# Real Time Operating Systems

## Getting familiar with Unix system

## The Unix file system and directory commands:

cd

ls

ls -l

ls -a

ls -1

man ls

mkdir rtos

ls -l

cd rtos

cd ..

rmdir laboratory1

cd /

ls -l

cd /etc

ls

ls -ltr

ls -l \*.conf

ls -l \*.?

ls -ld \*.?

ln

## File operations:

cd $HOME

touch empty\_file

cp empty\_file second\_empty\_file

mv second\_empty\_file second\_file

ls -l

rm empty\_file second\_file

echo "Be the change that you wish to see in the world." > quote.txt

cat quote.txt

for i in {1..300}; do echo "Well done is better than well said." >> quotes.txt; done

wc -l quotes.txt

less quotes.txt

Internal commands of the 'less' program:

SPACE - forward one screen

b - backward one screen

j - forward one line

k - backward one line

G - go to end of file

g - go to beginning of file

h - help: summary of commands

/string - search forward

n - search for next instance

?string - search backward

N - search for previous instance

q - quit

## File permissions:

cd $HOME

echo test > test\_file

chmod -r test\_file

ls -l test.txt



1. What do the r w x characters mean?
2. What does the term "filemon" and "students" mean here?
3. Why is the first character a '-'?

cat test\_file

chmod u+r test\_file

ls -l test\_file

cat test\_file

chmod go-rwx test\_file

ls -l test\_file

1. Ask your neighbor to create a file on their desktop and place their biggest secret there. Check if you can read your neighbor's secret. Check if you can modify your neighbor's secret. Secure the file.
2. Create a file with the solution to the task and change its permissions so that your neighbor can modify it.

## Operations on processes:

ps

ps -e

ps -ef

ps -ef | less

ps -el

man ps

top  
 kill pid

kill -9 pid

## Gathering information:

who

finger  
 last

last | less

## Scripts (bash):

1. Write a script that will output the name of your favorite game. Grant the file executable permissions and run it using ./script.sh or sh script.sh
2. Write a script that will list the files located in the directory passed as an argument ($1)
3. Modify the above script to perform the task for all the provided folders in the arguments.
4. Try this code

a='png extension must be removed.png'

b="${a%.[Pp][Nn][Gg]}"

echo $a

echo $b

## Run and compile C++ programs

1. Write a program in C++ that prompts the user for a number and displays a triangle made of asterisks.

Compile and run the program:

g++ program.cpp -o program

./program

## Find program:

1. Try a find command to locate in the /etc directory:

* files up to 1kB in size
* created within the last 3 days
* no larger than 250 kB
* created not less than a month ago but not more than three months ago (for simplicity: 30 and 90 days ago).
* combine the search criteria

## Tar program:

1. Get familiar with the tar program and practice

* archiving a few files and a directory structure
* restoring both the entire archived structure (including back to the original location in the file system, as well as to another location)

1. Check if files and directories have been completely restored.
2. Pay attention to how files/directories specified with a full, absolute path are saved in the archive, and how those specified with a local name are saved:

* Ask your neighbor to create an archive and verify the permissions of the restored files.

## special files

1. Get familiar with the 'mknod' command and create a special file with it, such as a named pipe.
2. Try writing something to this file (cat > file) or reading from it (cat file).
3. Open several terminal windows, write to the named pipe in some of them, and read from it in others. What happens to the texts written to the named pipe?

## Text editors / pdf readers

1. Check which editors and pdf readers are available in the laboratory

gedit

gvim

emacs

code

xpdf

mupdf

zathura

evince

atril

ocular

acroread

## Other usefull commands:

grep

gzip / zip

top / htop

xargs

sed

## 