Free Code Camp Course - Linux Essentials for Hackers - 4 hours

- linux is imoprtant for
  - security
  - system administration
  - personal use! (I want this :))
- linux can run on
  - virtual machine
  - baremetal

useful keyboard shortcuts

- open terminal
  - `ctrl + alt + t`
- move window to top, right, bottom, left
  - `window + arrowkeys`
- increase/decrease font size
  - `ctrl + shift + "+"`
  - `ctrl + "-"`
- clear the terminal
  - `ctrl + l`
  - `clear` command
- end the current process
  - `ctrl + c`
- see history commands
  - `top and bottom in arrowkeys`
- auto complete command
  - `tab`
  - `right-arrowkey`
- close the window
  - `ctrl + w`
  - `ctrl + shift + w`

file management and manipulation

- print working directory

  ```
  pwd
  ```

- list directory

  ```
  ls
  ```

- list direcotry in a table

  ```
  ls -l
  ```

- list directory also hiddens

```
ls -a
```

- list direcotry in a table human readable

```
ls -lh
```

- list direcotry recursively!

```
ls -R
```

- cahnge directory to home directory

```
cd
```

```
cd ~
```

- change directory to previous directory

```
cd -
```

- change directory to parent directory

```
cd ..
```

- change directory to any directory you want

```
cd wanteddirectorypath
```

- change directory to root directory

```
cd /
```

- see the one line documentation of a command!!!

```
whatis thecommand
```

- see the complete documentation of a command

```
man thecommand
```

- usual way to see the command(app) documentation

```
thecommand -h
```

```
thecommand --help
```

```
thecommand -help
```

- create new file

```
touch newfilename
```

- returning a line

```
echo yourlinegoeshere
```

- redirect output to a file

```
echo "something" > somefile
```

- see the content of a file

`cat somefile`

- copy content of a file to another file

  `cat somefile > anotherfile`

- remove a file or directory

  `rm yourfile`

  `rm -r yourdirectory`

- remove all files and folders in a directory

  `rm -r *`

- create a directory

  `mkdir yournewdirectoryname`

- copy a file to another directory

  `cp yourfilepath yournewdirectory`

- copy a directory to another directory

  `cp -r yourfolderpath yournewdirectory`

- move a file or folder to new directory

  `mv yourfileorfolderpath yournewdirectory`

- rename a file

  `mv yourfileorfolderpath yournewname`

- remove a direcotry

  `rmdir yourdirectorypath`

- open a file with nano editor

  `nano filepath`

- open a file with vim editor

  `vi filepath`

file and directory permissions

- in a file config: `drwxrwxrwx`
  - `d` shows directory, `-` shows file
  - `r` shows read permission
  - `w` shows write permission
  - `x` shows execute permission
  - `-` shows without permission
  - first `rwx` is for owner of file

  - second `rwx` is for group of file
  - third `rwx` is for others of file
- change the permission of file

  ```
  chmod mapuserstopermissions filepath
  ```
- map users to permissions
  - `ugo`: owner, group, others
  - `rwx`: read, write, execute
  - `=+-`: equal, append, delete
- also you can use from binary to map users to permissions
- change the permission of folder

  ```
  chmod -R mapuserstopermissions folderpath
  ```

file and directory ownership

- every file has a user and also a group
- change the owner of file

  ```
  chown newowner filepath
  ```
- change the group of file

  ```
  chgrp newgroup filepath
  ```
- see the groups

  ```
  groups
  ```
- see the groups of specific user

  ```
  groups username
  ```
- see the current logined users

  ```
  users
  ```

grep and piping

```
$ whatis grep
grep (1)             - print lines that match patterns
```

- usually `grep` used in 2 ways
  - direct
  - pipe the last command

direct `grep`

- searching in file

  `grep "word" filepath`

  `-i` usually used for case-insensitive way

`grep` with pipe

- simple use

  `yourfirstcommandwithoutput | grep "wordsearchinginoutput"`

finding files with `locate`

```
$ whatis locate
locate (1)           - find files by name, quickly
```

- actually `locate` is not very practical command in my opinion...

enumerating distribution and kernel information

- current user

`whoami`

- os(workstation) name

  `hostname`

- change `hostname`

  `sudo nano /etc/hostname`

- see the linux distribution

  `lsb_release -a`

  `cat /etc/issue`

  `cat /etc/os-release`

  `cat /etc/*release`

  `uname -a`

- cpu information

  `lscpu`

- pci information (see the graphic card)

  `lspci`

find and bandit challanges

```
$ whatis find
find (1)             - search for files(also directories) in a directory hierarchy
```

- very powerful command!
- usage

  `find pathtosearch specifiers`

- specifiers
  - `-type`
  - `-name`
  - `-iname`
  - `-size`
  - `-perm`

  you can see my `bandit` challange solutions in linux reference root folder `challange-bandit`

`xargs`

- get the output as an input for a command!!!

```
$ ls
another.txt  myfile.txt  README.md
$ cat another.txt myfile.txt README.md
inside text wrote file
inside text wrote file
inside markdown text
$ ls | grep -E "*.txt" | xargs cat
inside text wrote file
inside text wrote file
```

`file` command

```
$ whatis file
file (1)             - determine file type
```

```
file *
another.txt:      ASCII text
myfile.txt:       ASCII text
README.md:        ASCII text
```

shell

- the terminal interface that you works with it
- see the default shell

```
echo $SHELL
/usr/bin/zsh
```

- see the all available shells

```
cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/usr/bin/sh
/bin/dash
/usr/bin/dash
/bin/zsh
/usr/bin/zsh
```

bash and zsh

- bash config files

```
ls -a | grep .bash
.bash_history
.bash_logout
.bash_profile
.bashrc
```

- zsh config files

```
ls -a | grep .zsh
.oh-my-zsh
.shell.pre-oh-my-zsh
.zshenv
.zsh_history
.zshrc
```

- `history` command

```
history (3readline)  - GNU History Library
```

space usage

```
whatis du
du (1)               - estimate file space usage
```

du runs on current path recursively!

- very practical example usage

```
du -sh * | sort -rh
```

`sort -h` means human readable sort! (1G > 23K)

disk usage

```
whatis df
df (1)               - report file system disk space usage
```

not very practical!

tar

```
whatis tar
tar (1)              - an archiving utility
```

- archiving

  `tar -cvf tarnewfilename.tar folderpath`

  `tar -czvf tarnewfilename.tar.gz folderpath`

- unarchiving

  `tar -xvf archivedfile.tar`

  `tar -xzvf archivedfile.tar.gz`

users and groups in more detail

- how to change `sudoers` file

  `sudo nano /etc/sudoers`

  above command does not work!, use from below command!

`sudo visudo`

- create new user

  `sudo useradd yourusername`

- user add with advanced options

  `useradd -m -c "Akbar Akbari" -s /bin/bash akbar`

  `-m` makes a specific folder for user in `home`!

- create new group

  `sudo groupadd yourgroupname`

- see the groups of specific user

  `groups username`

- map a user to a group

```
usermod -aG groupname username
```

- remove a user

  ```
  userdel yourusername
  ```

  ```
  sudo userdel -r alexis
  ```

- important file about users and groups

  ```
  cat /etc/passwd
  ```

- see the all groups on system

  ```
  cat /etc/group
  ```

- `visudo` config format

  ```
  yourusername  <hosts>=(<users>:<groups>) <accesse_commands>
  ```

- also same for groups, but

  ```
  %yourgroupname ...
  ```