Linux

mthree

Introduction to Linux

Module 5: Things we can do with files

ALUMNI
BY MTHREE CONSULTING

Introduction to Linux

mthree

In this module we will go into more detail on interactions with files and things we can do with them.

We will particularly focus on being able to read files as well as searching them and being able to copy and move things around the filesystem.

Reading Files

mthree

Being able to see the contents of a file is a basic part of any job. However, in a command-based system like Linux, we can't simply click on a file to open it.

There are a couple of commands we will look at for reading the contents of a file.

Command	Comments
cat filename	(concatenate) is frequently used in Linux – it will read the data from the file and give the contents as output
cat filename1 filename2	Output multiple files contents to screen
cat –n filename	This will output with line numbers
cat filename1 > newfile	This introduces the > tool as well, this will display the contents of filename1 and put it into a newfile
cat filename1 >> newfile	This introduces the >> tool – the contents of filename1 will be appended to the end of newfile

Reading Files continued

mthree

Command	Comments
more filename	This allows you to view a file in a scrollable format. (You can use space to scroll one screen at a time, while enter scrolls one line at a time. Using / in more will allow you to search for something specific.)
cat filename | more	Here we introduce the | - an extremely useful tool to pipe the output of one command into another – this is how we string multiple commands together
cat filename | less	Similar to more but with easier navigation (down key and enter move down one line, up key moves up one line, space bar down one page . / will search downwards, ? Will search upwards)
man cat > ~/catOutput.log	Another example here of using the > command to put the output into another file
man less >> ~/catOutput.log	Here we are appending the results of running the man command on less to the catOutput.log file

Capturing output

mthree

Let's pull together some of the fundamentals we have learnt so far to capture output.

Command	Comments
TODAY=`date +%Y.%m.%d`	We will go into the use of date later, but this will set today's date into the variable TODAY
LOGFILE=~/$USER.man.$TODAY.log **echo $LOGFILE**	Set the variable LOGFILE to be something specific to the USER and with today's date in it
man cat > $LOGFILE	Redirect the output of the man command into the logfile
cat $LOGFILE | less	Allow us to view the contents of the file using less

Copying files

Being able to copy files and use them in different places is a key Linux skill to learn.

cp logfile logfile.20200430	Here we take a copy of the logfile and rename it to a file with a date extension – this is a super useful technique to back up a file as you know exactly when it is from. This will put the file into the current location
cp file1 file2 file3 logs/.	Here we copy multiple files into the logs directory. Notice we introduce the "." here used to specify the exact location you are pasting something
cp *.log logs/.	We can also use wildcards when running the copy command – here we copy all files with extension .log into the logs directory
cp –R logs /var/tmp/.	If you want to copy a directory to another location, you will need to use the –R switch the recursively copy everything in that directory over to your new location.

Moving files and directories

mthree

Sometimes you may wish to move a file/directory rather than copying it. Make sure that your application does not need this file/directory for any reason before you move it. It also is an easy way to rename files.

Command	Comments
mv logfile logfile.old	Here we move the logfile to logfile.old
mv log1 log2 log3 logs/.	Here we move multiple files into the directory logs
mv *.log logs/.	Here we use wildcards to move all files ending in .log into the logs directory
mv logs oldlogs **mv dir1 dir2 dir3 logs/.**	Moving a directory follows the same syntax as moving files – the same is also true if you want to move multiple directories
mv –n file1 file2 logs/.	The –n option will make sure that you do not overwrite any files if they already exist – you will effectively ignore them

Links

The ln command in Linux allows us to create hard and soft links in the environment.

Command	Comments
ln target.txt link.txt	Here we create a hard link (link.txt) to the target.txt file. Editing link.txt will change the data in target.txt as you are effectively touching the same data.
ln –s fixGenerator.sh fixGen	Using the –s option, we can create a soft link to the target file. You will see it displayed differently when you run ls after creating the link: [ec2-user@ip-172-31-40-21 ~]$ ls -l total 84 lrwxrwxrwx 1 ec2-user ec2-user 15 Apr 29 19:31 fixGen -> fixGenerator.sh Here we see it has a -> showing the link. You can then run commands on your soft link and it will refer to the target you are pointing to.

Searching within files

We have covered the use of / within tools such as less, but sometimes you may want to search within a file and put the results into another command. This is often called “grepping”.

grep ERROR logfile.log	Here we “grep” for the word ERROR in our example logfile.
grep –c ERR* logfile.log	Here we “grep” and ask for the count of the number of times we see something starting with ERR appear in our example logfile.
grep -R 'passwd' /etc	Here we search recursively though the etc directory in any files we have permission to read
grep -v WARN logfile.log	Here we show everything in the log file that is not a WARN
grep -i WARN logfile.log	Here we print out everything with WARN in it but ignore the case

Head and Tail

mthree

Head and tail are two commands that will allow you to show only a subset of a file

Command	Comments
head logfile	By default, head with no options will return the first 10 lines of the file
head –n 30 logfile	To set the specific number of lines you want to return use the –n option
head file1 file2 file3	This will display the first 10 lines of each file (in a nice format out to screen)
tail logfile	This will output the last ten lines of the logfile to standard out
tail –n 40 logfile	The –n option allows you to specify the exact number of lines you wish to return
tail file1 file2 file3	This will display the last 10 lines of each file in a nice format to the screen
tail –f logfile | grep ERROR	The –f option will watch for updates to a file and print them out as they come