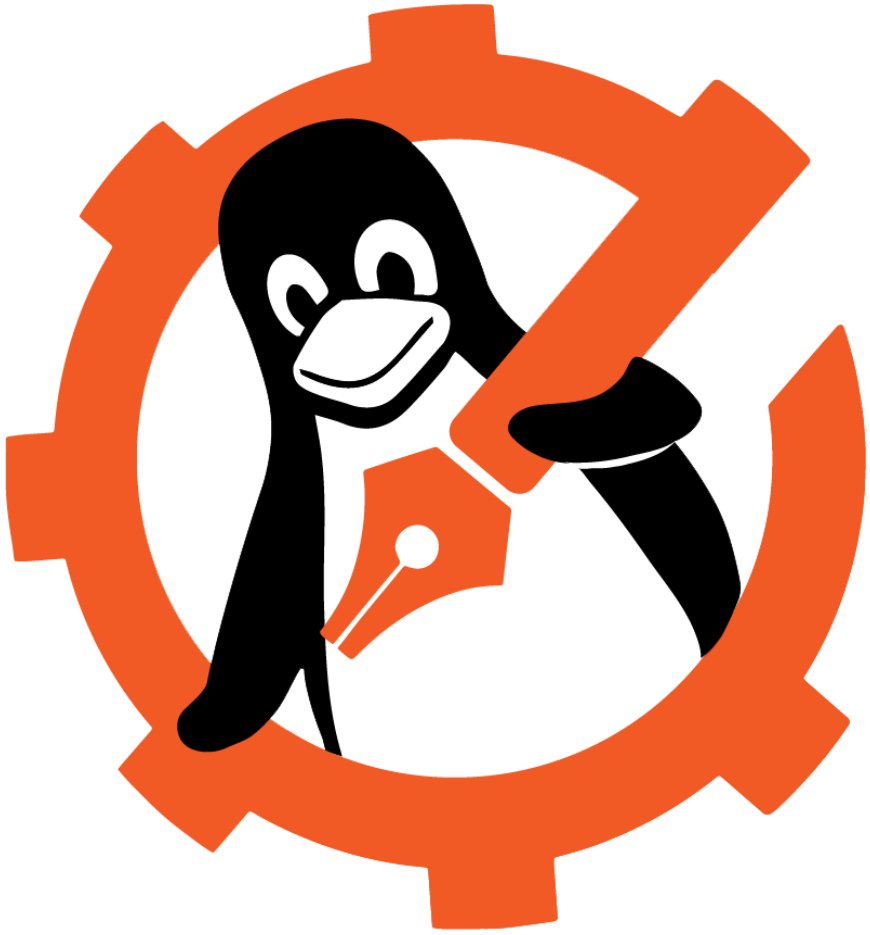


WORK ON THE COMMAND LINE
FILESYSTEM HIERARCHI STANDARD
ENVIRONMENT VARIABLES
MANAGE SHARED LIBRARIES

Presented by

Hosein Mohebbi

Ferdowsi University of Mashhad Linux User Group

t.me/fumlug

WORK ON THE COMMAND LINE

CTRL + ALT + SHIFT

- What is Command Line, Bash and Promt?
  - $ echo
  - $ gedit
- Why CL ?
- What is Root and Root User ?
- Running program as Root vs other Users
  - $ sudo su -
  - $ whoami
- Exiting shell
  - $ exit
  - ctrl + d
  - exec
    - $ exec gedit .bashrc
- Using ctrl+shift+c / ctrl+shift+v to copy/paste on terminal

- $ mkdir # make directories
- $ cd # change directory
  - .
  - ..
  - ~
- $ tree #list contents of directories in a tree-like format
- $ pwd #print name of current/working directory
- $ touch # change file timestamps
- $ ls # list directory contents (the current directory by default)
  - -l use a long listing format
  - -t sort by modification time, newest first
  - -r, --reverse reverse order while sorting
  - -h, --human-readable
  - -a, --all do not ignore entries starting with .
- $ clear # clear the terminal screen
- $ rmdir # remove empty directories
- $ rm # remove files or directories
  - -r, -R, --recursive remove directories and their contents recursively

- Using the tab and arrow keys is useful for your convenience

- $ cp # copy files and directories
  - -i, --interactive prompt before overwrite
  - -f, --force ignore nonexistent files and arguments, never prompt
- $ mv # move (rename) files and directories
- $ cat # concatenate files and print on the standard output
- $ sort # sort lines of text files
  - -r, --reverse reverse the result of comparisons
  - -n reverse the result of comparisons
- Pipe (|)
- Redirection (> , >>)
  - stdin: < stdout: 1> stderr: 2> &>
- $ wc # print newline, word, and byte counts for each file
  - -l, --lines print the newline counts
- $ head # output the first part of files
- $ tail # output the last part of files
  - -n
  - -f output appended data as the file grows
    - $ tail –f /var/log/syslog

- $ uniq # report or omit repeated lines
  - -c, --count prefix lines by the number of occurrences
  - -u, --unique only print unique lines
  - -d, --repeated only print duplicate lines, one for each group

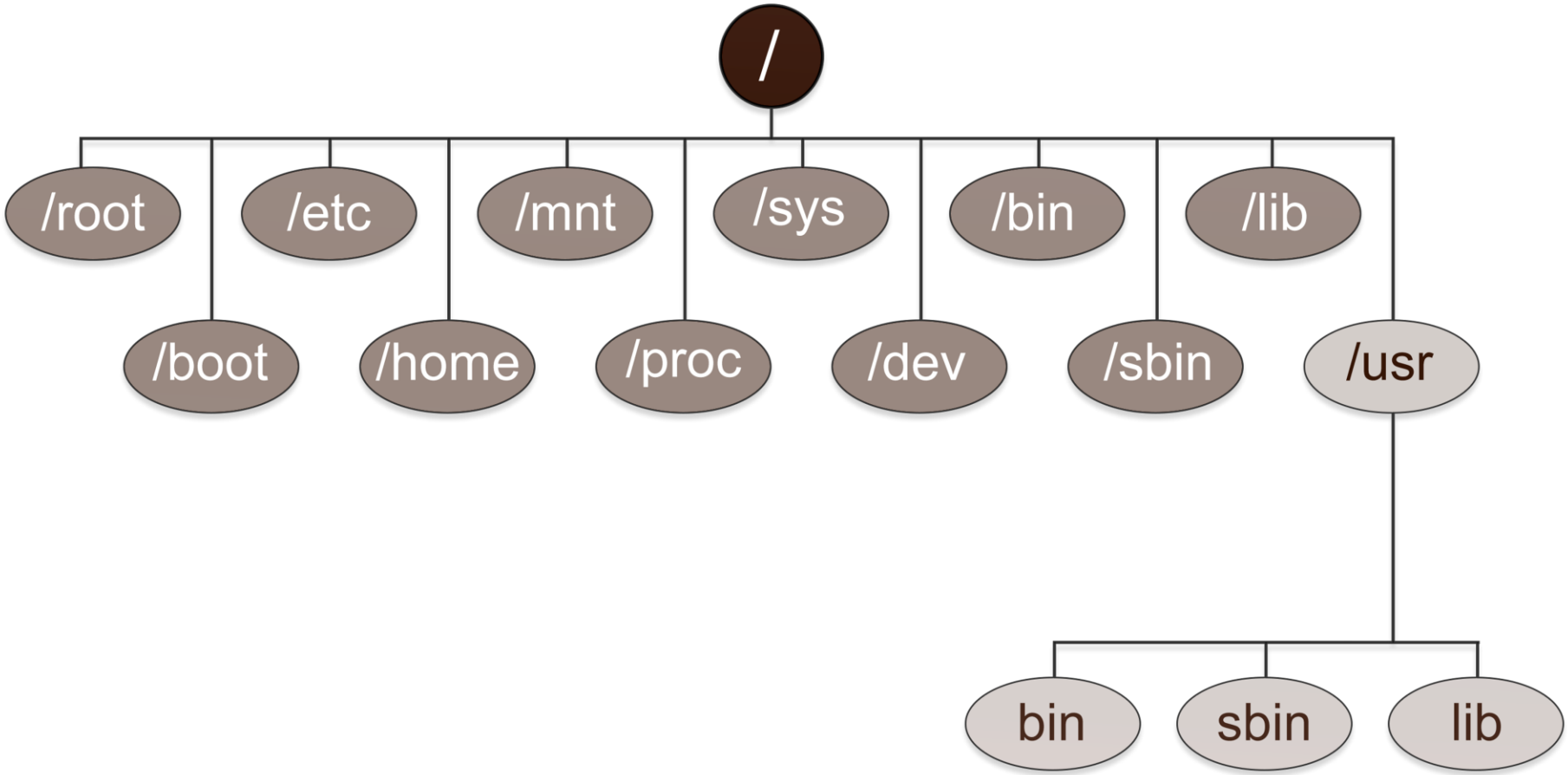
- $ cut # remove sections from each line of files

  - -f, --fields=LIST select only these fields
  - -c, --characters=LIST select only these characters
  - -d, --delimiter=DELIM use DELIM instead of TAB for field delimiter

- $ man # an interface to the on-line reference manuals
  - -f simple search
  - -k global search

- Control operators
  - ; || && | &

FILESYSTEM HIERARCHI STANDARD (FHS)


/
/root
/etc
/mnt
/sys
/bin
/lib
/boot
/home
/proc
/dev
/sbin
/usr
bin
sbin
lib

LOOKING AT FILE SYSTEMS

- Find out Hardware Abstractioon Layer (HAL)
- ls /dev
- $ lsusb
- $ lshw
- $ lsmod
  - $ rmmod
  - $ modprobe
  - ls /etc/modprobe.d/
- $ blkid #UUID
- cat /var/log/syslog

HAVE A SHORT BREAK :)

ENVIRONMENT VARIABLES

Name	Function
USER	The name of the logged-in user
UID	The numeric user id of the logged-in user
HOME	The user's home directory
PWD	The current working directory
SHELL	The name of the shell
?	The exit code of the last command
HISTFILE	Next slide
HISTSIZE	Next slide
PATH	Next slide

- Using $ to print value of envs

- $ env
- $ export
  - $ cat .bashrc
- $ history
  - $ history 10
  - $ cat .bash_history
- $ ctrl + r

- $ echo $PATH
- $ which
- $ whereis
- $ type

RUNNING YOUR COMMAND

- Give the full path
- Give the relative path (. , ..)
- It is possible to add to our PATH
  - PATH=$PATH:your path

EXERCISE

Write your command that print your name on the terminal :

$ printme

MANAGE SHARED LIBRARIES

STATIC VS DYNAMIC

- What is library ?
- **Static** linking is when you add this library to your executable program. In this method your program size is big because it has all the needed libraries. One good advantage is your program can be run without being dependent to other programs / libraries.
- **Dynamic** linking is when you just say in your program "We need this and that library to run this program". This way your program is smaller but you need to install those libraries separately. This makes programs more secure (because libraries can be updated centrally), more advanced (any improvement in a library will improve the whole program) and smaller.

- Where are main libraries ?
  - $ ls /lib
  - $ ls /usr/lib
  - $ ls /lib64

- $ ldd #print shared object dependencies
  - $ ldd /bin/ls
  - $ ldd /sbin/ldconfig

- LD_LIBRARY_PATH #Set priority in reading Libraries
  - $ export LD_LIBRARY_PATH=path1[:path2]

HAVE A BREAK :)