CH8:

The use of glob characters in Linux is similar to what many operating systems refer to as "wildcard" characters. Using glob characters, you match filenames using patterns.

will use the echo command to display this expansion process.

 following echo command to display all filenames in the current directory that match the glob pattern \*:

**sysadmin@localhost:~$** echo \*

Desktop Documents Downloads Music Pictures Public Templates Videos

**sysadmin@localhost:~$**

The following commands will display all the files in the current directory that start with the letter D, and the letter P:

**sysadmin@localhost:~$** echo D\*

Desktop Documents Downloads

**sysadmin@localhost:~$** echo P\*

Pictures Public

**sysadmin@localhost:~$**

The asterisk \* can be used anywhere in the string. The following command will display all the files in your current directory that end in the letter s:

**sysadmin@localhost:~$** echo \*s

Documents Downloads Pictures Templates Videos

**sysadmin@localhost:~$**

Type the following to display the filenames that are exactly six characters long:

**sysadmin@localhost:~$** echo ??????

Public Videos

**sysadmin@localhost:~$**

 Type the following to display the file names that start with the letter D and are exactly nine characters long:

**sysadmin@localhost:~$** echo D????????

Documents Downloads

Type the following to display the file names that start with the letter D and are exactly nine characters long:

**sysadmin@localhost:~$** echo D????????

Documents Downloads

In the first example, the first character of the file name can be either a D or a P. In the second example, the first character can be any character except a D or P:

**sysadmin@localhost:~$** echo [DP]\*

Desktop Documents Downloads Pictures Public

**sysadmin@localhost:~$** echo [!DP]\*

Music Templates Videos

In these next examples, a range of characters will be specified. In the first example, the first character of the file name can be any character starting at D and ending at P. In the second example, this range of characters is negated, meaning any single character will match as long as it is not between the letters D and P:

**sysadmin@localhost:~$** echo [D-P]\*

Desktop Documents Downloads Music Pictures Public

**sysadmin@localhost:~$** echo [!D-P]\*

Templates Videos

Make a copy of the /etc/hosts file and place it in the current directory. Then, list the contents of the current directory before and after the copy:

**sysadmin@localhost:~$** ls

**Desktop Documents Downloads Music Pictures Public Templates Videos**

**sysadmin@localhost:~$** cp /etc/hosts hosts

**sysadmin@localhost:~$** ls

**Desktop Downloads Pictures Templates** hosts

**Documents Music Public Videos**

**sysadmin@localhost:~$**

rm command is used to delete a file, -v switch displays the source and target when the cp command is executed.

 -v switch displays the source and target when the cp command is executed:

**Source**

`/etc/hosts'-> `hosts'

**Target**

`/etc/hosts'-> `hosts'

**sysadmin@localhost:~$** rm hosts

**sysadmin@localhost:~$** ls

**Desktop Documents Downloads Music Pictures Public Templates Videos**

**sysadmin@localhost:~$** cp -v /etc/hosts hosts

`/etc/hosts' -> `hosts'

**sysadmin@localhost:~$** ls

**Desktop Downloads Pictures Templates** hosts

**Documents Music Public Videos**

**sysadmin@localhost:~$**

the following commands to copy the /etc/hosts file, using the period . character to indicate the current directory as the target

**sysadmin@localhost:~$** rm hosts

**sysadmin@localhost:~$** ls

**Desktop Documents Downloads Music Pictures Public Templates Videos**

**sysadmin@localhost:~$** cp -v /etc/hosts .

`/etc/hosts' -> `hosts'

**sysadmin@localhost:~$** ls

**Desktop Downloads Pictures Templates** hosts

**Documents Music Public Videos**

 following commands to copy from the source directory and preserve file attributes by using the -p option:

**sysadmin@localhost:~$** rm hosts

**sysadmin@localhost:~$** ls

**Desktop Documents Downloads Music Pictures Public Templates Videos**

**sysadmin@localhost:~$** cd /etc

**sysadmin@localhost:/etc$** ls -l hosts

-rw-r--r-- 1 root root 150 Jan 22 15:18 hosts

**sysadmin@localhost:/etc$** cp -p hosts /home/sysadmin

**sysadmin@localhost:/etc$** cd

**sysadmin@localhost:~$** ls -l hosts

-rw-r--r-- 1 sysadmin sysadmin 150 Jan 22 15:18 hosts

**sysadmin@localhost:~$**

The first copy with the -p option preserved the original timestamp. Recall that the tilde ~ represents your home directory (/home/sysadmin).

The second copy specified a different filename (newname) as the target. Because it was issued without the -p option, the system used the current date and time for the target; thus, it did not preserve the original timestamp found in the source file /etc/hosts.

Finally, note that you can remove more than one file at a time as shown in the last rm command.

**sysadmin@localhost:~$** rm hosts

**sysadmin@localhost:~$** ls

**Desktop Documents Downloads Music Pictures Public Templates Videos**

**sysadmin@localhost:~$** cp -p /etc/hosts ~

**sysadmin@localhost:~$** cp hosts newname

**sysadmin@localhost:~$** ls -l hosts newname

-rw-r--r-- 1 sysadmin sysadmin 150 Jan 22 15:18 hosts

-rw-r--r-- 1 sysadmin sysadmin 150 Jan 22 16:29 newname

**sysadmin@localhost:~$** rm hosts newname

**sysadmin@localhost:~$**

To copy all files in a directory, use the -R option:

**sysadmin@localhost:~$** mkdir Myetc

**sysadmin@localhost:~$** cp -R /etc/udev Myetc

**sysadmin@localhost:~$** ls -l Myetc

total 0

drwxr-xr-x 1 sysadmin sysadmin 32 Jan 22 16:35 **udev**

**sysadmin@localhost:~$** ls -lR Myetc

Myetc:

total 0

drwxr-xr-x 1 sysadmin sysadmin 32 Jan 22 16:35 **udev**

Myetc/udev:

total 4

drwxr-xr-x 1 sysadmin sysadmin 56 Jan 22 16:35 **rules.d**

-rw-r--r-- 1 sysadmin sysadmin 218 Jan 22 16:35 udev.conf

Myetc/udev/rules.d:

total 8

-rw-r--r-- 1 sysadmin sysadmin 306 Jan 22 16:35 70-persistent-cd.rules

-rw-r--r-- 1 sysadmin sysadmin 1157 Jan 22 16:35 README

**sysadmin@localhost:~$**

To remove a directory, use the -r option to the rm command;

rmdir command can also be used to delete directories, but only if the directory is empty (if it contains no files).

Also note the -r option. This option removes directories and their contents recursively.

**sysadmin@localhost:~$** ls

**Desktop Downloads Myetc Public Videos**

**Documents Music Pictures Templates**

**sysadmin@localhost:~$** rm -r Myetc

**sysadmin@localhost:~$** ls

**Desktop Documents Downloads Music Pictures Public Templates Videos**

**sysadmin@localhost:~$**

| **Linux Command** | **Description** |
| --- | --- |
| touch premove | Creates an empty file called premove |
| mv premove postmove | This command “cuts” the premove file and “pastes” it to a file called postmove |
| rm postmove | Removes postmove file |

touch premove

**sysadmin@localhost:~$** ls

**Desktop Downloads Pictures Templates** premove

**Documents Music Public Videos**

**sysadmin@localhost:~$** mv premove postmove

**sysadmin@localhost:~$** ls

**Desktop Downloads Pictures Templates** postmove

**Documents Music Public Videos**

**sysadmin@localhost:~$**

CH9:

Use the following tar command to create an archive of the /etc/udev directory. Save the backup in the ~/mybackups directory:

**ysadmin@localhost:~$** cd

**sysadmin@localhost:~$** mkdir mybackups

**sysadmin@localhost:~$** tar -cvf mybackups/udev.tar /etc/udev

tar: Removing leading `/' from member names

/etc/udev/

/etc/udev/udev.conf

/etc/udev/hwdb.d/

/etc/udev/rules.d/

**sysadmin@localhost:~$** ls mybackups

**udev.tar**

**sysadmin@localhost:~$**

The tar command is used to merge multiple files into a single file. By default, it does not compress the data.

The -c option tells the tar command to create a tar file. The -v option stands for "verbose", which instructs the tar command to demonstrate what it is doing. The -f option is used to specify the name of the tar file.

Display the contents of a tar file by using the available options (t = list contents, v = verbose, f = filename):

**sysadmin@localhost:~$** tar -tvf mybackups/udev.tar

drwxr-xr-x root/root 0 2018-07-19 06:51 etc/udev/

-rw-r--r-- root/root 153 2018-04-20 16:55 etc/udev/udev.conf

drwxr-xr-x root/root 0 2018-04-20 16:55 etc/udev/hwdb.d/

drwxr-xr-x root/root 0 2018-04-20 16:55 etc/udev/rules.d/

**sysadmin@localhost:~$**

o create a tar file that is compressed use -z option: The -z option makes use of the gzip utility to perform compression.

**sysadmin@localhost:~$** tar -zcvf mybackups/udev.tar.gz /etc/udev

tar: Removing leading `/' from member names

/etc/udev/

/etc/udev/udev.conf

/etc/udev/hwdb.d/

/etc/udev/rules.d/

**sysadmin@localhost:~$** ls -lh mybackups/

total 16K

-rw-rw-r-- 1 sysadmin sysadmin 10K Dec 9 23:24 **udev.tar**

-rw-rw-r-- 1 sysadmin sysadmin 306 Dec 9 23:26 **udev.tar.gz**

**sysadmin@localhost:~$**

Extract the contents of an archive. Data is restored to the current directory by default:

cd mybackups

**sysadmin@localhost:~/mybackups$** ls

**udev.tar udev.tar.gz**

**sysadmin@localhost:~/mybackups$** tar -xvf udev.tar.gz

etc/udev/

etc/udev/udev.conf

etc/udev/hwdb.d/

etc/udev/rules.d/

etc/udev/rules.d/70-persistent-cd.rules

etc/udev/rules.d/README

**sysadmin@localhost:~/mybackups$** ls

**etc** **udev.tar udev.tar.gz**

**sysadmin@localhost:~/mybackups$** ls etc

**udev**

**sysadmin@localhost:~/mybackups$** ls etc/udev

**hwdb.d rules.d** udev.conf

**sysadmin@localhost:~/mybackups$** ls etc/udev/rules.d

70-persistent-cd.rules README

**sysadmin@localhost:~/mybackups$**

To add a file to an existing archive, use the -r option to the tar command. Execute the following commands to perform this action and verify the existence of the new file in the tar archive:

**sysadmin@localhost:~/mybackups$** tar -rvf udev.tar /etc/hosts

tar: Removing leading `/' from member names

/etc/hosts

**sysadmin@localhost:~/mybackups$** tar -tvf udev.tar

drwxr-xr-x root/root 0 2018-07-19 06:51 etc/udev/

-rw-r--r-- root/root 153 2018-04-20 16:55 etc/udev/udev.conf

drwxr-xr-x root/root 0 2018-04-20 16:55 etc/udev/hwdb.d/

drwxr-xr-x root/root 0 2018-04-20 16:55 etc/udev/rules.d/

-rw-r--r-- root/root 172 2018-12-09 21:17 etc/hosts

**sysadmin@localhost:~/mybackups$**

In the following examples, you will use gzip and gunzip to compress and uncompress a file. Execute the following commands to compress a copy of the words file:

**sysadmin@localhost:~/mybackups$** cp /usr/share/dict/words .

**sysadmin@localhost:~/mybackups$** ls -l words

-rw-r--r-- 1 sysadmin sysadmin 938848 Jan 25 07:39 words

**sysadmin@localhost:~/mybackups$** gzip words

**sysadmin@localhost:~/mybackups$** ls -l words.gz

-rw-r--r-- 1 sysadmin sysadmin 255996 Jan 25 07:39 **words.gz**

**sysadmin@localhost:~/mybackups$**

Linux provides a large number of compression utilities in addition to gzip/gunzip. Each of them has pros and cons (faster compression, better compression rates, more flexible, more portable, faster decompression, etc.).

The gzip/gunzip commands are very popular in Linux, but you should be aware that bzip2/bunzip2 are also popular on some Linux distributions. It is fortunate that most of the functionality (the way you run the commands) and options are the same as gzip/gunzip.

**sysadmin@localhost:~/mybackups$** ls -l words.gz

-rw-r--r-- 1 sysadmin sysadmin 259983 Dec 10 18:18 **words.gz**

**sysadmin@localhost:~/mybackups$** gunzip words.gz

**sysadmin@localhost:~/mybackups$** ls -l words

-rw-r--r-- 1 sysadmin sysadmin 971578 Dec 10 18:18 words

**sysadmin@localhost:~/mybackups$**

Using bzip2 and bunzip2 to compress and uncompress a file is very similar to using gzip and gunzip. The compressed file is created with a .bz2 extension. The extension is removed when uncompressed. Execute the following commands to compress a copy of the words file:

**sysadmin@localhost:~/mybackups$** ls -l words

-rw-r--r-- 1 sysadmin sysadmin 971578 Dec 10 18:18 words

**sysadmin@localhost:~/mybackups$** bzip2 words

**sysadmin@localhost:~/mybackups$** ls -l words.bz2

-rw-r--r-- 1 sysadmin sysadmin 345560 Dec 10 18:18 **words.bz2**

Execute the following commands to uncompress the words.bz2 file:

**sysadmin@localhost:~/mybackups$** ls -l words.bz2

-rw-r--r-- 1 sysadmin sysadmin 345560 Dec 10 18:18 **words.bz2**

**sysadmin@localhost:~/mybackups$** bunzip2 words.bz2

**sysadmin@localhost:~/mybackups$** ls -l words

-rw-r--r-- 1 sysadmin sysadmin 971578 Dec 10 18:18 words

Using xz and unxz to compress and uncompress a file is also very similar to using gzip and gunzip. The compressed file is created with a .xz extension. The extension is removed when uncompressed. Execute the following commands to compress a copy of the words file:

**sysadmin@localhost:~/mybackups$** ls -l words

-rw-r--r-- 1 sysadmin sysadmin 971578 Dec 9 23:35 words

**sysadmin@localhost:~/mybackups$** xz words

**sysadmin@localhost:~/mybackups$** ls -l words.xz

-rw-r--r-- 1 sysadmin sysadmin 198756 Dec 9 23:35 **words.xz**

**sysadmin@localhost:~/mybackups$**

Execute the following commands to uncompress the words.xz file

**sysadmin@localhost:~/mybackups$** ls -l words.xz

-rw-r--r-- 1 sysadmin sysadmin 198756 Dec 9 23:35 **words.xz**

**sysadmin@localhost:~/mybackups$** unxz words.xz

**sysadmin@localhost:~/mybackups$** ls -l words

-rw-r--r-- 1 sysadmin sysadmin 971578 Dec 9 23:35 words

Use the zip command to compress the words file:

**sysadmin@localhost:~/mybackups$** zip words.zip words

adding: words (deflated 73%)

**sysadmin@localhost:~/mybackups$** ls -l words.zip

-rw-rw-r-- 1 sysadmin sysadmin 260119 Dec 9 23:48 **words.zip**

**sysadmin@localhost:~/mybackups$**

Compress the /etc/udev directory and its contents with zip compression:

**sysadmin@localhost:~/mybackups$** zip -r udev.zip /etc/udev

adding: etc/udev/ (stored 0%)

adding: etc/udev/udev.conf (deflated 24%)

adding: etc/udev/hwdb.d/ (stored 0%)

adding: etc/udev/rules.d/ (stored 0%)

**sysadmin@localhost:~/mybackups$** ls -l udev.zip

-rw-rw-r-- 1 sysadmin sysadmin 771 Dec 9 23:50 **udev.zip**

**sysadmin@localhost:~/mybackups$**

To view the contents of a zip archive, use with the -l option with the unzip command:

**sysadmin@localhost:~/mybackups$** unzip -l udev.zip

Archive: udev.zip

Length Date Time Name

--------- ---------- ----- ----

0 2018-07-19 06:51 etc/udev/

153 2018-04-20 16:55 etc/udev/udev.conf

0 2018-04-20 16:55 etc/udev/hwdb.d/

0 2018-04-20 16:55 etc/udev/rules.d/

--------- -------

153 4 files

**sysadmin@localhost:~/mybackups$**

To extract the zip archive, use the unzip command without any options. In this example we first need to delete the files that were created in the earlier tar example:

**sysadmin@localhost:~/mybackups$** rm -r etc

**sysadmin@localhost:~/mybackups$** unzip udev.zip

Archive: udev.zip

creating: etc/udev/

inflating: etc/udev/udev.conf

creating: etc/udev/hwdb.d/

creating: etc/udev/rules.d/

CH10

Use the redirection symbol > along with the echo command to redirect the output from the normal output of stdout (to the terminal) to a file. The cat command can be used to display file contents and will be used in this example to verify redirected output to the file. Type the following:

**sysadmin@localhost:~$** echo "Hello World"

Hello World

**sysadmin@localhost:~$** echo "Hello World" > mymessage

**sysadmin@localhost:~$** cat mymessage

Hello World

**sysadmin@localhost:~$**

When you use the > symbol to redirect stdout, the contents of the file are first destroyed. Type the following commands to see a demonstration:

**sysadmin@localhost:~$** echo "Greetings" > mymessage

**sysadmin@localhost:~$** cat mymessage

Greetings

**sysadmin@localhost:~$**

You can avoid clobbering a file by using >> instead of >. By using >> you append to a file. Execute the following commands to see a demonstration of this

**sysadmin@localhost:~$** cat mymessage

Greetings

**sysadmin@localhost:~$** echo "How are you?" >> mymessage

**sysadmin@localhost:~$** cat mymessage

Greetings

How are you?

**sysadmin@localhost:~$**

The find command is a good command to demonstrate how stderr works. This very flexible command allows searching with a host of options such as filename, size, date, type and permission.

The find command will begin the search in the directory specified and recursively search all of the subdirectories. For example, to search for files beginning in your home directory containing the name bash:

**sysadmin@localhost:~$** find ~ -name "\*bash\*"

/home/sysadmin/.bash\_logout

/home/sysadmin/.bashrc

**sysadmin@localhost:~$**

**sysadmin@localhost:~$** find /etc -name hosts

/etc/hosts

find: '/etc/ssl/private': Permission denied

**sysadmin@localhost:~$**

To redirect stderr (error messages) to a file, issue the following command:

**sysadmin@localhost:~$** find /etc -name hosts 2> err.txt

/etc/hosts

**sysadmin@localhost:~$** cat err.txt

find: `/etc/ssl/private': Permission denied

**sysadmin@localhost:~$**

You can also redirect stdout and stderr into two separate files.

**ysadmin@localhost:~$** find /etc -name hosts > std.out 2> std.err

**sysadmin@localhost:~$** cat std.err

find: `/etc/ssl/private': Permission denied

**sysadmin@localhost:~$** cat std.out

/etc/hosts

To redirect both standard output (stdout) and standard error (stderr) to one file, first redirect stdout to a file and then redirect stderr to that same file by using the notation 2>&1.

**sysadmin@localhost:~$** find /etc -name hosts > find.out 2>&1

**sysadmin@localhost:~$** cat find.out

/etc/hosts

find: '/etc/ssl/private': Permission denied

**sysadmin@localhost:~$**