Introduction to PHY 224

1 Introduction PHY 224 block course

The PHY 224 is a block course dedicated for teaching how to program in C/C++ language in a linux environment. The course will cover some of the basic tools available in C and C++. The outline of the course can be found below:

1. Introduction to programming in Linux.

2. Introduction to basics of programming.

3. Data processing in C/C+.+.

4. Data handling in C/C++.

5. Memory management in C/C++.

6. Handling large data in C and C++.

7. Object and Classes.

8. Scripting using ROOT 5 and 6.

9. Programming using ROOT 5 and 6 packages.

10. Parallel programming in C/C++.

Each lecture will be held online using the following timetable:

– 0900∼0915: From the main Zoom channel, the lecturer will provide introduction to and some examples of tools to be used in the following practical session.

– 0915∼1015: The students will divided in to 3 sections of "breakout rooms" in Zoom. In each breakout room, either a lecturer or a TA will be available for the students to ask questions. The students are expected to go through the exercises given in the lecture notes during this time.

– 1015∼1030: There will be a 15 minutes break before continuing in the main Zoom channel with the lecturer.

– 1030∼1045: From the main Zoom channel, the lecturer will provide introduction to and some examples of more tools to be used in the following practical session.

– 1045∼1145: The students will divided in to 3 sections of "breakout rooms" in Zoom. In each breakout room, either a lecturer or a TA will be available for the students to ask questions. The students are expected to go through the exercises given in the lecture notes during this time.

– 1145∼1200: The students may ask any of the remaining questions that they may have regarding the lecture materials or the exercises. The lecturer will return everyone to the main Zoom room and give the students a concluding remark and end the lecture for the day.

The course will start on 2020-08-17 and end on 2020-08-28. The lectures will be held on each weekday from 0900 until 1200 until the last day of the course.

1.1 Structure of the lecture notes

All of the lecture notes are designed to be used during the course. The lecturer will display the lecture notes as well as a linux terminal during the lecture to demonstrate some examples through Zoom. The students are expected to never copy-and-paste the materials in the lecture notes. The PDF has been encoded to be incompatible to be copied-and-pasted for programming purposes.

However the students are welcome type the given example codes letter-by-letter from the lecture notes. This is to encourage the students to write a proper program from scratch rather than copying-and-pasting.

Within the lecture notes students will find that the examples of codes and scripts have been colour-coded. There are three colours:

```
bashcommand
```

is used for demonstrating what to type into a linux terminal (commands).

```
1  bashscript
```

Is used for typing into a text editor and saved as a file to be executed from a linux terminal (scripts).

```
1  c++code
```

Is used for proper c++ code written into a text editor, saved then compiled into an executable and executed from the linux terminal (code).

More information of the above can be found in the first lecture note (https://www.uzh.ch/cmsssl/physik/dam/jcr:f7bb92d8-10e9-40c8-a065-4b51dea554a5/Lecture_1_SJL_CPP.pdf).

1.2 Course requirements

Obviously since the course is being held online through Zoom, the students must have access to a computer with internet access with Zoom installed. The students must also be able to access some of the local computers located physically in the physics institute through secure shell (ssh) remotely. You will find in the next section a set of instructions on how to use SSH from a computer with any linux operating system, Windows 10 and MacOS X. However if you have any questions regarding how to gain access to the machines in the physics institute please contact one of our lecturers Dr. Roland Bernet https://www.physik.uzh.ch/en/researcharea/lhcb/team/senior-researchers/bernet.html.

2 Secure shell (SSH)

Secure shell is a very basic tool available for many operating systems by default. SSH allows users to open up a "shell" or an encrypted enclosed environment remotely allowing users to log in and operate the computers remotely. However there are multiple versions and functions of SSH and for the purpose of this course, you need to be able to forward the X display. In linux environment, the X-server handles display by default (unless reconfigured) and forwarding X display is typically not an issue. However some external tools are required such as XMing (http://www.straightrunning.com/XmingNotes/) or XQuartz (https://www.xquartz.org/) for Windows and MacOS platforms. Below you will find instructions on how to use them. Please note, for the purpose of this course, only download the .exe and .dmg available on the course website https://www.physik.uzh.ch/de/lehre/PHY224/FS2020.html.

2.1   From Linux

As stated above, using SSH with X display forwarding is the most straight forward. You need to open a terminal and type in the following:

```
ssh -X username@remote.com.put.er
```

Obviously you need to change the "username" and "remote.com.put.er" with your actual username and the domain. For instance

```
~> ssh -X stelee@spinor12.physik.uzh.ch
The authenticity of host 'spinor12.physik.uzh.ch (130.60.165.162)' cant be
↪   established.
ECDSA key fingerprint is SHA256:x7ERYVt/uJs0Q44o4sBqo28r96ecHthfmrD2KW9q3pQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'spinor12.physik.uzh.ch,130.60.165.162' (ECDSA)
↪   to the list of known hosts.
Password: #Here, you will not see your password being typed.
Have a lot of fun...
stelee@spinor12:~>
```

To make sure that the X display is being forwarded correctly, you should try opening one of the simplest X applications.

```
xclock
```

You should be able to see the following:

Figure 1: X-clock from your display may appear in different colours.

If you can see the X-clock, your system is good to go.

2.2   From MacOS X

The MacOS X will allow you to SSH into a linux computer remotely by default from the terminal. However, MacOS X does not have a X server installed by default. This is where you have to install something external. The instructions are as follows.

1. Download and install XQuartz (https://www.physik.uzh.ch/dam/jcr:ee978d2d-2c55-4ad5-949f-aea05edd XQuartz-2.7.11.dmg).

2. Using a text editor add the following lines

```
1    XAuthLocation /opt/X11/bin/xauth
```

to

```
/etc/ssh/ssh_config
```

If done properly, above file should look as follows:

```
#     Tunnel no
#     TunnelDevice any:any
#     PermitLocalCommand no
#     VisualHostKey no
#     ProxyCommand ssh -q -W %h:%p gateway.example.com
#     RekeyLimit 1G 1h

Host *
        SendEnv LANG LC_*
XAuthLocation /opt/X11/bin/xauth
```

Figure 2: Your /etc/ssh/ssh_config should look similar to above.

3. Test your setup by connecting to a remote server and opening the X-clock.

```
ssh -X username@remote.com.put.er
```

Obviously you need to change the "username" and "remote.com.put.er" with your actual username and the domain. For instance

```
ssh -X stelee@spinor12.physik.uzh.ch
```

and once logged in

```
xclock
```

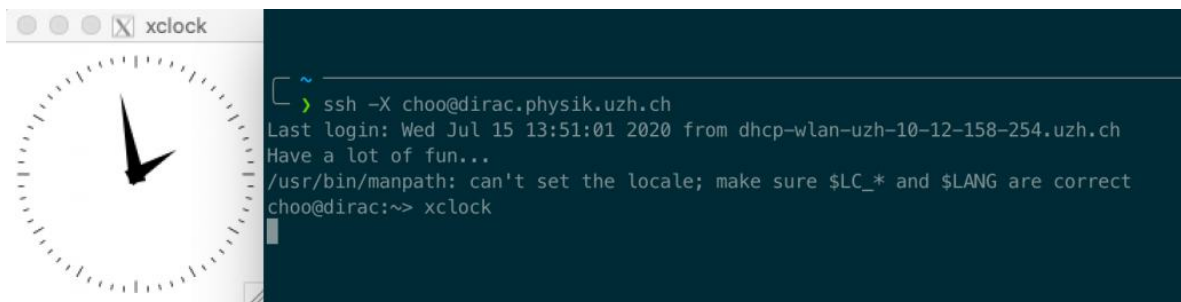You should see something similar to the following:

Figure 3: When you log into your own account in the physics server, your terminal output should show similar results. You should also use xclock command to check that the X display is being forwarded correctly.

If you can see the X-clock remotely, you are good to go. If you have any trouble getting X-server forwarding to work on MacOS X. Please contact our TA Kenny Choo (https://www.physik.uzh.ch/en/groups/neupert/team/choo.html).

2.3   From Windows 10

Windows 10 is a little bit trickier. You must install Xming and also enable Windows Subsystem for Linux, and install a linux operating system within the Windows 10 operating system. The instructions are as follows:

1. Download and install X-Ming (https://www.physik.uzh.ch/dam/jcr:0b8cc71b-7788-4a7c-b0d0-d47b228b
Xming-6-9-0-31-setup.exe)

2. Enable Windows Subsystem for Linux by following the red highlights in the following figure:
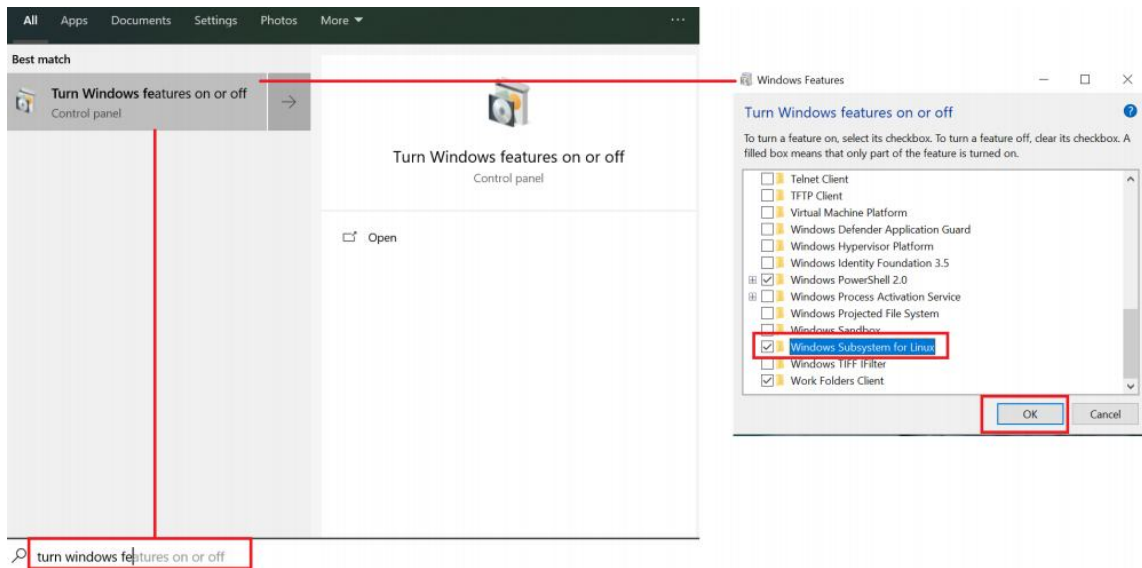

Figure 4: You must search for "Turn Windows features on or off" from Start menu, and click on "Turn Windows features on or off (Control Panel)". You need to then check "Windows Subsystem for Linux and hit OK

.

3. Windows will then install the feature and you must reboot your Windows 10 computer.

4. When the computer reboots, go to Microsoft store and install OpenSUSE LEAP 15.2 following the red highlights in the following figures
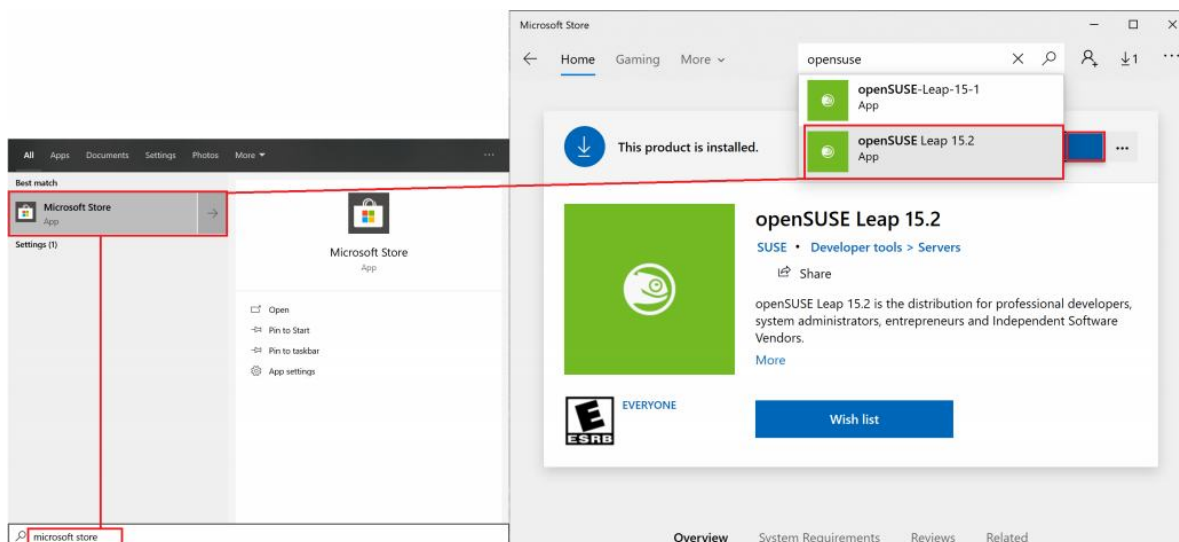

Figure 5: You must search for "Microsoft Store" on "Microsoft Store (App)". You then need to search for OpenSUSE Leap 15.2 from the Microsoft Store and install it.

5. Once the Microsoft store installs OpenSUSE Leap 15.2, open it and follow the instructions displayed to create a user, and initialize it.

6. once initialized, become the super user or "root" by typing in

```
su
```

and when you are prompted to do so, type in your password.

7. You then need to install a text editor called "emacs" using the following line.

```
zypper in emacs
```

and say yes when prompted for confirmation.

8. You then need to get out of super user by typing in

```
exit
```

9. You need to modify your default loading environment called ".bashrc" by the following.

```
emacs -nw ~/.bashrc
```

10. Add the following entry to the very bottom of your ".bashrc"

```
1  export DISPLAY=0:0
```

11. save and exit by pressing "Ctrl+x, Ctrl+s" then "Ctrl+x, Ctrl+c". If EMACs asks if you want to add a line at the end, or to confirm saving the file. Type in yes.

12. close your OpenSUSE LEAP 15.2 session by typing in:

```
exit
```
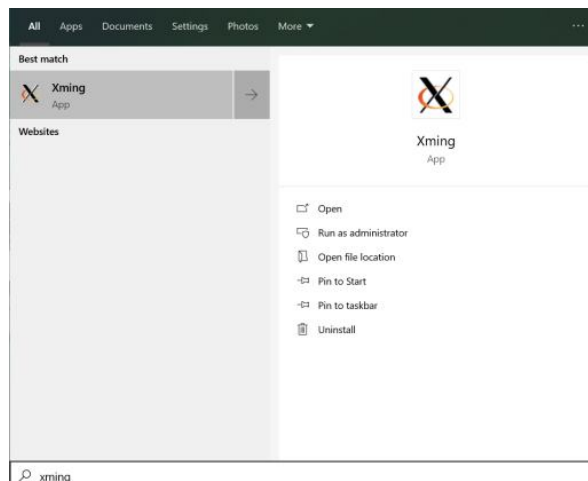
13. Turn on X-Ming from the start menu.

Figure 6: Search for "Xming" in start menu and click on Xming (App)
.

14. At this point you have linux installed within your Windows 10 operating system. You can now test it in the same manner as you could on linux by opening OpenSUSE Leap 15.2 application.

```
ssh -X username@remote.com.put.er
```

Obviously you need to change the "username" and "remote.com.put.er" with your actual username and the domain. For instance

```
~> ssh -X stelee@spinor12.physik.uzh.ch
The authenticity of host 'spinor12.physik.uzh.ch (130.60.165.162)' cant
↪  be established.
ECDSA key fingerprint is
↪  SHA256:x7ERYVt/uJs0Q44o4sBqo28r96ecHthfmrD2KW9q3pQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
↪  yes
Warning: Permanently added 'spinor12.physik.uzh.ch,130.60.165.162'
↪  (ECDSA) to the list of known hosts.
Password: #Here, you will not see your password being typed.
Warning: No xauth data; using fake authentication data for X11
↪  forwarding.
Have a lot of fun...
stelee@spinor12:~>
```

To make sure that the X display is being forwarded correctly, you should try opening one of the simplest x applications.

```
xclock
```

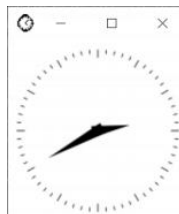You should be able to see the following:

Figure 7: X-clock from your display may appear in different colours.

Once you have xclock working from Windows 10 through Xming and OpenSUSE. You are good to go. Remember to turn on X-ming before you try to forward X display, otherwise nothing will be displayed.

3   Version Control

The students should note that the only reason that we are asking you to use the local machines in the physics institute is because we know the versions and the packages installed on these machines well.

All of the exercises and the programs for this course have been tested on the machines in the institute. However your local linux, MacOS X or Windows Subsystem Linux (WSL) can just as well handle everything locally from your computer.

This is very much encouraged since we have limited number of computers available at the institute.

This means that as long as you have the correct packages and correct versions of the packages installed in your local machine, you can do everything for this course without the use of SSH.

Please let us know as soon as possible that you would like to use your own local machine instead of the one at the institute remotely. Just make sure you check that your version of gcc and g++ are later than 7 by typing in:

```
gcc --version ; g++ --version
```

from your linux (or Unix[1], or WSL) terminal.

The students should also note that the 8th and 9th lecture will require them to use packages from ROOT (https://ph-root-2.cern.ch/). First, please note that "ROOT" is is different from the super-user "root" and that depending on your programming background, the installation of ROOT may not be trivial.

Students are encouraged try to install ROOT themselves in their local machines as there are many resources available online. However ROOT is already installed and available for use on the local machines at the institute.

Steven J. Lee, Roland Bernet, Kenny Choo   14. August 2020

___

[1]MacOS is Unix based