

# Your programming environment: introduction to *Linux/Unix OS*. Your first program

Log into your Linux account. The username and the password are student. Start the terminal.

Table 1: Basic Linux terminal commands

Command	Usage
<b>ls</b>	short listing of directory contents
<b>cd</b>	navigate to directories hierarchically above or below
<b>pwd</b>	print working directory
<b>mkdir</b>	make directories
<b>rmdir</b>	remove an existing directory
<b>cp</b>	copy files
<b>rm</b>	delete files
<b>mv</b>	move or rename files
<b>emacs, vi, mcedit</b>	text files editors
<b>man</b>	prints the specified manual page

More details how to use these and more commands can be found for example here:  
<http://www.math.bas.bg/~nkirov/2012/NETB101/unixbasic.html>  
Most of the terminal commands accept parameters, for example:

```
cp [options]... Source Dest
cp [options]... Source... Directory
```

Using the command `man cp` you can view the manual page which contains all the information about the usage, parameters and options of the command `cp`.

**Exercise 1.** Using some of the commands above, create a directory in an appropriate location which will be your working directory for this class.

**Exercise 2.** Check whether the *The GNU Compiler Collection (GCC)* is installed on your system by simply typing `gcc` in the command line. If the command is not recognized, install GCC by typing:

```
sudo apt-get install build-essential
```

**Exercise 3.** Using a text editor of your choice (for example mcedit) type the following C++ program and save it in your working directory under the filename `hello.cpp`:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     cout << "Hello , World!" << endl;
8     return 0;
9 }
```

Listing 1: `hello.cpp`

**Exercise 4.** Compile the program in Listing 1. The command `g++` is the front end for C++ of the GCC compiler collection:

```
g++ hello.cpp
```

It will produce the default binary `a.out` which can be executed by typing

```
./a.out
```

The result of the execution of the program will be the appearing of the message `Hello , World!` in the terminal. If you want to specify the name of the binary produced by the compiler you have to pass the name to the compiler as an option:

```
g++ -o hello hello.cpp
```

Then you can run the program by typing

```
./hello
```

## Problems

1. Create another directory in your working directory and copy the file `hello.cpp`. Then modify the program text so it will be typed on a single line. Try to compile and run the modified program. Is it working correctly? **C++ has *free-form layout*.**
2. Modify the code in Listing 1 so it will print a message of your choice.
3. Write a program which asks the user for his first name and then prints a greeting, for example: `Hello, John!`
4. What is *integrated development environment* (IDE) and what is the difference between using an IDE and the development process, described in this workshop?