C++ Programming for CE
C++PCE
Winter Term 2014/2015

Assignment sheet 0 - Basics, Tools & Introduction

Assignments that are marked with *StudOn submission* are **mandatory** and must be submitted via StudOn in time – please check in the StudOn for the respective deadlines.

Getting Started

1 CIP-Account registration

If you have not already done so, please register an account for the computer science department CIP pools. The procedure can be found here:

`https://wwwcip.informatik.uni-erlangen.de/documentation/studentAccount.en.html`

Short summary (tl;dr):

- Login with the username cipan and the password cipan
- Your account will be created in the next few days

In order to verify your work to be submitted *it is curtial that you have an account as we will only accept solutions that actually do compile on any of the machines in room 01.155-113*

`https://wwwcip.informatik.uni-erlangen.de/cipPools/roomIndex.en.html#cip1`

Tools

2 Remote login via ssh

Although we requite your submissions to compile on any CIP1 system it is not necesaary to be physically presend in the room. Using a tool called ssh (secure shell) you can connect to the system from your computer at home, laptop or almost any computer that has an ssh client installed.

If you are using Windows as an operating system, there is a tool called PuTTY, which you can download here:

`http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html`

Linux and OS X based systems usually come with an ssh client installed by default. All following commands apply for Linux and OS X and have to be typed in a terminal.

Connecting to a remote host is as easy as typing the following command:

`ssh <user name>@<remote host>`

Where <user name> is a placeholder for your remote user name and <remote server> is either an IP address or a fully qualified host name (e.g., faui06a.informatik.uni-erlangen.de).

3 Using a command line text editor

In order to modify your source code remotely, you have to use a command line based editor (or copy the files from your local machine to the remote host).

We recommend to use vim for that purpose. You can find tons of documentation on vim online (http://www.vim.org) or use the built-in help function.

In a nutshell:

- Start vim by typing: `vim <file name>`
- Switch from command to insert mode by typing: 'i'
- enter, modify your file
- Switch back from insert to command mode by typing '<esc>'
- save your changes (in command mode) by typing: 'w'
- Save and exit vim (in command mode) by typing: 'x'
- Quit vim (in command mode) by typing: 'q'
- If you have made changes that you do not want to save: 'q!'

4 Compiling with gcc

Now that you know how to connect to the CIP machines from remote locations and how to use a command line text editor, you will write your first C++ program using these tools.

1. Connect to the computer to your left or right using ssh
2. Create a new folder (mkdir hello) and change into the folder (cd hello)
3. Create a new file hello.cpp using vim (vim hello.cpp)
4. Write the code for your first program using vim (see below)

```
#include <iostream>

int
main(){
  std::cout << "hello world!";
  return 0;
}
```

Now compile your program using g++ from the GCC-suite: `g++ hello.cpp`

The output will be a file named "a.out", which you can run by typing "`./a.out`". If you want to provide your own name for the binary, use the following command:

`g++ hello.cpp -o <output name>`

The compiler does quite a few things for this seemingly easy case:

1. Transforming your C++ source code into object code
2. Linking your code with functions from other libraries (e.g., iostream)
3. Generating a single binary executable file (a.out)

You can find out about most basic option of g++ by typing "`man g++`" which will get you to the so called man page. More information about the manpage system can be obtained via the command "man man" :)

5 Makefiles – more automation, less typing

The process of "software development" usually consists of continuous iterations of modifying and compiling your code. For this repetitive and boring process you should take advantage of the GNU make system.

For this purpose you simply place a file called "Makefile" in the source directory of your project. This file consists of so called make rules. Each rule has a unique name, also called make target, followed by a colon and a list of files or other rules it depends on. The next lines must be starting with a <tab> character (not blank spaces!) and contain the instructions to be carried out for the rule. Each rule may contain an arbitrary number of instructions.

Example:

```
all: hello

hello:
  g++ hello.cpp

clean:
  rm hello.o a.out

.PHONY: clean
```

6 Advanced ssh

In order to access remote systems even more conveniently you can use the so called public-private-key-authentication. For this method it is necessary to generate your individual key pair. On the local Linux system of the CIP machines this can be done via:

ssh-keygen -t rsa -b 4096

Now you should enter a password to protect your key. As a result you will now find two files in your local .ssh directory. One of them ends with a .pub extension, this is the public part of your key. In order to authenticate via keys, the public part of your key must be appended to a file called "authorized_keys", located in the .ssh directory of the respective system.

Thus, for a first test go through the following steps

1. cat /.ssh/<name of your key>.pub » /.authorized_keys

2. ssh to the computer left or right of you in the CIP pool
3. enter the password of your key

You should now be automatically logged into the other machine.

You may also use a tool under windows to generate your keys, yet we do not provide any support for this way.

Important information about your future submissions

As already mentioned in the first section, we want to stress again, that **we will only consider, correct and grade submissions that do compile on the machines of CIP1!** This means that either you really have to sign up for an account, or you feel very comfortable about your skills to submit code and Makefiles that compile on other systems.