Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section:\_\_\_\_\_\_\_\_\_\_\_

**Topics: Feeling at home. Ease of work. Resources? - Resources!**

This lab exercise is to be submitted **at the end** of the lab session!

*Today we want to obtain much more confidence about our machine, understand what we do, who and where we are, how the files and directories are organized, and some more functions relevant for the administration of the system.*

Once your machine has started, type **pwd** and note the result:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This command always indicates where within the file system you are.

You can as well see that the users' home directories are all under /home. You can always go to **your own home** by typing

**cd**

This command returns the prompt, **$**, as last symbol. This Dollar-sign means that the system is waiting for your input. In front of the prompt, you can see the **hostname**, so that you always know in front of which machine you sit, followed by the directory in which you currently are. The **~** is the abbreviation for your home directory. If the command 'cd' is followed by a directory, it will change directory to the target directory. Try this out with

**cd /usr/bin**

The prompt now looks like

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

You can see that the username is followed by '@' and the name of the host, then by the current directory, finally by the prompt ($) itself.

Review question: Which three **commands** do you use to retrieve the three parts of the prompt? :

Name of the user: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of the machine: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Current directory: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The command that gets you back to your home directory: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The users have all their files in their respective directories under /home. All the configuration and settings are located in /etc. The command to go to the /etc directory is:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Check with **pwd** that you are there (in /etc)!

In order to see the **list** of files in a directory, you type '**ls**'. What you get, is a large number of names of files, but nothing else. That is not very helpful. Therefore, we usually use '**ls -l**', where the 'l' stands for 'long' and the dash ('-') indicates that you add an **option**. Try this command, **ls -l**, in /etc, and note the last two entries:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The details that you can see have not been discussed yet, but you are able by now to navigate the system, go to other directories and back home easily.

As one example for the configuration files, when you do ls -l in /etc, you see a file with the name 'timezone'. I think it is obvious, which setting this file is responsible for? [Please give your own opinion. The correctness is not considered at marking.]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

You can confirm this by looking at the contents of this file. One of many commands to view the contents of a file is 'cat'. Try it out: **cat timezone**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In case you mistakenly just typed 'cat' without any **file name** behind it, you will see an empty line. This is the sign that the program waits for your input. You can just stop or close the program by using **Ctrl-C** (Ctrl is the key lowest to the left on your keyboard. Press 'C' at the same time).

Let us try what happens when you ask for the contents of a file that does not exist:

**cat timezoneeze**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

You see, there is no need to be afraid of the machine!

[Good system administrators actually are never afraid of a machine, but they are always the masters of their machines, they are fully in command!]

Now, go back to your home directories. The command use is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Create a new directory named ‘mydir’ using this command:

**mkdir mydir**

Check the directory either it have been created or not. The command use is :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Change into that directory:

**cd mydir**

As to be expected, there are no files in this directory. Let us create some files. We noticed earlier, that touch is a possibility. Therefore, type

**touch file1**

**touch file2.txt**

When you type

**date**

the date will be displayed on the screen.

Now try

**date > file1**

**ls –l**

**cat file1**

What is the output?:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What do these do? You might have already guessed it, it **redirects** the output into a **file**, with the name of 'demo'.

*Do not confuse it with our earlier 'pipe', or '|'.*

*The pipe takes the output and sends it to another* ***command****; as input for that next command.*

The redirection takes the output and sends it to a device, a file, etc.

Create new directory name ‘mybackup’. Copy the file1 to the mybackup using this command

**cp file1 mybackup**

ls -l

Now you can see that mybackup have a file1 as a new file.

In the course of our labs, we have done a lot of things, and maybe you don't remember exactly what you did, which command you used. Do not despair! All is not lost! The command 'history' shows you everything that has been typed by you until now, also in the earlier sessions. Try it out: **history**. There are some commands beyond the top of the screen, and now invisible. Here we have a little helper, the **pipe** '|'. The idea had occurred Dennis Ritchie, when he had to plug a number of hoses together, to water the most remote end of his garden. He decided that we should be able to 'pipe' the output of one command into another one, and so forth. The command 'more' is specific for displaying only one single screen at a time. So with **history | more** we 'pipe' (or send) the output of the command 'history' into another command, that displays only one screen at a time. By using the 'blank'- or 'space'-key you can now scroll through your earlier commands and see them.

Whatever you wrote here, it will probably work, but we can do better: use the command 'less' instead. In 'less' you can scroll up and down with the cursor keys. Try it out, and write the 10th command that you entered during your 'history' here: **history | less**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

And now scroll up to your second command, which was:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

With 'q' you can always leave 'less' whenever you like.

======================================================

*Next, we as system administrators need to monitor the resource use of our system: How many are loogon on, how much memory is used, etc.*

The first means of choice is 'top'. It gives us a regular update of our main system resources. Issue

**top**

and watch the show. The display is updated once per 5 seconds. We don't need to understand everything yet, but we can already make out some relevant information from the top 5 lines. Have courage and **guess**, what is shown in these 5 lines: [Give your opinion, based on what you can see, about the 5 topmost lines]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

('q' gets you out of the display.)

Another relevant information are the times of starting and halting the machine, the userlog- on details, etc. This is given by the command 'last'. At the prompt, type

**last**

and indicate the **last, most recent**, 'reboot' (take care here, check the date and time):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Unfortunately, the display of 'last' is a bit messy: It shows (re-)boots and logins mixed.

But that does not matter, because we have the pipe, and we have 'grep'. Grep looks for the occurrence of a term in a line. If the term is there, the line is displayed. If the term is not there, the line is suppressed. Try it out:

**last | grep boot**

displays all the lines with 'boot' in them, so you get all the (re-)boot occurrences.

**last | grep tty**

displays all the login times [tty are the consoles, compare lab exercise 2] Using grep, we can easily separate the different entries.

Another relevant resource is of course space on the hard disk. The command here is

**df**

But when you look at the output, it is not very helpful: It gives the sizes in blocks of 1kByte. However, with the correct option 'h' for 'human readable', the display makes much more sense; because it displays M(egaByte) and G(igaByte):

**df -h**

Still, there are some pseudo-file-systems, that are in memory instead of the hard disk. You can recognize them by the 'none' as mount point [compare lab exercise 1, installation and setting of mount points]. We can use grep again, to suppress all those lines that have no sda in them [compare lab exercise 1, partitioning]: **df -h | grep sda**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Looking carefully, you notice that not only all the lines with 'none' have disappeared, but also the first line that contained the explanation of the columns. Luckily, we have a command as well to just show the first lines: 'head'. Here we only need one line, so we can use

**df –h | head -n 1**

followed by

**df -h | grep sda**

to see the two outputs one after the other. If you are a person with aesthetic underpinnings, you might find this 'just not beautiful'. Okay, I'll help you:

**cd**

**df -h | head -n 1 > demo**

**df -h | grep sda >> demo**

**cat demo**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Can you guess, what has happened here?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We have used so many commands until here, and you might have forgotten one or another. Most of all, you might not remember the options. Luckily, for all commands, there is a complete set of explanations available in the form of the so-called **man-pages**.

So if you require some explanations of a command, and know the options, you best type the command 'man' followed by the command that you want some extra information about. For example, the command 'head' can be examined by

**man head**

('q' gets you out of the display.) Try the man pages of some other commands..

*Do not forget to shut down your machine; like we did at the end of lab exercise 2, close Virtualbox, and log out.*

*It is still suggested that you export your installation after the shutdown for reasons of security. Don't forget to submit your lab sheet.*