cat and vim

In the current directory we have one "special" file, testfile. This file is a copy of syslog file and contains a lot of information. Enough, to see different ways to read the file.

Let's check if the file is available.

ls -l testfile

First, we will use this opportunity to learn a new command. cat can print all files, but it is not necesairly smart to print binary files with cat. It is usable for text files. So, let's learn if the file really *is* the text file.

file testfile

cat

The most basic and I think, mostly used command is cat. As usual in linux the name of the file is a kind of contraction of longer word, in this case - concatenate.

Ok, now we can read the file.

cat testfile

Well... Yeah, the file was printed, but the case here is that the file is fully printed. From the start to the end. We will see how to avoid this a little bit later.

vim and view

Let's be honest. The true linux admin uses vim. Or even more, the true admin uses vi :) most modern distributions contains vi, vim (in fact in most cases, vi is replaced by vim), emacs, nano... here are many text editors available.

It is important to understand one thing. cat just prints the file. vi is fully functional, very powerful text editor.

Before we run vim, let's learn new, very important thing. Please, try to guest, what is this?

dwfi4t0gjvg0eg5jqh58wgpoweh4w8ghbo;witheaigh

It is what one does when incidentally opens vi and doesn't have any idea how to quit.

Right, in order to exit vim, press ESC and then :q - what means quit.

If you changed something in the file and you don't want to save the changes, press :q!.

If you want to save these changes, press :wq - write and quit.

By using ESC we switch between modes in vim. One is **INSERT** mode. In this mode we can edit the file. Second is **COMMAND** mode, where we can interact with vim and do actions. Like save and quit. Third mode is **VISUAL**. in this mode we can do selection of text.

In most cases, when we enter vim, it starts in COMMAND mode.

Ok, let's see our file with vim.

vim testfile

We will touch vim in the lab later. Now it is enough to say that you can navigate using arrows (mostly :)). When you want to quit, please press ESC and then :q!

So, this is vim.

We have another possibility to work with vim. We can run it in *read only* mode:

view testfile

In order to quit, please execute - :q.

# Less and more

It is time to learn two new commands to print file content. We already know cat. The downside of cat is that this command prints the whole file in one shot. It is rather inconvenient to go through large files in this way.

## more

The first command which we can use in order to have better experience is more.

more testfile

Now, we are in somehow interactive mode and we are able to navigate through file. However more has it's own limitations. We can go forward only, using enter. In order to quit, press

q

## less

There is a more sophisticted tool to read the file. It is called less. This command gives us better navigation. We can go back and forth using the same keys like in vi, moreover, we can search for strings, using / sign.

less testfile

Now, please try to go through the file. Down and upwards.

/Resolution

This searches through the file for Resolution string. Remember, linux is case-sensitive!

By n

and

N we go through search results, down and upwards.

In order to quit, press q

## Pipes

What is interesting, we can use cat, more and less together with pipes.

cat testfile | more

cat testfile | less

# Print part of the file

Despite the possibility of printing the whole file and navigate through it, we can print part of the file. We have two possibilities.

## head

It we want to print first lines of the file, we can use head. By default head shows 10 lines.

head testfile

If we wish to see different number of lines, we can use -n argument and pass the number for lines.

Let's try with 2 lines

head -n2 testfile

And now for 22 lines:

head -n22 testfile

## tail

tail does exactly the same thing as heead does, but from the end of the file.

tail testfile prints 10 last lines.

tail -n2 testfile will print 2 last lines, and

tail -n22 testfile will show 22 lines.

similar like with less and more, both head and tail can be used with pipes.

Let's see on the examples. Please try to understand what you will see before you execute the commands!

cat testfile | tail -n1

cat testfile | head

cat testfile | head -n4 | tail -n2