

# Using Directories Listing Files

Linux Essentials  
Session-4



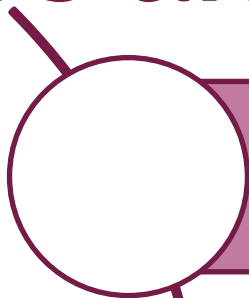
1

# Files and Directories

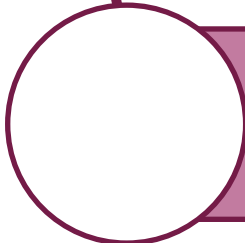




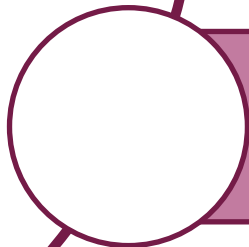
# Files and Directories



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.



# 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

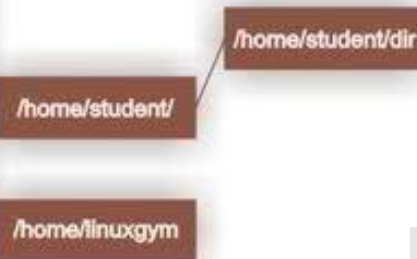
/	the root directory
/bin	user binaries
/boot	static boot files
/dev	device files
/etc	configuration files
/home	home directories
/lib	shared libraries
/mnt	temporary mount points
/opt	optional packages
/proc	kernel and process files
/root	root user home directory
/run	application state files
/sbin	system administration binaries
/srv	service data
/tmp	temporary files
/usr	user binaries
/var	variable data files

**ROOT DIRECTORY  
OF THE ENTIRE  
FILE SYSTEM  
HIERARCHY**

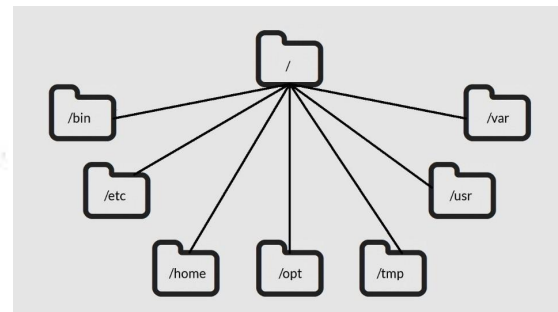
**/**

**PRIMARY HIERARCHY**

/bin/	ESSENTIAL USER COMMAND BINARIES
/boot/	STATIC FILES OF THE BOOT LOADER
/dev/	DEVICE FILES
/etc/	HOST-SPECIFIC SYSTEM CONFIGURATION <small>REQUIRED DIRECTORIES: OPT, X11, SGML, XML</small>
/home/	USER HOME DIRECTORIES
/lib/	ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/	MOUNT POINT FOR REMOVABLE MEDIA
/mnt/	MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/	ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/	SYSTEM BINARIES
/srv/	DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/	TEMPORARY FILES
/usr/	(MULTI-)USER UTILITIES AND APPLICATIONS <small>SECONDARY HIERARCHY</small> <small>REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE</small>
/var/	VARIABLE FILES
/root/	HOME DIRECTORY FOR THE ROOT USER
/proc/	VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES



## FILESYSTEM HIERARCHY STANDARD ( FHS )





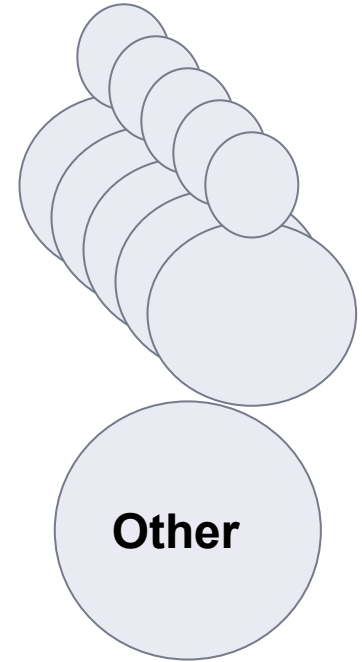
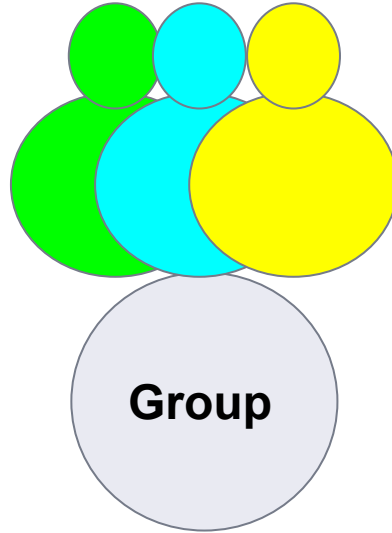
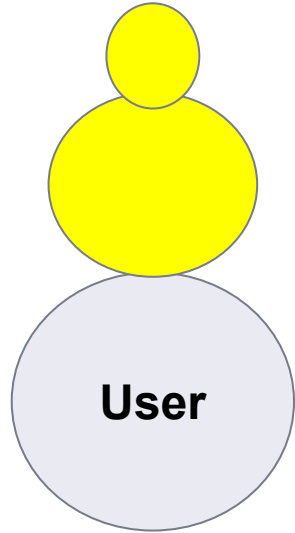
2

# File Permission

# File Permission



## Ownership



# File Permission



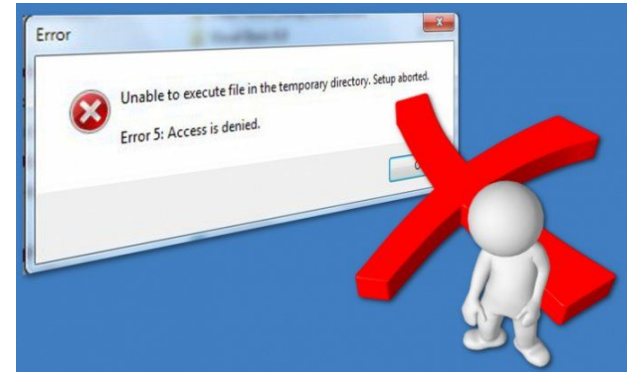
# Permissions



**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

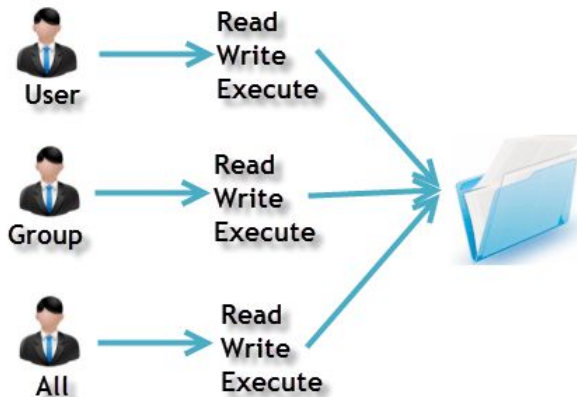
- This permission give you the authority to open and read a file.

Write

- The write permission gives you the authority to modify the contents of a file.

Execute

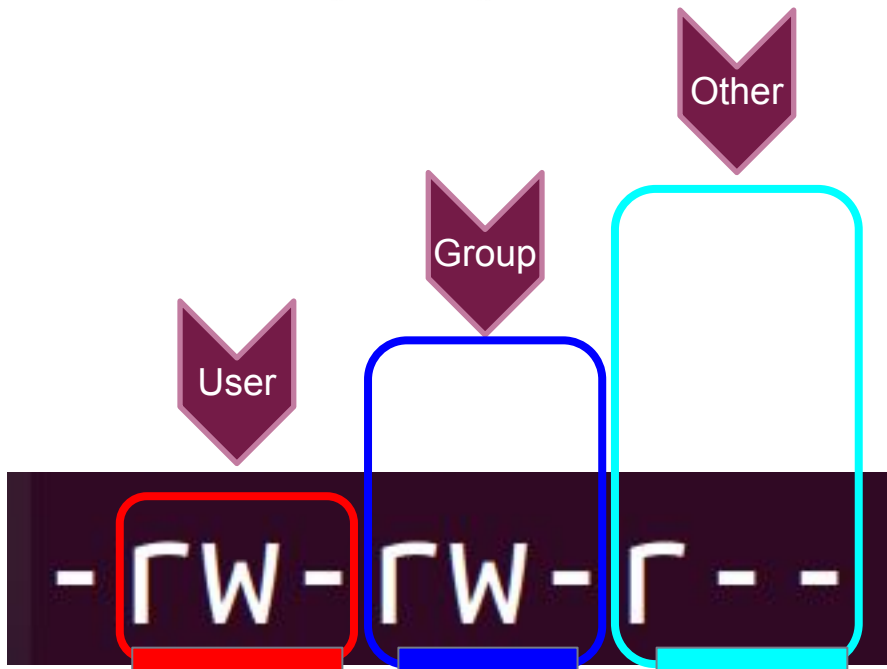
- you cannot run a program unless the execute permission is set.



# File Permission



## Ownership



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



# 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:~$
```

lesson.txt Properties

Basic	Permissions	Open With
Owner:	Me	
Access:	Read and write	
Group:	adm	
Access:	Read and write	
Others		
Access:	Read-only	
Execute:	<input type="checkbox"/> Allow executing file as program	
Security context:	unknown	

# File Permission



```
gakeko2018@DESKTOP-JA07K2U:~$ ls
cert.pem
gakeko2018@DESKTOP-JA07K2U:~$ ls -la
.  .. .bash_history .bash_logout .bashrc .local .profile .ssh cert.pem
gakeko2018@DESKTOP-JA07K2U:~$ 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 Access 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

Group

User Others

r: Read  
w: Write  
x: Execute

-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

# 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

**read=4;**

**rwX**

4 +2 +1

**7**

**write=2;**

**r-X**

4 +0 +1

**5**

**execute=1**

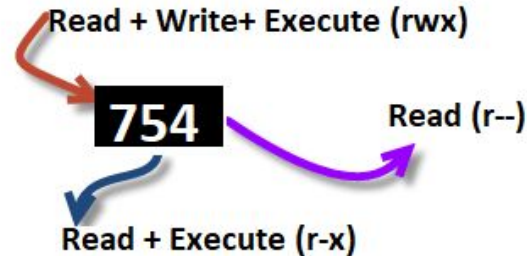
**r--**

4 +0 +0

**4**

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

`chmod 754 myfile`





# File Permission

zk@ubuntu:~/ASSIGNMENT/Lessons/HTML\$ ls -l

total 0

! !

-rwx-----	1	zk	zk	0	Dec	7	15:39	cas.txt
-----rwx---	1	zk	zk	0	Dec	7	15:39	html.txt
-----rwx	1	zk	zk	0	Dec	7	15:39	java.txt
-rwxrwxrwx	1	zk	zk	0	Dec	7	17:10	js.js
-rwxrw---x	1	zk	zk	0	Dec	7	17:11	k.txt
-r--r--r--	1	zk	zk	0	Dec	7	17:13	l.txt



# File Permission



	Owner			Group			Other Users		
- or d	r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1	
7			7			7			

Read + Write + Execute (rwx)

764

Read (r--)

Read + Write (rw-)

d	r	w	x	r	-	x	r	-	-
	read	write	exec	read	write	exec	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

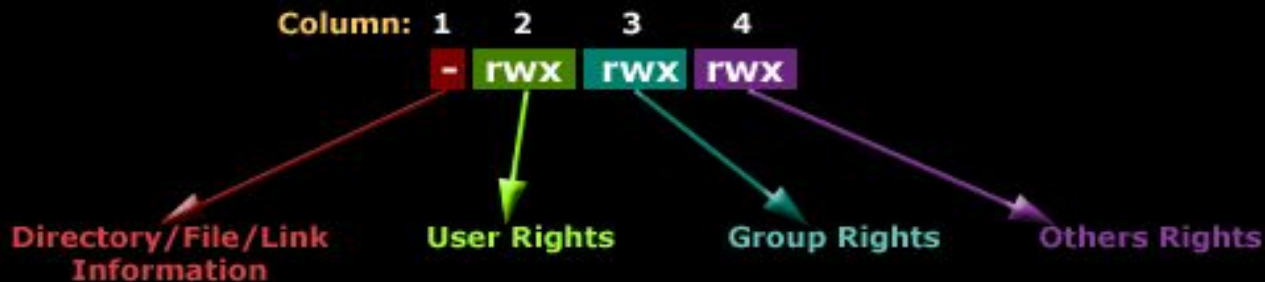
others : no access



Students, write your response!



## Understanding The Linux File Permissions



While the first column defines a directory, file or link, the next 3 columns (2, 3, 4) define the permissions for the User, Group and Others (everyone else) groups.

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

File type

Owner (rw-)

Group (r--)

Other (r--)

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

### Linux Permissions Made Easy

user group everyone

- rwx rwx rwx

4 2 1 4 2 1 4 2 1

7 7 7

Final calculated permissions

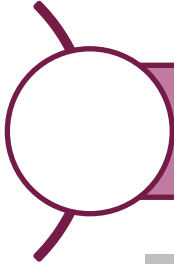
This example shows us how the permissions can be calculated using the simple method of addition, where each permission is assigned a number. Adding them will produce the appropriate number for the rights given.



3

# Ping & SSH Command

# 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-JA07K2U:~$ 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
```



# 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.

The `ping` utility is a part of the `iputils` (or `iputils-ping`) package, which is pre-installed on nearly all Linux distributions. `ping` is also available on Windows, macOS, and FreeBSD.



# 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-JA07K2U:~$ 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 ~]$
```



# Kahoot!





# THANKS!

## **Any questions?**



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.



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

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



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.