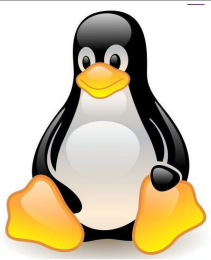


Using Directories Listing Files

Linux Essentials
Session-4

CLARUSWAY
WAY TO REINVENT YOURSELF

Recap

CLARUSWAY
WAY TO REINVENT YOURSELF

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]

pushd   goto another directory and save the current directory
        pushd [directory]

popd    return to the last directory saved with pushd
        popd
```


Recap: Command Line Basics (4)

```
head [-n]  show the first n (default 10) lines of a file
           head [filename]

tail [-n]  show the last n (default 10) lines of a file
           head [filename]

tail -f    show the last lines of a file and keep refreshing
           tail -f [filename]

sudo su    switch to root user
           sudo su

chmod      change mode (permissions) on a file
           chmod [permissions] [filename]
```

Recap: Command Line Basics (5)

```
grep        search for a regular expression
            grep [expression] [filename]

|           "pipe": send output to the next command
            e.g. cat filename | grep "hello"

Also:
 -  you can use absolute or relative paths
 -  use the "tab" key for autocomplete
 -  commands and file names are case sensitive
```


Recap: Globbing

```
*       match any number of characters
        e.g. ls f*.*

?       match exactly one character
        e.g. ls file?.txt

[ ]     match range of characters in brackets
        e.g. ls | grep [a-m]*.*

^       match pattern at the start of the string
        e.g. ls | grep ^b

$       match pattern at the end of the string
        e.g. ls | grep s$
```

Exercise 1

ls	pushd
cd	popd
mkdir	head
rmdir	tail
pwd	sudo
touch	chmod
rm	*
cp	?
mv	[ ]
cat	^
echo	$
>	
>>	

1. show the first 5 lines of the file httpd.conf
2. continuously show the last line in the file httpd.log as it updates
3. output the directory listing to a file called listing.txt
4. create a hidden file (choose your own filename)
5. delete the directory /tmp/old which is not empty
6. restore the directory /tmp/old from the recycle bin
7. go to the directory /var/log, but save your current directory so you can return
8. return to the directory you just left

CLARUSWAY
WAY TO REINVENT YOURSELF

Exercise 1

ls	pushd
cd	popd
mkdir	head
rmdir	tail
pwd	sudo
touch	chmod
rm	*
cp	?
mv	[ ]
cat	^
echo	$
>	
>>	

1. show the first 5 lines of the file httpd.conf
   `head -5 httpd.conf`
2. continuously show the last line in the file httpd.log as it updates
   `tail -f httpd.log`
3. output the directory listing to a file called listing.txt
   `ls > listing.txt  (or ls >> listing.txt)`
4. create a hidden file (choose your own filename)
   `touch .myfile.txt [must start with a '.']`
5. delete the directory /tmp/old which is not empty
   `rm -rf /tmp/old`
6. restore the directory /tmp/old from the recycle bin
   `trick question: there is no Recycle Bin!  Careful!!`
7. go to the directory /var/log, but save your current directory so you can return
   `pushd /var/log`
8. return to the directory you just left
   `popd`

CLARUSWAY
WAY TO REINVENT YOURSELF

CLARUSWAY
WAY TO REINVENT YOURSELF

Break
return @ 8:00pm

amazon web services

Exercise 2

ls	pushd
cd	popd
mkdir	head
rmdir	tail
pwd	sudo
touch	chmod
rm	*
cp	?
mv	[ ]
cat	^
echo	$
>	
>>	

1. list all of the files that look like this:
   file1.txt, file10.txt, file2.txt, file 100.txt, etc…
2. list all of the files that look like this:
   file1.txt, file2.txt, file3.txt...file9.txt
3. list all the files that start with numbers 0-9
4. list all the files that end with .sh
5. rename the file /var/log/httpd.log to /var/log/httpd.log.bak

Exercise 2

ls	pushd
cd	popd
mkdir	head
rmdir	tail
pwd	sudo
touch	chmod
rm	*
cp	?
mv	[ ]
cat	^
echo	$
>	
>>	

1. list all of the files that look like this:
   file1.txt, file10.txt, file2.txt, file 100.txt, etc…
   `ls file*.txt`
2. list all of the files that look like this:
   file1.txt, file2.txt, file3.txt...file9.txt
   `ls file?.txt`
3. list all the files that start with numbers 0-9
   `ls | grep ^[0-9]`
4. list all the files that end with .sh
   `ls | grep .sh$`
5. rename the file /var/log/httpd.log to /var/log/httpd.log.bak
   `mv /var/log/httpd.log /var/log/httpd.log.bak`

Quick intro to Binary Numbers

Binary Numbers

- In the **decimal (base 10)** system, we have numbers whose digits range from **0-9**
  - e.g. 5 4 1 7

- Similarly, in the **binary (base 2)** system, we have numbers whose digits range from **0-1**
  - e.g. 1 1 0 1

Decimal Equivalents

- In the **decimal** system, each digit has a value corresponding to a power of **10**
- i.e. 1, 10, 100, 1000, etc…
- e.g. 5417

5	4	1	7
1000	100	10	1

- So the value is:
  - 5**x1000** + 4**x100** + 1**x10** + 7**x1**
  - =5417

- In the **binary** system, each digit has a value corresponding to a power of **2**
- i.e. 1, 2, 4, 8, 16, 32 etc…
- e.g. 101

1	0	1
4	2	1

- So the value is:
  - 1**x4** + 0**x2** + 1**x1**
  - = 5

How Many Numbers?

- If you have a **decimal (base 10)** created from **n digits**, how many possible numbers can you have?
  - 1 digit
    - x
    - 0-9
    - 10 numbers
    - 10 = 10^1
  - 2 digits
    - x x
    - 0-99
    - 100 numbers
    - 10x10 = 10^2
  - 3 digits
    - x x x
    - 0-999
    - 1000 numbers
    - 10x10x10 = 10^3
  - n digits
    - x x x x …. x
    - 10^n

CLARUSWAY
WAY TO REINVENT YOURSELF

How Many Binary Numbers?

- If you have a **binary (base 2)** created from **n digits**, how many possible numbers can you have?
  - 1 digit
    - x
    - 0,1
    - 2 numbers
    - 2 = 2^1
  - 2 digits
    - x x
    - 00, 01, 10, 11
    - 4 numbers
    - 2x2 = 2^2
  - 3 digits
    - x x x
    - 000, 001, 010, 011, 100, 101, 110, 111
    - 8 numbers
    - 2x2x2 = 2^3
  - n digits
    - x x x x …. x
    - 2^n

CLARUSWAY
WAY TO REINVENT YOURSELF

And so …

- To get the decimal value of a binary number:

  ___ ___ ___ ___ ___
  16 8 4 2 1 ← add this number to the total if the digit above is a '1'

- Examples

  0 0 1 0 0 = 4

  1 0 1 0 0 = 16+4 = 20

  1 1 0 0 1 = 16+8+1 = 25

  0 0 1 0 1 = 4 + 1 = 5

And also …

- To get the total possible numbers:

  — — — — — —

  ← n digits →

- 2^n

Exercise 3

1. What is the decimal equivalent of 0 0 1
2. What is the decimal equivalent of 1 0 1
3. What is the decimal equivalent of 1 1 0
4. What is the decimal equivalent of 1 1 1
5. What is the decimal equivalent of 1 0 0
6. How many possible numbers for a binary number with three digits - i.e. _ _ _?
7. What is the binary equivalent of 7?
8. What is the binary equivalent of 4?
9. What is the binary equivalent of 5?

CLA
WAY TO F

Exercise 3

1. What is the decimal equivalent of 0 0 1
   1
2. What is the decimal equivalent of 1 0 1
   5
3. What is the decimal equivalent of 1 1 0
   6
4. What is the decimal equivalent of 1 1 1
   7
5. What is the decimal equivalent of 1 0 0
   4
6. How many possible numbers for a binary number with three digits - i.e. _ _ _?
   2^3 = 2x2x2 = 8
7. What is the binary equivalent of 7?
   1 1 1
8. What is the binary equivalent of 4?
   1 0 0
9. What is the binary equivalent of 5?
   1 0 1

CLA
WAY TO F

1

Files and Directories

CLARUSWAY
WAY TO REINVENT YOURSELF

Files and Directories

- The file system hierarchy standard (FHS) defines the structure of the file systems on Linux.
- In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices.
- Most of these directories exist in all UNIX, however, they are not considered authoritative for platforms other than Linux.

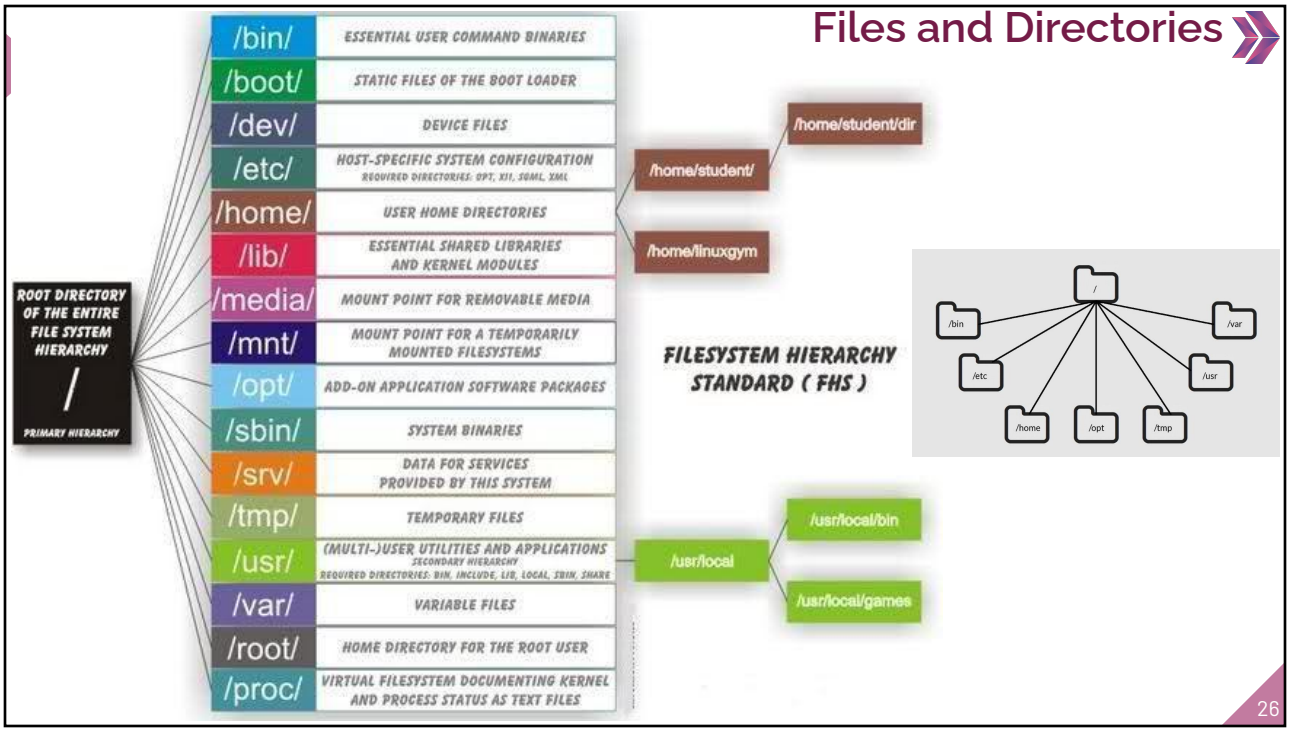
CLARUSWAY
WAY TO REINVENT YOURSELF

Files and Directories

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

CLARUSWAY
WAY TO REINVENT YOURSELF

Files and Directories

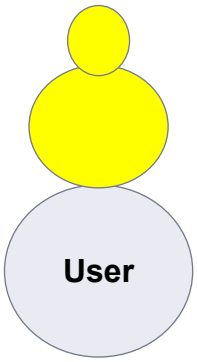


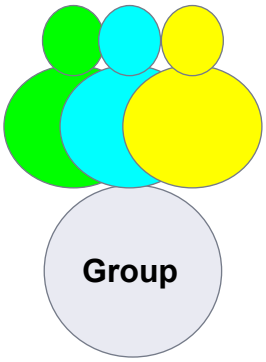
2 File Permission

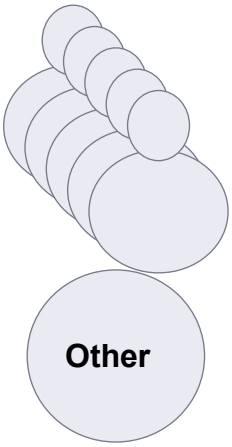
CLARUSWAY
WAY TO REINVENT YOURSELF

File Permission

Ownership

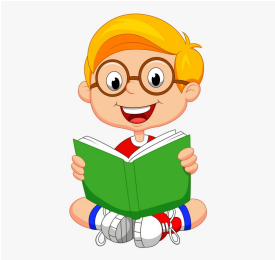






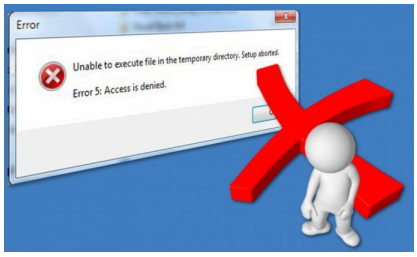
File Permission

Permissions



Read

Write

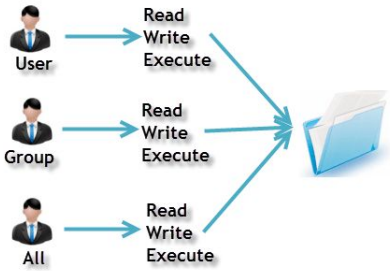


Execute

File Permission

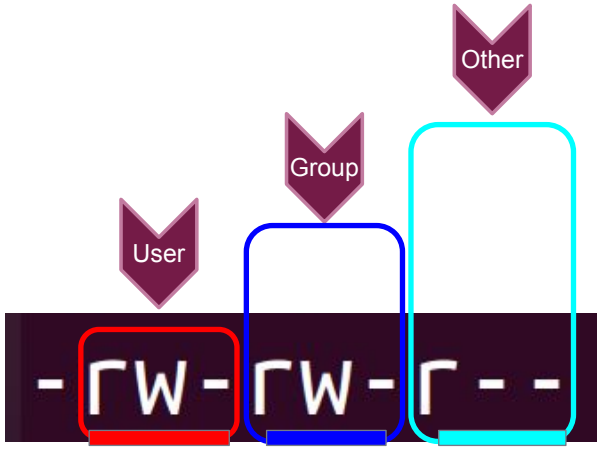
Ownership	
User	• A user is the owner of the file.
Group	• A user- group can contain multiple users.
Other	• Any other user who has access to a file.

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.

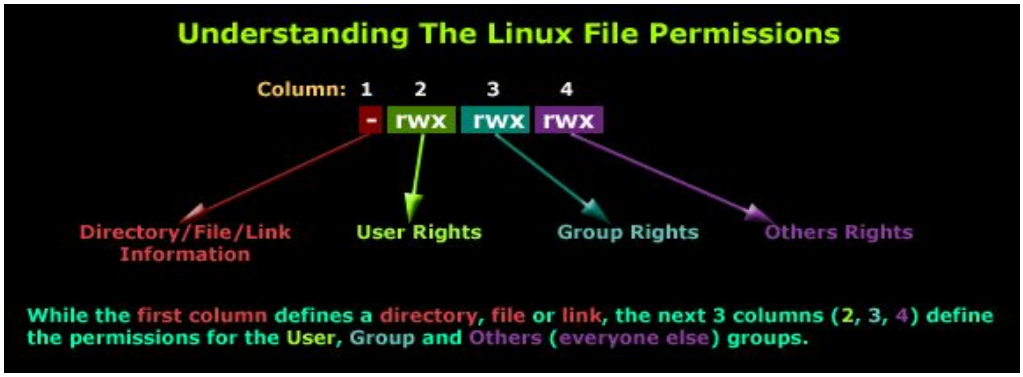


File Permission

Ownership



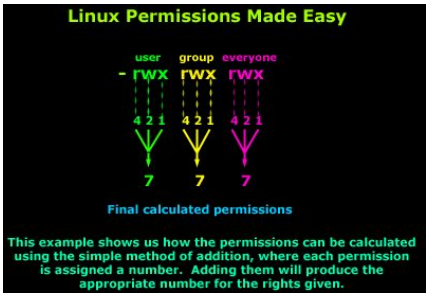

```
-rw-rw-r-- 1 zk zk 0 Dec  7 15:39 html.txt
```


```
# ls -l file
-rw-r--r-- 1 root root 0 Nov 19 23:49 file
```

Other (r - -)
Group (r- -)
Owner (rw-)
File type

r = Readable
w = Writeable
x = Executable
- = Denied



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
```

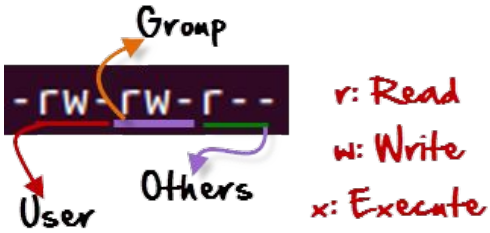
File type and Acces Permissions

```
-rw-r--r-- 1 gakeko2018 gakeko2018  807 Dec 25 18:19 .profile
```

indicates File

```
drwxr-xr-x 1 gakeko2018 gakeko2018 4096 Jan 13 09:41 .
```

d represents directory




```
-rw-rw-r--
```

no execute permission

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

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

Break

return @ 9:00pm

amazon web services

Who Can Change File Permissions?

Only **file owner** and **root**

Octal Notation

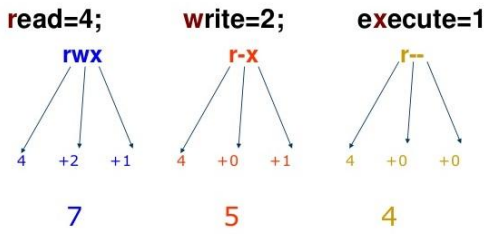
```
root@DESKTOP-4QQ1S5L:~# ls -l
total 0
-rw-rw-rw- 1 root root 0 Dec 29 17:53 file1
-r--r--rwx 1 root root 0 Dec 29 17:53 file2
root@DESKTOP-4QQ1S5L:~# chmod 754 file2
root@DESKTOP-4QQ1S5L:~# ls -l file2
-rwxr-xr-- 1 root root 0 Dec 29 17:53 file2
root@DESKTOP-4QQ1S5L:~#
```

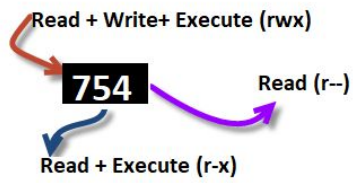
754 code says;

- Owner can read, write and execute
- User's group can read and execute
- Other can only read

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

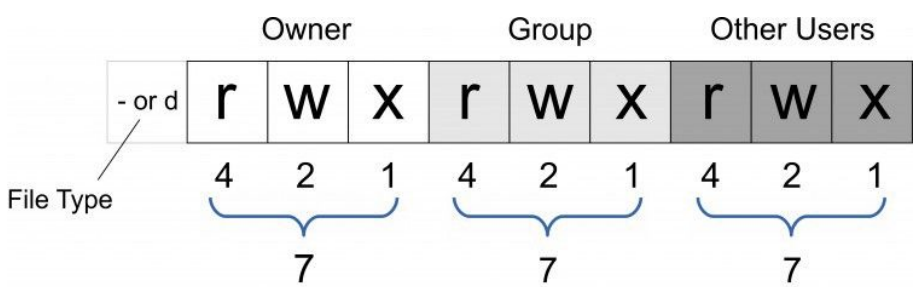
Permissions





CLARUSWAY
WAY TO REINVENT YOURSELF

File Permission



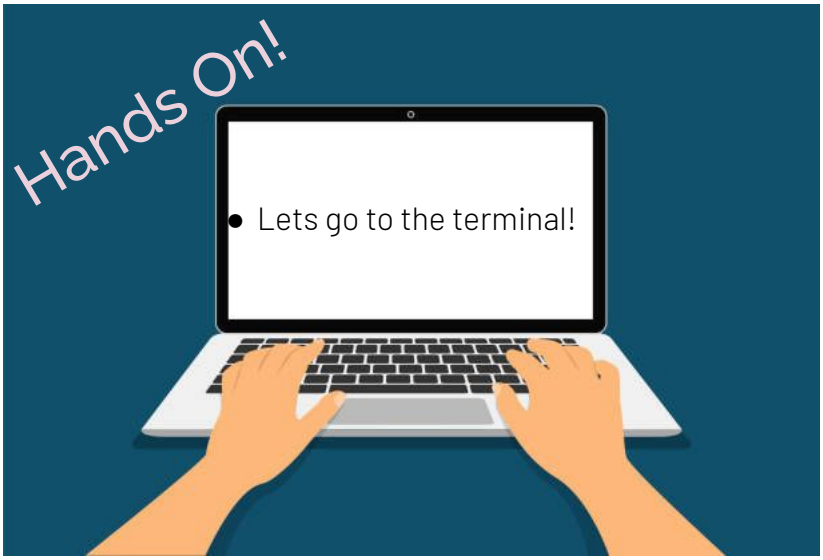
d	r read	w write	x exec	r read	- write	x exec	r read	- write	- exec
File type	Owner permissions			Group permissions			User permissions		
(directory)	4	2	1	4	2	1	4	2	1
	7			5			4		

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

By using three octal digits, we can set the file mode for the owner, group owner, and world.

File Attributes	Meaning
-rwx------	A regular file that is readable, writable, and executable by the file's owner. No one else has any access.
-rw-------	A regular file that is readable and writable by the file's owner. No one else has any access.
-rw-r--r--	A regular file that is readable and writable by the file's owner. Members of the file's owner group may read the file. The file is world-readable.
-rwxr-xr-x	A regular file that is readable, writable, and executable by the file's owner. The file may be read and executed by everybody else.
-rw-rw----	A regular file that is readable and writable by the file's owner and members of the file's group owner only.
lrwxrwxrwx	A symbolic link. All symbolic links have “dummy” permissions. The real permissions are kept with the actual file pointed to by the symbolic link.
drwxrwx---	A directory. The owner and the members of the owner group may enter the directory and create, rename and remove files within the directory.
drwxr-x---	A directory. The owner may enter the directory and create, rename, and delete files within the directory. Members of the owner group may enter the directory but cannot create, delete, or rename files.

Notation	Meaning
`u+x`	Add execute permission for the owner.
`u-x`	Remove execute permission from the owner.
`+x`	Add execute permission for the owner, group, and world. This is equivalent to `a+x`.
`o-rw`	Remove the read and write permissions from anyone besides the owner and group owner.
`go=rw`	Set the group owner and anyone besides the owner to have read and write permission. If either the group owner or the world previously had execute permission, it is removed.
`u+x,go=rx`	Add execute permission for the owner and set the permissions for the group and others to read and execute. Multiple specifications may be separated by commas.



3 Ping & SSH Command

CLARUSWAY
WAY TO REINVENT YOURSELF

Ping Command

Ping or Packet Internet Groper is a network administration utility used to check the connectivity status between a source and a destination device.

ping host-name/IP

```
ping 54.93.34.220
```

```
gakeko2018@DESKTOP-JAO7K2U:~$ ping 54.93.34.220
PING 54.93.34.220 (54.93.34.220) 56(84) bytes of data.
64 bytes from 54.93.34.220: icmp_seq=1 ttl=243 time=62.6 ms
64 bytes from 54.93.34.220: icmp_seq=2 ttl=243 time=93.5 ms
64 bytes from 54.93.34.220: icmp_seq=3 ttl=243 time=66.8 ms
64 bytes from 54.93.34.220: icmp_seq=4 ttl=243 time=67.6 ms
64 bytes from 54.93.34.220: icmp_seq=5 ttl=243 time=62.7 ms
64 bytes from 54.93.34.220: icmp_seq=7 ttl=243 time=84.6 ms
64 bytes from 54.93.34.220: icmp_seq=8 ttl=243 time=64.6 ms
64 bytes from 54.93.34.220: icmp_seq=9 ttl=243 time=72.0 ms
```

CLARUSWAY
WAY TO REINVENT YOURSELF

Ping Command

The `ping` command is one of the most used utilities for troubleshooting, testing, and diagnosing network connectivity issues.

Ping works by sending one or more ICMP (Internet Control Message Protocol) Echo Request packages to a specified destination IP on the network and waits for a reply. When the destination receives the package, it will respond back with an ICMP echo reply.

With the `ping` command, you can determine whether a remote destination IP is active or inactive. You can also find the round-trip delay in communicating with the destination and check whether there is a packet loss.

Ping Command

The `ping` command resolves the domain name into an IP address and starts sending ICMP packages to the destination IP. If the destination IP is reachable it will respond back and the ping command prints a line that includes the following fields:

- The number of data bytes. The default is 56, which translates into 64 ICMP data bytes - `64 bytes`
- The IP address of the destination - `from ….`
- The ICMP sequence number for each packet. `icmp_seq=1`
- The Time to Live. - `ttl=53`
- The ping time, measured in milliseconds which is the round trip time for the packet to reach the host, and for the response to return to the sender. - `time=41.4 ms`

By default, the interval between sending a new packet is one second.

The `ping` command will continue to send ICMP packages to the Destination IP address until it receives an interrupt. To stop the command, just hit the `Ctrl+C` key combination.

Ping Command

```
$ ping clarusway.com

Pinging clarusway.com [54.164.151.235] with 32 bytes of data:
Reply from 54.164.151.235: bytes=32 time=132ms TTL=237
Reply from 54.164.151.235: bytes=32 time=130ms TTL=237
Reply from 54.164.151.235: bytes=32 time=130ms TTL=237
Reply from 54.164.151.235: bytes=32 time=130ms TTL=237

Ping statistics for 54.164.151.235:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 130ms, Maximum = 132ms, Average = 130ms
```

```
$ ping www.google.com

Pinging www.google.com [172.217.169.132] with 32 bytes of data:
Reply from 172.217.169.132: bytes=32 time=19ms TTL=116
Reply from 172.217.169.132: bytes=32 time=18ms TTL=116
Reply from 172.217.169.132: bytes=32 time=18ms TTL=116
Reply from 172.217.169.132: bytes=32 time=19ms TTL=116

Ping statistics for 172.217.169.132:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 18ms, Maximum = 19ms, Average = 18ms
```


Ping Command

```
$ ping 54.164.151.235

Pinging 54.164.151.235 with 32 bytes of data:
Reply from 54.164.151.235: bytes=32 time=131ms TTL=237
Reply from 54.164.151.235: bytes=32 time=130ms TTL=237
Reply from 54.164.151.235: bytes=32 time=130ms TTL=237
Reply from 54.164.151.235: bytes=32 time=130ms TTL=237

Ping statistics for 54.164.151.235:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 130ms, Maximum = 131ms, Average = 130ms
```

SSH Command

* ssh stands for “Secure Shell”.
* It is a protocol used to securely connect to a remote server/system.

ssh user@host(IP/Domain_name)

```
ssh -i cert.pem ec2-user@54.93.34.220
```

```
gakeko2018@DESKTOP-JAO7K2U:~$ ssh -i cert.pem ec2-user@54.93.34.220
The authenticity of host '54.93.34.220 (54.93.34.220)' can't be established.
ECDSA key fingerprint is SHA256:lvCnUtJiig4s2U4aojBonZOSbzGPBMOpB9yPPoGjVEo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.93.34.220' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 13 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-35-15 ~]$
```

CLARUSWAY
WAY TO REINVENT YOURSELF

Basic Shell Commands

whoami	current user
hostname	shows the system hostname
hostname -i	Show the IP address of the system

```
robert@robert-virtual-machine: ~
robert@robert-virtual-machine:~$ users
robert
robert@robert-virtual-machine:~$ hostname
robert-virtual-machine
robert@robert-virtual-machine:~$ hostname -i
127.0.1.1
robert@robert-virtual-machine:~$
```

CLARUSWAY
WAY TO REINVENT YOURSELF

Kahoot!

Kahoot!

THANKS!

Any questions?