

LW-02. LINUX/UNIX COMMAND LINE BASICS.

1. TARGET.

Get the initial skills of working with a Linux/UNIX command line via bash shell. The Linux/UNIX commands are used:

```
adduser, userdel, passwd, sudo, logout, poweroff, apt,  
man, apropos, help info, whatis, whereis, type  
who, whoami, id, uname, date, clear, ps, df, banner, cal, bc, echo,  
pwd, cd, ls, mkdir, rmdir, rm, mv, ln, ln -s, du,  
touch, cat, cp, mv, rm, more, tail, head, ls, file, type, find.
```

2. ASSIGNMENTS.

(NOTE. Start Your **UbuntuMini** Virtual Machine on your VirtualBox. You need only Linux Terminal to complete the lab tasks.

Before completing the tasks, make a SnapShot of your Virtual Linux, if there are problems, you can easily go back to working condition!)

2.0. Generate You Variant Nr.

2.1. User account management basic. **(make a Screenshot 1)**

2.2. Use apt Package Manager on Ubuntu Command Line. **(make a Screenshot 2)**

2.3. Command line structure. (Read only)

2.4. Use the man, info, help. (Read only)

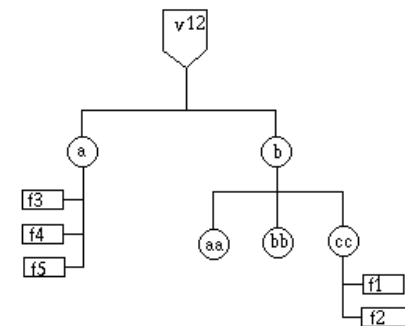
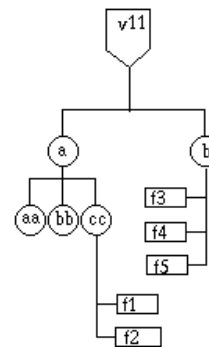
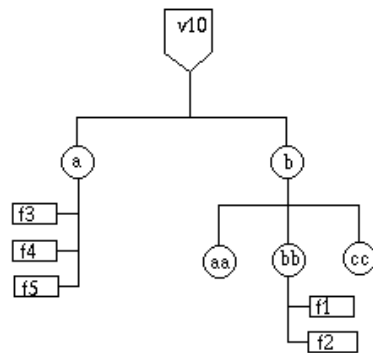
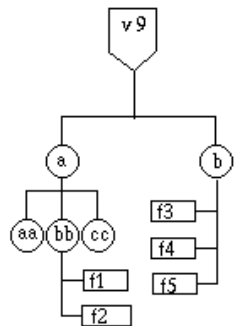
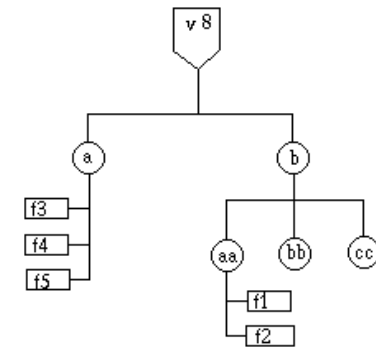
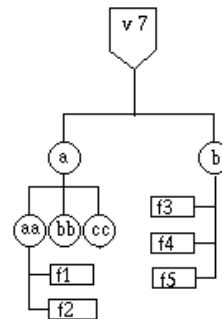
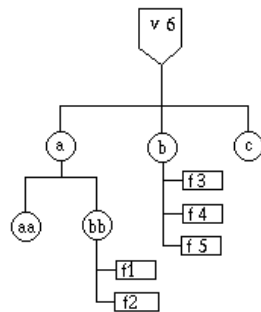
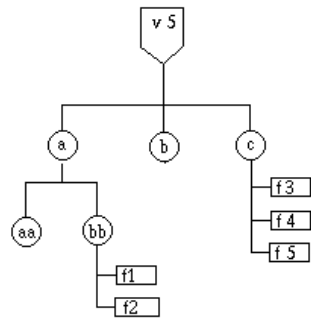
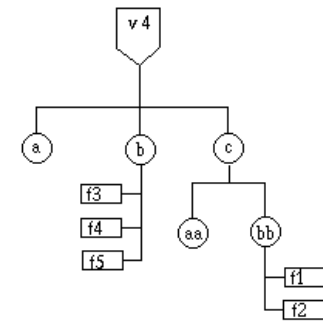
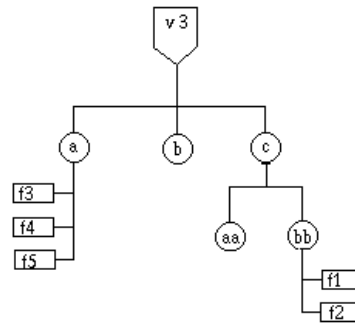
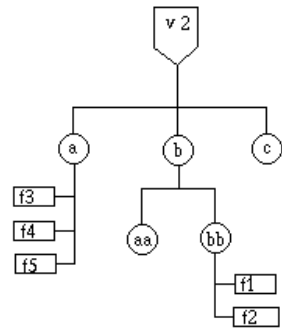
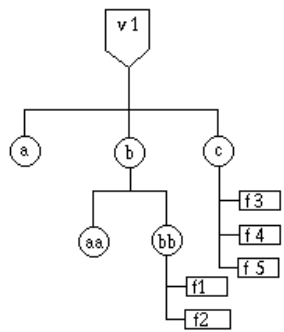
2.5. System commands. **(Fill in a Table 1)**

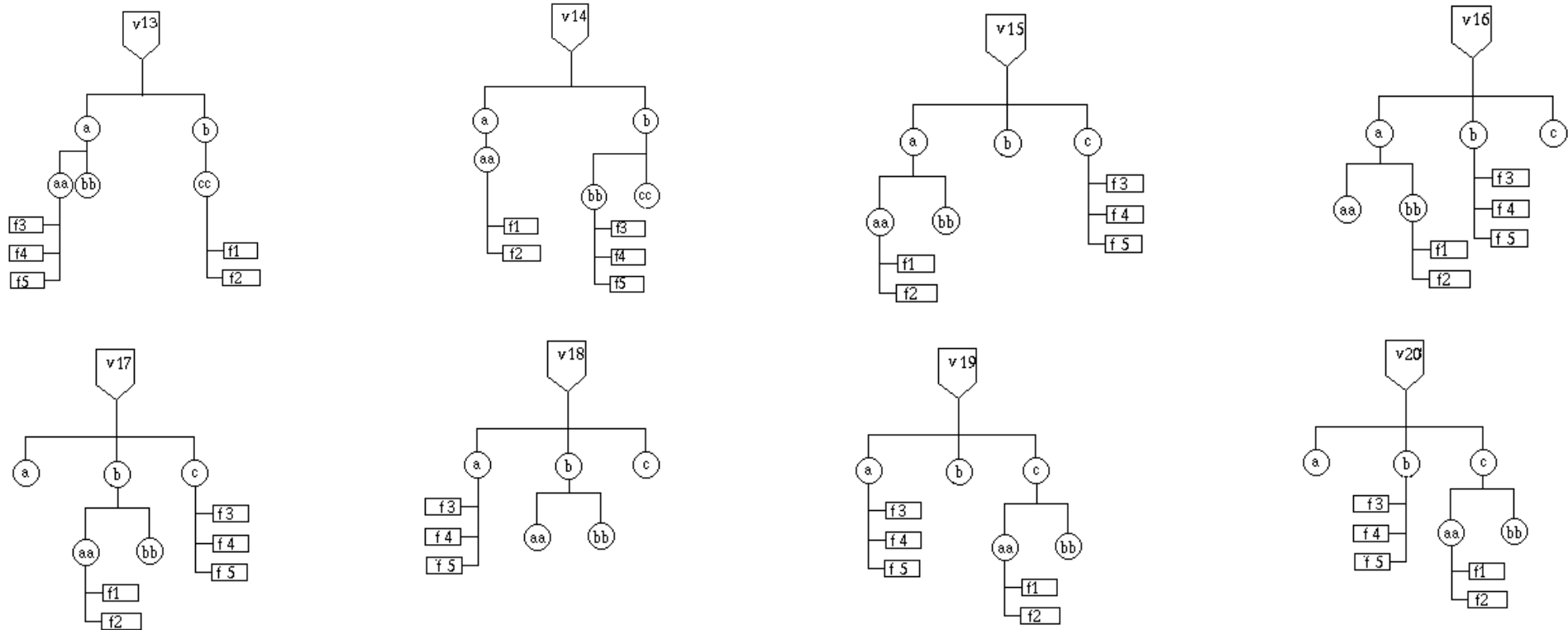
2.6. Directory commands. **(Fill in a Table 2)**

2.7. File commands. **(Fill in a Table 3)**

2.8. Create a bash-script **report-vN**. **(make a Screenshot 3), (make a Screenshot 4)**

VARIANTS OF THE STRUCTURE = VN, where N –variant Number generated from Your Name.





3. REPORT.

The report is provided electronic form with Report Blank Form. Make Screenshots 1-4 and Tables 1-3.

LW-02 REPORT. LINUX/UNIX COMMAND LINE BASICS.

Student Name Surname	Student ID (Variant Nr)	Date

Insert: Variant Generation, Screenshot 1, Screenshot 2, Table 1, Table 2, Table 3, Screenshot 3, Screenshot 4.

4. GUIDELINES.

4.0. GENERATE OF YOU VARIANT NR.

a) Write your LastName in the letters of the English alphabet. Must be at least 7 letters, if not enough, then add the required number of letters from the FirstName (if not enough, then repeat LastName and FirstName).

For example, for Yurijs Li there will be LIYURIJS.

b) Replace the first 7 letters with their ordinal numbers in the alphabet.

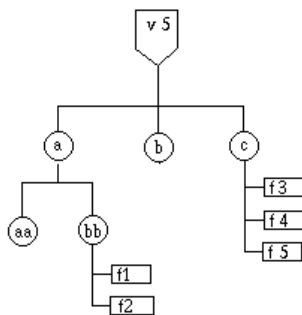
For example, 12 09 25 21 18 09 10.

c) Consistently add modulo 20 these 7 numbers and add 1.

For example, $(12 + 09 + 25 + 21 + 18 + 09 + 10) \bmod 20 + 1 = 104 \bmod 20 + 1 = 4 + 1 = 5$

d) The resulting will be your variant Nr.

For example, Variant Nr = 5 (odd Variant Nr).



4.1. USER ACCOUNT MANAGEMENT BASIC.

4.1.1. Create new user account, example stud. Use **adduser** command. (make a Screenshot 1 for adduser)

```
$ sudo adduser stud
```

(NOTE. You can use the command “**userdel -rf stud**” to delete stud account from your Linux. **Do not do it now!**)

- Logout from student account (**logout**) and login as stud.
- Use the **passwd** command for change password for stud.
- Test execution of command **sudo -i poweroff** for stud (not sudo user).
- Logout from stud account (**logout**) and login as student.
- Test execution of command **sudo -i poweroff** for student (sudo user).

4.2. USE APT PACKAGE MANAGER ON UBUNTU COMMAND LINE.

When you begin to use the Ubuntu operating system, the first choice of installing software is through the graphical Ubuntu Software manager.

Although many programs are available in this collection, you will soon know that it lacks many powerful utilities especially that for the command line. Then is the time to switch to the more powerful software installation utility, the apt-get. With apt you can:

- Listing available and installed packages
- Update the package repositories
- Upgrade installed software
- Search for available packages
- Install a packages
- Remove a software from your system

4.2.1. List the software packages on Ubuntu (all and installed).

```
$ sudo apt list
```

```
$ sudo apt list --installed
```

4.2.2. Update the package repositories with apt.

A repository is a list of available software packages in a given snapshot in time. Since the packages in this list are sometimes added, removed and updated, you need to update your system repositories from time to time. The lists of available packages are fetched from the location(s) specified in `/etc/apt/sources.list`.

It is best to update the repositories before installing a new software by running the following command as root:

```
$ sudo apt update
```

4.2.3. Upgrade installed software with apt.

Option `upgrade` is used to install the newest versions of all packages currently installed on the system from the sources enumerated in `/etc/apt/sources.list`. Use the following command as root in order to upgrade software on your system:

```
$ sudo apt upgrade # Caution, there may be space problems with your Linux, first make SnapShot!
```

4.2.4. Search for available packages with apt (testing with bc)

```
$ apt search <package-name-keyword>
```

4.2.5. Install a package with apt (testing with bc)

```
$ sudo apt install <package-name>
```

4.2.6. Remove a software from your system (testing with bc)

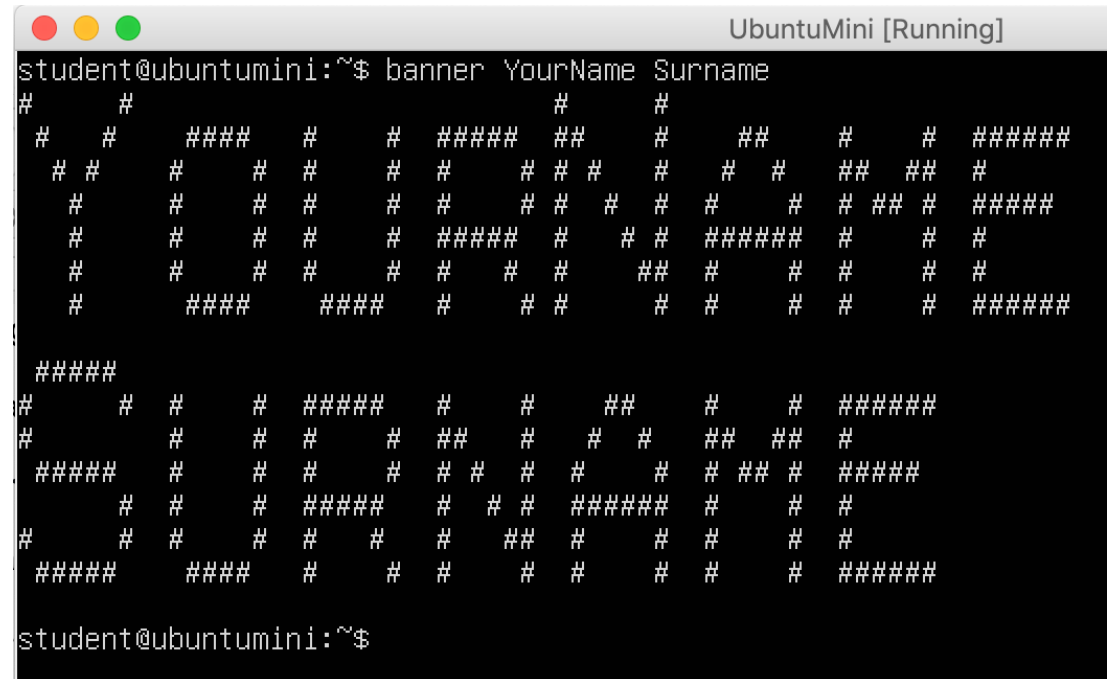
```
$ sudo apt remove <package-name>
```

4.2.7. Install `sysvbanner` package for banner command.

Login as a student. Use “`sudo apt install sysvbanner`”.

Test `banner` command work. (make a Screenshot 2)

```
$ banner YourName Surname
```



```
UbuntuMini [Running]
student@ubuntumini:~$ banner YourName Surname
# # # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # #
#####
# # # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # #
##### # # # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # #
##### # # # # # # # # # # # # # # # #
student@ubuntumini:~$
```

4.3. COMMAND LINE STRUCTURE.

4.3.1. Command line.

The command line may contain one or more commands listed through a semicolon.

Each command has a name and may have modifiers and arguments.

Command line structure (Bakus-Naur Form):

```
<command line>: = <command> [; <command>] ...  
<command>: = <name> [-<modifiers>] [<arguments>] [&]  
<modifiers>: = [<modifier> [ <modifier>] ...]  
<arguments>: = [<argument> [<argument>] ...]
```

Examples:

```
$ pwd                # no modifiers, no arguments  
$ cd /home           # have only 1 arguments  
$ cat f1 f2 f3       # have 3 arguments  
$ ps -aux            # have 3 modifiers
```

Example:

```
$ ls -al -u file1 dir1 & # Remark string
```

Here:

```
$ - command line prompt (equal $PS1 (Prompt String 1)  
ls (from listing - list) - the name of the command;  
-a -l -u – 3 modifiers (-alu variant);  
file1 and dir1 are arguments;  
& - sign indicates the background execution of the team.  
# - after this sign is a remark string (for script notation)
```

All these elements are **lexemes** of the command.

Space, Tab, NewLine - **separators** between lexemes.

4.3.2. Name and path of the file (or directory)

- Files are identified by name. These names usually somehow reflect the content.
- File names can contain any characters (except the slash “/” character) and are limited to **255 characters** in length.
- Importantly, all file and command names are **case sensitive**.
- File names can contain a dot at any position, for example, the name “...” is acceptable.
- Files starting with a dot is **hidden** in UNIX and are not used in many commands without special instructions (compare \$ ls ; ls -a).
- UNIX systems have a standard File System Hierarchy (**FSH**) directory structure.
- FSH is a directory tree starting with the root directory “/”.
- Each user has his own separate directory (specified in the /etc/passwd file) as his **home directory**. (\$ more /etc/passwd)
- Usually the user's home directory is under the /home directory and is called the username, for example /home/student.
- At the initial login, your home directory automatically becomes your **working directory**.
- You can access a file through its **absolute name**, or, based on your location, through a **relative name**.

- The following **synonyms** can be used for directories:

~ (tilde)- home,

. (dot) - current

.. (dot-dot) - parent directories.

Example,

```
$ cd /home/student      # absolute name
$ cd abc                 # relative name
$ cd ~/bin               # relative name
$ cd ../../usr/sbin     # relative name
```

To read/write/execute commands and move to some directories, you must have permissions.

Example,

```
$ cd /root               # Testing "Permissions denied" message
```

4.3.3. File (Directory) Name Generation.

The following **wildcards** can be used for file/directories name generation:

? - any non-empty character in the given position of the template, For example, ***.???** - all files ending with a "dot and three symbol";

***** - any number of allowed characters, including empty. For example, ***** - all files except those starting with a period;

[list-range] - any one of the characters specified in the list or range, for example: **[-abk3]** or **[A-Za-z0-9_]**;

[^ list-range] - any of the characters not specified in the list, for example: **[^ -abk3]** or **[^ A-Za-z0-9_]**.

4.4. USE THE MAN, INFO, HELP.

Every UNIX has manual pages. The man help system contains documentation for various system commands, resources, and configuration files.

<p>The UNIX Reference Manual contains 8 sections:</p> <ol style="list-style-type: none">1. Commands;2. System calls;3. Subroutines;4. Special files - descript. sys-files;5. File format and convention;6. Games;7. Macro packages and language conventions;8. Administrator commands and procedures.	<p>The description of the command consists of the following sections:</p> <ul style="list-style-type: none">• NAME- name and function;• SYNOPSIS- syntax;• DESCRIPTION- description of the function;• OPTIONS- description of options;• SEE ALSO- related teams;• FILE- used files;• DIAGNOSTIC- answers to errors;• BUGS- noticed incorrectness;• IMPLEMENTATION - who did it.
<p>Navigation Commands in man:</p> <ul style="list-style-type: none">• <q> - exit from the manual (quit);• <space>, <f>, <number> - scrolling forward (forward);• - scrolling the manual back (back);• </pattern> - search (search) for the pattern forward;• </> - search by the previous pattern of the next coincidence;• <?> - search by the previous template of the previous coincidences.	<p>Man call:</p> <p>Use man [i] <name> where</p> <p>i is the section number, if i is omitted, i = 1 is implied;</p> <p><name> is the name of the command.</p> <p>For example,</p> <pre>\$ man ls</pre>
<p>Exercises 1. Testing:</p> <pre>\$ man man \$ man 7 signal \$ man passwd \$ man 5 passwd</pre> <p>Exercises 2. Learn commands yourself:</p> <p>apropos, help info, whatis, whereis, type.</p>	<pre>LS(1) FSF LS(1) NAME ls - list directory contents SYNOPSIS ls [OPTION]... [FILE]... DESCRIPTION List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuSUX nor --sort. -a, --all do not hide entries starting with etc</pre>

4.5. SYSTEM COMMANDS.

Exercises. You must learn and use the following system commands yourself:

```
$ who; whoami; id, uname; date; clear; ps; df; banner; cal; bc; echo
```

Table 1. Examples of using system commands. **Fill in the table and add it to your Report Table 1.**

Note 1. To save space, several logically related commands are written compact on one line and divided (;)

Nr	Commands	Your short interpretation of command string (odd commands Nr for odd Variant Nr; even commands Nr for even Variant Nr)
1	<code>who -m</code>	
2	<code>whoami</code>	
3	<code>who -q</code>	
4	<code>who -l</code>	
5	<code>id; id -u; id -un</code>	
6	<code>id; id -g; id -gn; id -G</code>	
7	<code>uname; uname -s</code>	
8	<code>uname; uname -v</code>	
9	<code>uname -v</code>	
10	<code>uname -i</code>	
11	<code>date +%d%m%Y; date +%s</code>	
12	<code>date +%A; date +%H%M\ %p</code>	
13	<code>clear; ps;</code>	
14	<code>ps -aux</code>	
15	<code>df /dev/sda1</code>	
16	<code>clear; df;</code>	
17	<code>banner `date +%Y`</code>	
18	<code>banner `two words`</code>	
19	<code>cal; cal -y; cal -y 1</code>	
20	<code>cal; cal -y 9999; cal -j</code>	

For Linux fans.

```
$ echo "1234567890^1234"| bc
$ echo "scale=10; 4*a(1)" | bc -l
$ echo "scale=1000; 4*a(1)" | bc -l
$ echo "scale=10000; 4*a(1)" | bc -l # calculate very LONG time !!
```

Note. The bc calculator supports up to **23,860,929 significant digits** (applause for UNIX). To exit the calculator, use the quit command.

4.6. DIRECTORY COMMANDS.

Exercises. You must learn and use the following directory commands yourself:

```
$ pwd; cd; ls; mkdir; rmdir; rm; mv; ln; ln -s; du; type
```

Table 2. Examples of using directory commands. **Fill in the table and add it to your Report Table 2.**

Note 1. To save space, several logically related commands are written compact on one line and divided (;)

Nr	Commands	Your short interpretation of command string (odd commands Nr for odd Variant Nr; even commands Nr for even Variant Nr)
1	<code>pwd</code>	
2	<code>cd /; pwd</code>	
3	<code>cd ~; pwd</code>	
4	<code>cd /home/vasja; pwd</code>	
5	<code>cd ../../usr; pwd</code>	
6	<code>cd; pwd</code>	
7	<code>ls; ls -u; ls -t;</code>	
8	<code>ls -l; ls -i; ls -a;</code>	
9	<code>ls -d; ls -r; ls -R; ls -F</code>	
10	<code>mkdir ~/letters</code>	
11	<code>cd ~/letters; pwd</code>	
12	<code>mkdir a b c; ls</code>	
13	<code>rmdir a; cd c; pwd; rmdir ../b</code>	
14	<code>cd; rm -r letters</code>	
15	<code>mkdir a b c; ls -il</code>	
16	<code>ln -s a e; ls -il</code>	
17	<code>mv c a; ls -il; ls -l a</code>	
18	<code>cd ~ ;du</code>	
19	<code>cd ~; du -a; du -a /</code>	
20	<code>cd ~; rm -r *</code>	

4.7. FILE COMMANDS.

Exercises. You must learn and use the following file commands yourself:

```
$ touch; cat; cp; mv; rm; more; tail; head; ls; ln; file; type; find
```

Table 3. Examples of using file commands. Fill in the table and add it to your Report Table 3.

Nr	Commands	Your short interpretation of command string (odd commands Nr for odd Variant Nr; even commands Nr for even Variant Nr)
1	<code>touch letter0</code>	
2	<code>> letter1 <Ctrl>+<D></code>	
3	<code>cat > letter2 <Ctrl>+<D></code>	
4	<code>cat letter1</code>	
5	<code>more letter1 letter2</code>	
6	<code>pr letter2</code>	
7	<code>cp /etc/passwd .</code>	
8	<code>tail passwd</code>	For example. Show the last lines of the passwd file (10 lines by default)
9	<code>tail +20 passwd</code>	For example. Show the last 20 lines of the passwd file
10	<code>head passwd</code>	
11	<code>head -5 passwd</code>	
12	<code>cat > masha</code>	
13	<code>ln masha dasha; ls -il masha dasha</code>	
14	<code>rm masha; ls -il</code>	
15	<code>ls -il</code>	
16	<code>ln -s dasha masha; ls -il</code>	
17	<code>rm dasha; ls -il</code>	
18	<code>cp letter1 dasha; ls -il</code>	
19	<code>file masha dasha</code>	
20	<code>file passwd</code>	
21	<code>file /usr/bin/sh .. letter0</code>	
22	<code>type file; type *</code>	
23	<code>find / -name passwd</code>	
24	<code>find ~ -size 10</code>	
25	<code>find / -perm 4000</code>	
26	<code>find /usr -type d</code>	
27	<code>find / -user vasja; find / print</code>	
28	<code>find / -group root -user 0</code>	
29	<code>find . -mtime -7 -exec ls -l {} \;</code>	
30	<code>find /home -name *.bak -exec rm {} \;</code>	

4.8. CREATE A "REPORT-VN" BASH-SCRIPT FILE.

4.8.1. Install **tree** command package on your Linux Virtual Machine in the classroom or at home.

Login as a student. Use the **sudo apt install tree** command to install tree package on your system. Test tree command work (`$ tree /`)

4.8.2. Create script-file report-vN (use **nano** or **vi** or **mc-editor**).

The report-vN creates your version (**vN = Your Variant Nr**) of the directory-file structure and performs the following actions:

1. On your home directory makes directory vN (**mkdir**).
2. On directory vN makes sub-directory structure a, b, c, aa, bb, cc (**mkdir**).
3. Creates a file f1 containing your Last Name and First Name (**echo**> f1).
4. Creates a calendar for the year and month of your birth, the result is written to the file f2 (**cal**> f2).
5. Combines the received files f1 and f2, and writes the result to the file f3 (**cat** f1 f2> f3).
6. Creates a hard link f4 to the file f3 (**ln** f3 f4).
7. Counts the number of lines, words, characters in the file f2, and the result is written to the file f5 (**wc** f2> f5)

4.8.3. Make report-vN file executable (**chmod**). To run report-vN, use (**./**):

```
$ chmod +x report-vN
$ ./report-vN
```

4.8.4. Recursively delete invalid files and directories with the command **rm**:

```
$ rm -rf vN
```

4.8.5. Make sure that the structure you created matches the task (**ls -R** or **tree**):

```
$ tree vN
```

4.8.6. Add a report-vN script content to the report: **(make a Screenshot 3)**

```
$ cat report-vN
```

4.8.7. Add a screenshot of the tree command output to the report: **(make a Screenshot 4)**.

```
$ tree vN
```