**Shell Worksheet**

In this exercise, you will learn and explore the command line interface in UNIX. First start the shell. For this open the Dash (press on icon in upper left corner) and type Terminal. Click the resulting Icon. This is the Terminal, running the **bash** shell. A shell can execute and combine the commands you enter.

First, familiarize yourself with the following commands. For this have a look at the following resources and write a short explanation of what each command does:

* Manual pages (manpages): For example: type **man ls** into the shell (the command line) to see the description for **ls**
* Google: You can just google ‘shell ls’ or ‘man ls’ (or other keywords)
* Or see the following comprehensive links:
  + <http://linuxcommand.org>
  + <http://community.linuxmint.com/tutorial/view/100>
  + <http://arachnoid.com/linux/shell_programming.html>
  + <http://www.youtube.com/watch?v=QGvvJO5UIs4>
  + <http://www.freeos.com/guides/lsst>

1. sudo and sudo apt-get install execute a command as another user
2. whoami displays your effective user ID as a name.
3. man format and display the on-line manual pages
4. echo writes any specified operands, separated by single blank(` ') characters and followed by a newline (`\n') character, to the standard output.
5. Pwd return working directory name
6. cd change directory
7. Ls list directory contents
8. Cp copy files
9. Rm remove directory entries
10. Mv move files
11. Mkdir make directories
12. History view all the previous executed commands.
13. Cat utility reads files sequentially, writing them to the standard output.
14. Grep searches any given input files, selecting lines that match one or more patterns.
15. Cut cut out selected portions of each line of a file
16. Sort sort or merge records (lines) of text and binary files
17. Wc word, line, character, and byte count
18. Sleep suspend execution for an interval of time
19. Nano Nano's ANOther editor, an enhanced free Pico clone
20. Gedit To open a specific file you can specify the filename after the gedit command
21. Vi is a text editor that is upwards compatible to Vi. It can be used to edit all kinds of plain text. It is especially useful for editing programs.
22. Date display or set date and time
23. Ps displays a header line, followed by lines containing information about all of your processes that have controlling terminals.
24. kill, killall, xkill sends a signal to the processes specified by the pid operands.
25. ifconfig (type /sbin/ifconfig)is used to assign an address to a network interface and/or configure network interface parameters.
26. Ping is used to assign an address to a network interface and/or configure network interface parameters.
27. ln and ln –s creates a new directory entry (linked file) which has the same modes as the original file.
28. Ssh is a program for logging into a remote machine and for executing commands on a remote machine.

Next, try to do the following tasks. Record the shell command/commands that you used to do them.

1. Write a hello-world program in C and save it in a file called **hello.c** Compile **hello.c** (using **gcc**) and test your program (default name is **a.out**) by running it.What is the shell command to show the message “Hello world”?
2. gcc -o hello hello.c
3. ./hello
4. How do you print out the following:

Hello world!

Not too cold today.

1. How do you write the lines from #2 into a file called **output01.txt**?

echo "Not too cold today" > /Users/zuldyz/Desktop/output01.txt

1. How do you show the contents of **output01.txt**?

cat /Users/zuldyz/Desktop/output01.txt

1. Add the current time and date to the **output01.txt.**

Date >> /Users/zuldyz/Desktop/output01.txt

1. Add your name to the end of **output01.txt**.

echo 'Dana Zhumabekova' >> /Users/zuldyz/Desktop/output01.txt

1. Sort **output01.txt** into **output02.txt**.

sort output01.txt > /Users/zuldyz/Desktop/output02.txt

1. How do you show only the lines from **output01.txt** which have the string “lo” in them?

grep 'lo' output01.txt

1. Count the number of lines and words in **output01.txt**. Verify that this is the same in **output02.txt**.

wc -lw output01.txt

1. Create a folder called **mywork** and move your 2 output files into it.

mkdir mywork

mv output01.txt mywork

mv output02.txt mywork

1. *Link* the file **output01.txt** to **output01-01.txt**.

ln output01.txt output01-01.txt

1. Delete **output01.txt**

rm output01.txt

1. Look at **output01.txt**, what is strange?
2. Copy **output02.txt** to **output02-01.txt**.

cp output02.txt output02-01.txt

1. Link *symbolically* **output02-01.txt** to **output-link.txt**.

ln -s output02-01.txt output-link.txt

1. Delete **output02-01.txt**.

rm output02-01.txt

1. Look at **output-link.txt** – what happened?

*It cannot be open*

1. Write a line of shell code which counts (prints the numbers) from 1 to 3 and makes a 5 seconds break between each number.

echo 1; sleep 5; echo 2; sleep 5; echo 3

1. Start the counter (from 18) in the background and then issue a command to print “Hello world” directly afterwards.

MacBook-Air-Zuldyz:mywork zuldyz$ echo 1; sleep 5; echo 2; sleep 5; echo 3

1

echo '2

Hello, 3

MacBook-Air-Zuldyz:mywork zuldyz$ echo 'Hello, World'

Hello, World

MacBook-Air-Zuldyz:mywork zuldyz$

1. Do the previous task but write the outputs of both commands to **output03.txt**.
2. Open **gedit** in the background. Kill it by looking its ID up with **ps** and then issuing the **kill**

command.

kill ps -l | grep ol

1. Repeat last step but try using **killall** and **xkill** instead.
2. Look up the IP addresses of your and your neighbor’s computers, and **ping** one another (interrupt **ping** with ctrl-C).
3. Let the output of 10 pings to your neighbor be written in the file **output04.txt**.

ping 10.1.136.141 -c 10 > outpu04.txt

1. Write the history of all your commands to **myhistory.txt**.

history > myhistory.txt