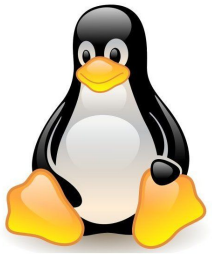
Command Line Basics 2/2



Linux Essentials
Session-3

CLARUSWAY
WAY TO REINVENT YOURSELF

Essential to OS Management

File Management	User Management	Performance Management	Getting Help
• File structure • Create, delete, modify • Copy, move • Home directories • Special directories • Navigating directories • Searching (files, dirs) • Editing files	• Add, delete, modify • Password management • File permissions • Application permissions	• System troubleshooting • System performance • Log file access • Tracking usage • Network troubleshooting • Network performance	• Command line help • Command reference

Understanding OS: Important for DevOps

- Responsibility for DevOps Engineers:
  - Automate
  - Configure OS, network, software → entire ecosystem
- To automate:
  - Must understand OS tools available for this
- To configure systems:
  - Must understand them, including OS

Recap: Linux Overview

- Linux is widely used in industry
  - Has been available for 35+ years
  - Is open source
  - Secure & robust
- 100+ Distributions
  - Debian / Ubuntu
  - Fedora / CentOS / Amazon Linux 2
  - RedHat
- Many companies migrating to AWS have significant Linux footprint

Recap: What is the SHELL?

- **Primary interface** on Linux systems to process commands
- Bash (**B**ourne **A**gain **SH**ell) is the SHELL on most Linux systems

01	Dual Purpose	• Command **interpreter** • **Programming** language
02	Prompt	• Usually includes **user**, **host** & **directory** • Can **customize** it
03	User info in Prompt	• **$** sign at the end signifies non-root user • **#** sign at the end signifies root user
04	Execution	• Sent to interpreter for **parsing & execution**

Recap: Command Line Basics (1)

```
ls      lists directory contents
        ls -l, ls -la, ll

cd      change current directory
        cd [dir], cd .., cd /, cd ~
        note: home directory is ~

mkdir   create a new directory
        mkdir [dir]

rmdir   delete an empty directory
        rmdir [dir]

pwd     show the current path
```

Recap: Command Line Basics (2)

touch create an empty file
touch [filename]

rm delete a file (or directory)
rm [file], rm -f [file], rm -rf [non-empty dir]

cp copy a file from one location to another
cp [source] [dest]

mv move a file from one location to another
mv [source] [dest]

cat show the contents of a file
cat [file]

Recap: Command Line Basics (3)

echo show a message on the screen
echo Hello

>, >> redirect the output
echo hello > file
echo Hello >> file, [appends to file]

Also:
- sudo su switches you to the root (super) user
- you can use **absolute** or **relative** paths
- don't forget **pushd** and **popd** commands
- use the "tab" key for autocomplete
- commands and file names are case sensitive

Exercise

```
ls
cd
mkdir
rmdir
pwd
touch
rm
cp
mv
cat
echo
>
>>
```

1. list contents of current directory in detailed/long format
2. write “wow” into a file called file.txt, which doesn't already exist
3. show which directory you are currently in
4. you are in your home directory; delete the file call /var/log/httpd/httpd.log
5. you are in the root directory, there is no /var/www; create the directory /var/www/html
6. show the contents of the file /var/log/cloud-init.log from your home directory
7. copy the file /etc/timezone to the directory you are currently in
8. rename the file /etc/sudoers to /etc/sudoers.bak

CLARUSWAY
WAY TO REINVENT YOURSELF

Exercise

```
ls
cd
mkdir
rmdir
pwd
touch
rm
cp
mv
cat
echo
>
>>
```

1. list contents of current directory in detailed/long format
   `ls -l`
2. write “wow” into a file called file.txt, which doesn't already exist
   `echo wow > file.txt`
3. show which directory you are currently in
   `pwd`
4. you are in your home directory; delete the file call /var/log/httpd/httpd.log
   `rm /var/log/httpd/httpd.log`
5. you are in the root directory, there is no /var/www; create the directory /var/www/html
   `mkdir /var/www`
   `mkdir /var/www/html`
6. show the contents of the file /var/log/cloud-init.log from your home directory
   `cat /var/log/cloud-init.log`
7. copy the file /etc/timezone to the directory you are currently in
   `cp /etc/timezone timezone or`
   `cp /etc/timezone .`
8. rename the file /etc/sudoers to /etc/sudoers.bak
   `mv /etc/sudoers /etc/sudoers.bak`

CLARUSWAY
WAY TO REINVENT YOURSELF

Kahoot!

CLARUSWAY
WAY TO REINVENT YOURSELF

Break
return @ 8:02pm

amazon web services

Hidden Files and Directories

Any file or directory starts with period (.)
.abc

```
root@DESKTOP-4QQ1S5L:~# ls
root@DESKTOP-4QQ1S5L:~# ls -a
.  ..  .bash_history  .bashrc  .profile  .viminfo
root@DESKTOP-4QQ1S5L:~# touch .file5
root@DESKTOP-4QQ1S5L:~# ls -a
.  ..  .bash_history  .bashrc  .file5  .profile  .viminfo
root@DESKTOP-4QQ1S5L:~#
```

How to Hide Files And Directories in Linux

Basic Shell Commands

head [-x] <file> show first 10 (or x) lines from file contents
tail [-x] <file> show last 10 lines from file contents
special case:
tail -f <file>

```
robert@robert-virtual-machine:~$ head a.txt
You do not do, you do not do
Any more, black shoe
In which I have lived like a foot
For thirty years, poor and white,
Barely daring to breathe or Achoo.

Daddy, I have had to kill you.
You died before I had time—
Marble-heavy, a bag full of God,
Ghastly statue with one gray toe
```

```
robert@robert-virtual-machine:~$ tail a.txt
Where it pours bean green over blue
In the waters off beautiful Nauset.
I used to pray to recover you.
Ach, du.

In the German tongue, in the Polish town
Scraped flat by the roller
Of wars, wars, wars.
But the name of the town is common.
My Polack friend
```

Simple Globbing

Globbing is the process of matching patterns in filenames or text by using a wildcard characters to create a pattern.

Character	Name	Function
?	Question mark	Match any single character
*	Asterisk	Match any number of character(s)
[]	Brackets	Match character from a range
^	Caret	Used to match starting character
$	Dollar sign	Used to match ending character
{}	Curly brace	Used to match more than one pattern
|	Pipe	Used for applying more than one condition

Simple Globbing - Examples

```
* - any characters
? - single character
[] - range of characters
^ - starts with
$ - ends with
```

- Use "*" and ? with ls directly
  - `ls b*.*`
  - `ls file?.txt`
- Use [], ^ and $ with grep utility
  - `ls | grep ^b`
  - `ls | grep s$`
  - `ls | grep ^[a-m]`

Simple Globbing with Grep

Character	Name	Function
[]	Brackets	Match character from a range
^	Caret	Used to match starting character
$	Dollar sign	Used to match ending character
|	Pipe	Used for applying more than one condition

list.txt

```
Apple
4000
Banana
700
Orange
850
Pear
9000
Jackfruit
```

```
grep '^[P-R]' list.txt
grep '^[A-C]' list.txt
grep a$ list.txt
grep 50$ list.txt
grep '^[A-C]' list.txt | grep '^[B]'
```

CLARUSWAY
WAY TO REINVENT YOURSELF

Basic SHELL Commands

File Commands

ls – directory listing
ls -al – formatted listing with hidden files
cd *dir* - change directory to *dir*
cd – change to home
pwd – show current directory
mkdir *dir* – create a directory *dir*
rm *file* – delete *file*
rm -r *dir* – delete directory *dir*
rm -f *file* – force remove *file*
rm -rf *dir* – force remove directory *dir* *
cp *file1 file2* – copy *file1* to *file2*
cp -r *dir1 dir2* – copy *dir1* to *dir2*; create *dir2* if it doesn't exist
mv *file1 file2* – rename or move *file1* to *file2* if *file2* is an existing directory, moves *file1* into directory *file2*
ln -s *file link* – create symbolic link *link* to *file*
touch *file* – create or update *file*
cat > *file* – places standard input into *file*
more *file* – output the contents of *file*
head *file* – output the first 10 lines of *file*
tail *file* – output the last 10 lines of *file*
tail -f *file* – output the contents of *file* as it grows, starting with the last 10 lines

Process Management

ps – display your currently active processes
top – display all running processes

System Info

date – show the current date and time
cal – show this month's calendar
uptime – show current uptime
w – display who is online
whoami – who you are logged in as
finger *user* – display information about *user*
uname -a – show kernel information
cat /proc/cpuinfo – cpu information
cat /proc/meminfo – memory information
man *command* – show the manual for *command*
df – show disk usage
du – show directory space usage
free – show memory and swap usage
whereis *app* – show possible locations of *app*
which *app* – show which *app* will be run by default

Compression

tar cf *file.tar files* – create a tar named *file.tar* containing *files*
tar xf *file.tar* – extract the files from *file.tar*
tar czf *file.tar.gz files* – create a tar with Gzip compression
tar xzf *file.tar.gz* – extract a tar using Gzip
tar cjf *file.tar.bz2* – create a tar with Bzip2 compression
tar xjf *file.tar.bz2* – extract a tar using Bzip2
gzip *file* – compresses *file* and renames it to *file.gz*

CLARU
WAY TO REINVEN

Basic SHELL Commands

File system Commands	
ls	lists directories and files
ls -a	lists all files including hidden files
ls -lh	formatted list including more data
ls -t	lists sorted by date
pwd	returns path to working directory
cd *dir*	changes directory
cd ..	goes to parent directory
cd /	goes to root directory
cd	goes to home directory
touch *file_name*	creates en empty file
cp *file file_copy*	copy a file
cp -r	copy files contained in directories
rm *file*	deletes a file
rm -r *dir*	deletes a directory and its files
mv *file1 file2*	moves or renames a file
mkdir *dir_name*	creates a directory
rmdir *dir_name*	deletes a directory

Text handling commands	
command > *file*	saves STDOUT in a file
command >> *file*	appends STDOUT in a file
cat *file*	concatenate and print files
cat *file1 file2 > file3*	merges files 1 and 2 into *file3*
cat *fasta > all.fasta	concatenates all fasta files in the current directory
head *file*	prints first lines from a file
head -n 5 *file*	prints first five lines from a file
tail *file*	prints last lines from a file
tail -n 5 *file*	prints last five lines from a file
less *file*	view a file
less -N *file*	includes line numbers
less -S *file*	wraps long lines
grep '*pattern*' *file*	Prints lines matching a pattern
grep -c '*pattern*' *file*	counts lines matching a pattern
cut -f 1,3 *file*	retrieves data from selected columns in a tab-delimited file
sort *file*	sorts lines from a *file*

Using the Command Line to Get Help

CLARUSWAY
WAY TO REINVENT YOURSELF

Table of Contents

1 Man Pages

Man Pages

man [command]

A man page (short for manual page) is a form of software documentation usually found on a Unix or Unix-like operating system.

if we install a package to do some task, the man Page for that package will typically be installed at the same time. This gives us the ability to take a look at that documentation and make sure that we're using it in a manner consistent with its design.

The man page for a particular command is invoked by preceding the command with **man.**

Man Pages

man ls

```
LS(1)                                   User Commands                                   LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about the FILEs (the current directory by default).  Sort entries alphabetically if none of
       -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              scale sizes by SIZE before printing them; e.g., '--block-size=M' prints sizes  in  units  of  1,048,576
              bytes; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with  -lt:  sort  by,  and show, ctime (time of last modification of file status information); with -l:
              show ctime and sort by name; otherwise: sort by ctime, newest first

       -C     list entries by columns

       --color[=WHEN]
              colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below

       -d, --directory
              list directories themselves, not their contents
Manual page ls(1) line 1 (press h for help or q to quit)
```

NAME
Program or Function name(s) followed by descriptions of functionality.

SYNOPSIS
A short overview of available options

DESCRIPTION
Detailed information about arguments and options.

2 Info Pages

CLARUSWAY
WAY TO REINVENT YOURSELF

Info Pages

info [command]

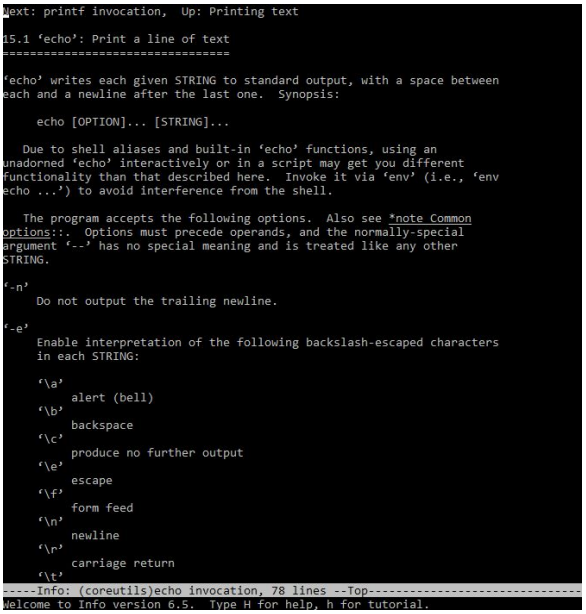
Info pages are additional documentation with more robust capability in detail. Info Page normally provides more detailed information about a command than its respective man page. Additionally, Info uses a structure for linking these pages together, and they may be assembled into a larger collection.

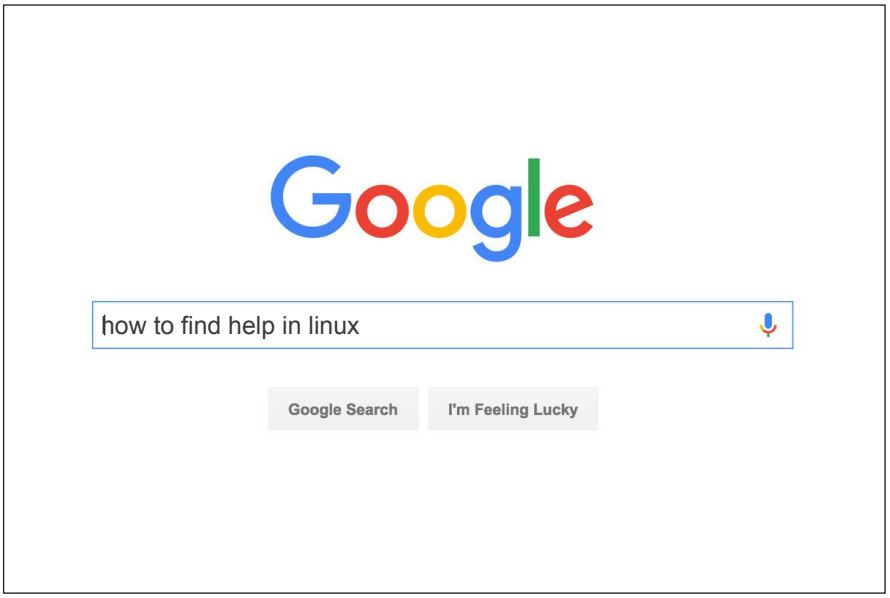
The info page for a particular command is invoked by preceding the command with **info**

CLARUSWAY
WAY TO REINVENT YOURSELF

Info Pages

info echo





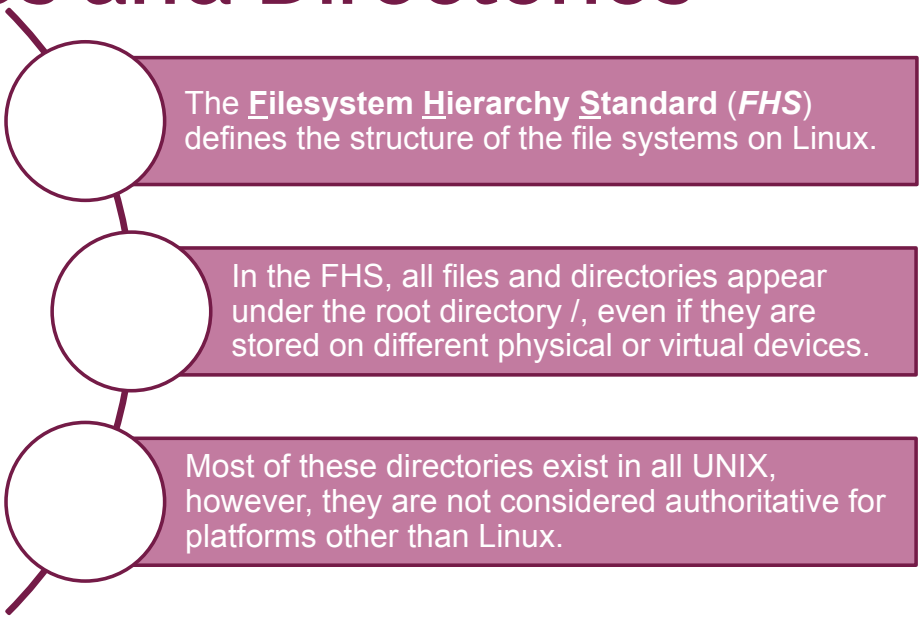
CLARUSWAY
WAY TO REINVENT YOURSELF

amazon web services

Break
return @ 9pm

Files and Directories

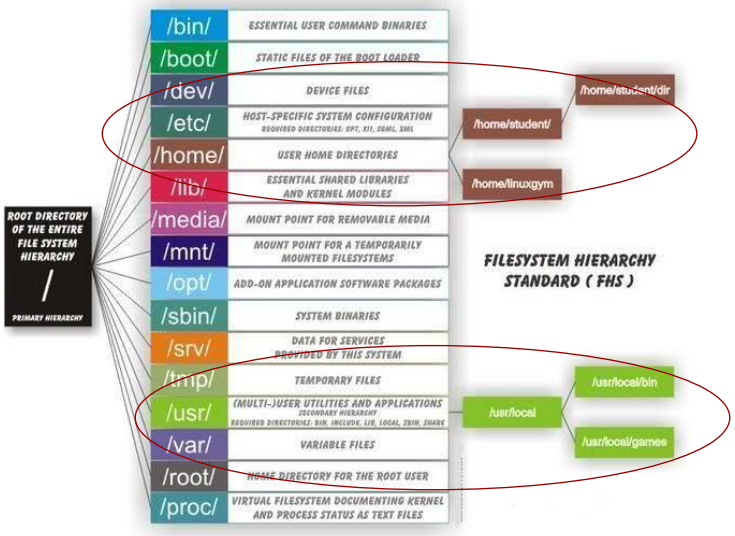
Files and Directories



Files and Directories

/boot	Boot loader files
/dev	Essential device files
/mnt	Temporarily mounted filesystems.
/proc	Contains information about system
/bin	Essential command binaries
/sbin	Essential system binaries
/lib	Libraries essential for the binaries
/opt	Optional application packages
/root	root directory of the root user
/home	Users' home directories
/etc	Host-specific configuration files
/var	Variable data files
/tmp	Temporary files

Files and Directories

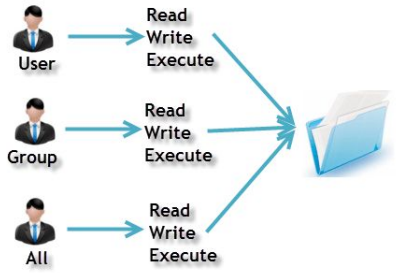


2 File Permission

File Permission

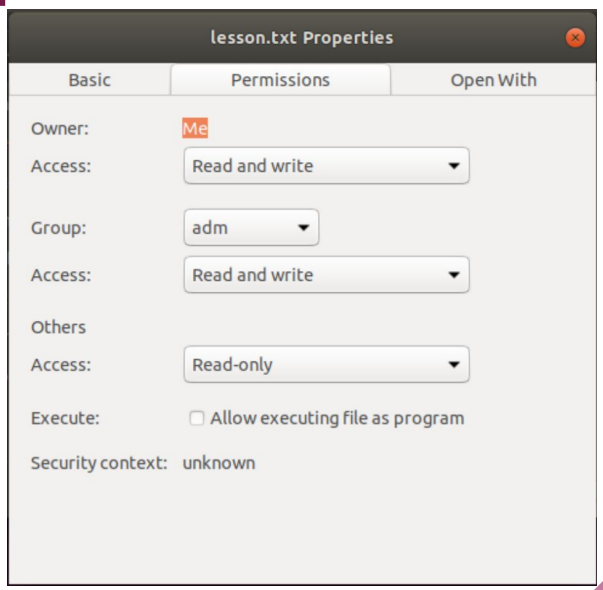
Ownership	
User	• A user is the owner of the file.
Group	• A user- group can contain multiple users.
Other	• Any other user who has access to a file.

Permission	
Read	• This permission give you the authority to open and read a file.
Write	• The write permission gives you the authority to modify the contents of a file.
Execute	• you cannot run a program unless the execute permission is set.



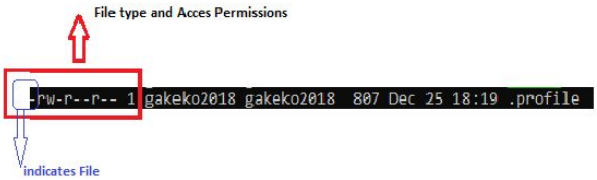
File Permission

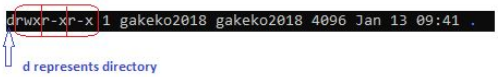
```
raymond@clarusway-linux: ~
File  Edit  View  Search  Terminal  Help
raymond@clarusway-linux:~$ ls -l lesson.txt
-rw-rw-r-- 1 raymond adm 8 Mar  2 21:19 lesson.txt
raymond@clarusway-linux:~$
```

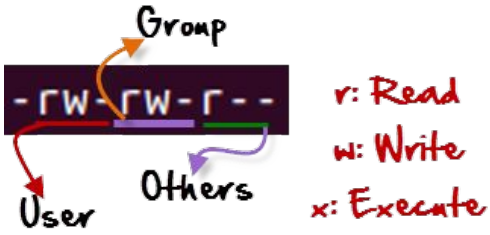


File Permission

```
gakeko2018@DESKTOP-JAO7K2U:~$ ls
cert.pem
gakeko2018@DESKTOP-JAO7K2U:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .local  .profile  .ssh  cert.pem
gakeko2018@DESKTOP-JAO7K2U:~$ ls -al
total 12
drwxr-xr-x 1 gakeko2018 gakeko2018 4096 Jan 13 09:41 .
drwxr-xr-x 1 root       root       4096 Dec 25 18:19 ..
-rw------- 1 gakeko2018 gakeko2018  236 Jan 14 12:21 .bash_history
-rw-r--r-- 1 gakeko2018 gakeko2018  220 Dec 25 18:19 .bash_logout
-rw-r--r-- 1 gakeko2018 gakeko2018 3771 Dec 25 18:19 .bashrc
drwxrwxrwx 1 gakeko2018 gakeko2018 4096 Jan 13 09:38 .local
-rw-r--r-- 1 gakeko2018 gakeko2018  807 Dec 25 18:19 .profile
drwx------ 1 gakeko2018 gakeko2018 4096 Jan 13 09:41 .ssh
-r-------- 1 gakeko2018 gakeko2018 1675 Jan 13 09:38 cert.pem
```







r = read permission
w = write permission
x = execute permission
- = no permission

CLARUSWAY
WAY TO REINVENT YOURSELF

File Permission

Changing Permission with chmod Command

We can use the chmod command which stands for **change mode**.
we can set permissions (read, write, execute) on a file/directory for the owner, group and the world.

chmod permissions filename

chmod u=rwx,g=rx,o=r myfile

Symbol	Permission Type
---	No Permission
--x	Execute
-w-	Write
-wx	Execute+Write
r--	Read
r-x	Read+Execute
rw-	Read+Write
rwx	Read+Write+Execute

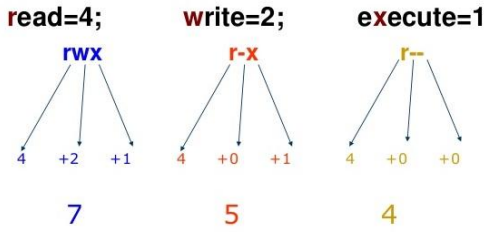
CLARUSWAY
WAY TO REINVENT YOURSELF

File Permission

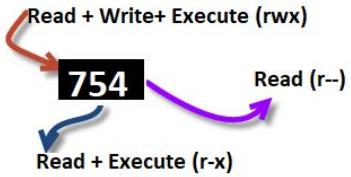
```
root@DESKTOP-4QQ1S5L:~# ls -l
total 0
-rw-rw-rw- 1 root root 0 Dec 29 17:53 file1
-r--r--rwx 1 root root 0 Dec 29 17:53 file2
root@DESKTOP-4QQ1S5L:~# chmod 754 file2
root@DESKTOP-4QQ1S5L:~# ls -l file2
-rwxr-xr-- 1 root root 0 Dec 29 17:53 file2
root@DESKTOP-4QQ1S5L:~#
```

754 code says;
- Owner can read, write and execute
- User's group can read and execute
- Other can only read

Permissions

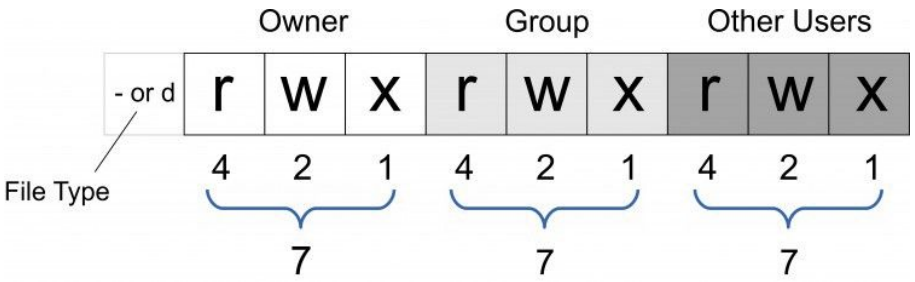



```
chmod u=rwx,g=rx,o=r myfile
chmod 754 myfile
```



CLARUSWAY
WAY TO REINVENT YOURSELF

File Permission



d	r (read)	w (write)	x (exec)	r (read)	- (write)	x (exec)	r (read)	- (write)	- (exec)
File type	Owner permissions			Group permissions			User permissions		
(directory)	4	2	1	4	2	1	4	2	1
	7			5			4		

Set permissions of myfile.txt to;

owner : full access
group : read and execute
others : no access

THANKS!

Any questions?