

Programming Assignment #1

Introduction to Computer Networks

Mindset

This is not going to be like your regular programming course where the instructor exhaustively covers the syntax of a programming language from do-while loop generation to I/O file access. Rather, you will be experiencing what every professional programmer has gone through at some point in life – trying to self-teach a programming language that's new and interesting/promising. For the professionals, the reason could be simply fun, but more so to stay competitive in the job market. For you, the reason is so that you see a little better that college-style teaching takes you a long way but it does not quite prepare you for the changes to come. Scared? Survive the fittest, Darwin says. Polly says, survive the dare most. One would not know if he/she fits not even trying.

Style

Polly understands that you are not quite yet a professional. Therefore, for each programming assignment (PA in short), there will be a pdf file specifying what you need to do/program and what output is expected. There will also be an accompanying lab session in video recording that demonstrates the use of APIs/commands to get you started.

Read the pdf file first. It will give you a rough feeling of what your code needs to achieve. Go ahead and watch the video next. Try and repeat Polly's moves demonstrating the APIs/commands. See for yourself what happens when you change the lines here and there. Then, ponder more seriously what your code needs to do and see if you can associate that with the APIs/commands you've just learned.

What will be more fun is the process of trial and error – this is also called the process of debugging. You will definitely be searching for example code, detailed description the API, Q&As, additional APIs and etc. This is going to take time and the duration varies from person to person. You want to start early if you are new to such troubleshooting style of learning to code. Google.com is already your friend when it

comes to information searching. You might find stackoverflow, reddit, quora, and discussion bulletin boards as such becoming your friends as well. Many experienced programmers (who are once novices) share what they know openly as they've benefited from such open sharing before. 'Openness' is an important part of the coder/developer's culture. Polly's jumping ahead a bit here by saying – suffering is inevitable, but the more you suffer now, the easier the life is tomorrow.

Journey

Throughout the assignments, you will learn to (1) program Internet applications using the TCP socket APIs, as well as (2) the web APIs. All the assignments will be developed in Go on a Unix-based OS. This simple Web server will allow downloading of html pages. To simplify the programming task and to proceed incrementally, we will lead you through the implementation in 2 stages. At the 1st stage, you will be asked to program a file upload server using TCP sockets. At the 2nd stage, we move to program a Web server using the web APIs provided conveniently in Go.

Stage 1

There will be 6 assignments at the 1st stage, i.e., PA1 to PA6. You'll learn the basic Unix commands you need to work with a Unix-based workstation (a remote server) in PA1. The remaining assignments are then about network programming in Go. Respectively, you'll learn to:

- access files on the disk in PA2,
- write a file upload client using TCP socket in PA3,
- write a file upload server using TCP socket in PA4,
- extend PA4 to allow repeated file uploads in PA5
- extend PA5 to allow concurrent file uploads in PA6

The Assignment

Let's learn to work with a Unix-based workstation where you will upload your code and perform tests. It is important that you test your code on the workstation, because it is also where Polly will test your code on. In a way, every submission will be graded on the same base, i.e., same version of compiler and same environment setting.

Now follow the instructions below and practice the basic commands to get around the Unix operating system.

1. Getting an account

You will be uploading/testing your code on a Unix-based workstation. Please feel free to code/debug locally on your own computer and upload/test later on the workstation. The workstation runs on this IP address: 140.112.42.221. Each team will be granted an account. After you've completed forming your team and reported to Polly, she will respond with your team's account username and password. Those who form the team or join the course late will have a limited amount of time to work on the 1st assignment which is due the 2nd Friday of the semester.

2. Preparation

Only Wintel users will need this before you may proceed to the following exercises. If you are a Wintel user, install an 'ssh' client first. 'ssh' stands for **Secure SHell**. It enables you to log on to a Unix-based system securely. There are a lot of ssh clients for Wintel (or even for macOS) – PuTTY, Pietty and etc. Install either one or anyone that supports ssh. For downloading and installation instructions, please refer to:

- Putty: <http://www.chiark.greenend.org.uk/~sgtatham/putty/>
- Pietty: <http://ntu.csie.org/~piaip/pietty/>

3. Getting on to a Unix-based system

With the ssh client installed, you may now log on to the Unix workstation (140.112.42.221) using your account. If the login is successful, you should see the Unix prompt:

```
$
```

Find out about the current time by:

```
$ date
```

4. Checking out about the existing files

Check out the files in the current directory by:

```
$ ls
```

Create a file that contains your login time by:

```
$ date > login-time.txt
```

Check whether there is a new file created in the directory by:

```
$ ls
```

Check the content of the file by:

```
$ cat login-time.txt
```

5. Helping yourself

You may find out how to use ls for more information about files by:

```
$ man ls
```

From the output above, try if you can find the flag to show the files in long format.

```
$ ls -l
```

6. Moving from directory to directory

Create a directory by:

```
$ mkdir PA1  
(rmdir deletes a directory)
```

Copy the login time file to the above created directory by:

```
$ cp login-time.txt PA1
```

Go to the above created directory by:

```
$ cd PA1
```

Check whether the login time file is copied to the new directory by:

```
$ ls -l > file-check.txt
```

Go back to the original directory by:

```
$ cd ..
```

7. Renaming and removing

Rename the login time file in the original directory to tmp.txt by:

```
$ mv login-time.txt tmp.txt
```

(Directories can be renamed the same way.)

Remove the tmp.txt file by:

```
$ rm tmp.txt
```

8. Downloading file

Only Microsoft/MacOS users will need an SFTP client for the purpose. If you are a Microsoft/MacOS user, install an SFTP client first. Almost all FTP client supports SFTP these days. 'SFTP' stands for **Secure FTP**. It enables you to upload/download files to/from a system supporting ssh connectivity essentially.

There are a lot of FTP clients for Microsoft Windows or MAC based system – FileZilla and etc. Install either one or anyone that supports SFTP. For downloading and installation instructions, please refer to:

- FileZilla: <https://filezilla-project.org/>
- Tutorial video: <https://www.youtube.com/watch?v=rUNQphoGVwQ&fbclid=IwAR31MjUZuWSsX2FKDgOunt68KRYgVBd6IVQII1s7AE23ZhPVE6Pd5mhLFvE>

Now use the SFTP client to download the login-time.txt file (in your PA1 directory) on the workstation to your computer.

9. Uploading file

Edit the login-time.txt file on your computer now, by adding stackoverflow's URL. Change also the file name to login-time-new.txt.

Now use the SFTP client to upload the login-time-new.txt file up to the workstation and move it into your PA1 directory.

10. Logging out

Check if you have created the PA1 directory and these three files – login-time.txt, login-time-new.txt and file-check.txt – are in the directory. These will be your proof of completing the PA1.

And finally, simply log out by:

```
$ logout
```

11. Online Unix Tutorials

If you are interested in learning more, there are plenty UNIX operating system tutorials around on the Web. Just google. You should find, for example, the UNIX Tutorial for Beginners at: <http://www.ee.surrey.ac.uk/Teaching/Unix/>
Or this one at Utah Math: <http://www.math.utah.edu/lab/unix/unix-tutorial.html>

12. Submit your PA1

There is no need to submit anything. Polly will check the team accounts on the workstation and verify whether the instructed activities are carried out by each team.