CSE 240 Homework 1

Chen, Acuña 1/12/2021

Introduction

The aim of this assignment is to make sure that you understand and are familiar with the concepts covered in the lectures, including Unix operating system, programming paradigms, the structure of programming languages, and the differences between a macro and a procedure. By the end of the assignment, you should have

- Learned a brief history of programming languages and the characteristics of the languages.
- Gotten started with Unix and GNU GCC the programming environment.

This assignment is related to the outcomes 1-2 and 1-3 listed in the syllabus:

- Students will understand the control structures of functional, logic, and imperative programming languages.
- Students will understand the execution of functional, logic, and imperative programming languages.

Optional Reading (6e): Read chapter 1, chapter 2 (sections 2.1, 2.2, and 2.3), appendix (sections B.1 and B.2), and course notes (slides).

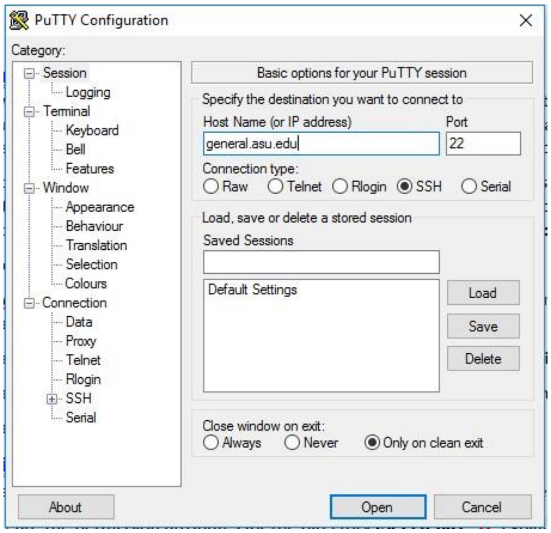
You are encouraged to ask and answer questions on the course Slack. (However, **do not share your answers or code**.)

Pre-requisite

This homework requires using the GCC compiler installed on ASU's general server. By the time you start this homework, you should have already read the page called "M1 & 2: GCC and Visual Studio" on Canvas. This page of the homework reiterates some of the ideas of accessing GCC.

To do the C assignments using GNU GCC, you need to have basic Unix knowledge. If you are not familiar with basic Unix commands, you may need to read the Unix Tutorial given on Canvas.

1) Run a program such as **PuTTY** or **SSH** on your PC or Mac. Configure the terminal program to connect to **general.asu.edu** on **port 22**. Note: if you are running on Linux or a Mac, then you can start an SSH connection to **General** by opening a terminal window and typing **ssh *yourasuruteid*@general.asu.edu** at the prompt.



Specify the destination you want to connect to

Host Name (or IP address): general.asu.edu

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

2) Log in to **General** using your ASUAD credentials.
3) Do an **ls** command to see the files in your home directory.
4) Do an **ls -a** command to see **all** of the files in your home directory, including hidden files.
5) Do an **ls -l** command to produce a **long** listing of the non-hidden files in your home directory.
6) Do an **ls -al** command to produce a **long** listing of **all** of the files in your home directory.

Make a new directory (e.g. MyDir or CSE240) by using the command: *mkdir MyDir*, use this to store all of your C and C++ programs in a specific directory. You may create subdirectories in this directory.

Enter the directory by using Change Directory command: *cd MyDir*

To write a GNU GCC program on a Unix server, you can either use a Unix editor, e.g., nano, *vim* or *pico* or upload (myFiles) a file into your Unix directory *MyDir*. The name of a C program should have an extension *.c* and the name of a C++ program should have an extension *.cpp*

Programming Exercise (50 points)

1. Review the lecture slides which discuss Very Simple Programming Languages (VSPL). Next, review VSPL as defined below and identify which sequences are valid.

```
<letter>        ::= a | b | c | d | e
<LETTER>        ::= V | W | X | Y | Z
<number>        ::= 0 | 1 | 2 | 3 | 4
<letters>       ::= <letter> | <letter> <letters>
<LETTERS>       ::= <LETTER> | <LETTER> <LETTERS>
<numbers>       ::= <number> | <number> <numbers>
<sequence>      ::= <letters> <LETTERS> <numbers> | <LETTERS> <letters> <numbers>
```

Which of the following are valid sequences? You must clearly identify for each of the following sequences, which are valid and which are invalid. Each sequence is worth 1 point. Submit your answer as hw01q1.pdf.   [10 points]

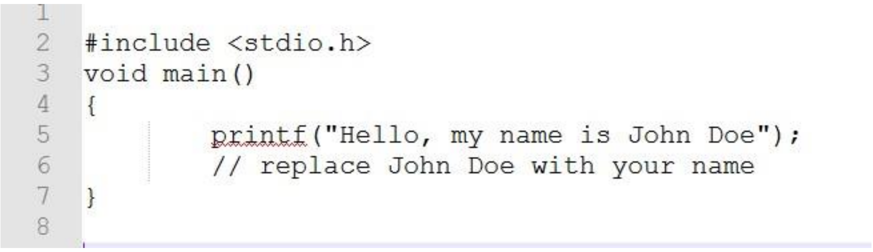
1. CSE240
2. ebcXYZ125
3. XYZcde344
4. VWXa10
5. edc135790Z
6. dYaZeWkV
7. Zbad00
8. aZ21
9. XYZabc23
10. Ey701

2. Follow the Textbook Appendix B and the Unix Tutorial PPT given in the homework folder to complete the following tasks on the ASU General server:

2.1 Create a new directory called CSE240. Use cd CSE240 to enter the new directory. Use pwd command to find the path from the Unix root directory to your current directory.

2.2 Create three new subdirectories inside CSE240. Name the directories d1, d2, and d3 respectively.

2.3 Enter the directory d1 and use nano or vim to create a new program called hello.c and enter the following code into the program. You must use your name to replace the name “John Doe”. Save the file.



```
#include <stdio.h>
void main()
{
	printf("Hello, my name is John Doe");
	// replace John Doe with your name
}
```

Use *gcc hello.c -o hello* to compile the program. Fix any compilation errors if any.

2.4 Use ls, ls -l, and ls -al respectively to list files in d1 directory. Use command ./hello to execute the compiled code. . [3 points]

Take screenshots of each command in this question. You may take one screenshot with all commands in it.

2.5 Enter d2 directory and then use one command to copy (not move) the files hello.c and hello from d1 directory into d2 directory. Use ls -al to view all the files in d2.

Take screenshots of each command in this question. You may take one screenshot with all commands in it. [3 points]

2.6 In d2 directory, use chmod 660 hello to change the permission. Use ls -l to view the files. Use command ./hello to execute the compiled code. Take a screenshot of this output. Use chmod command again to make hello an executable, but not readable and not writeable for all users. Use command ./hello to execute the compiled code. Take a screenshot of this output. [3 points]

2.7 Enter CSE240 directory. Use ls -l to view all the files. Then, use one command to delete d2 directory and all the files in d2 directory. Use ls -l to view all the files.

Take screenshots of each command in this question. You may take one screenshot with all commands in it. [3 points]

Screenshots are needed for questions 2.4 through 2.7. Put all the screenshots along with their question numbers in a word file and convert to pdf. Submit the file as hw01q2.pdf

3. In this question, you will use Unix tools to edit, debug, and execute a small C program. The purpose of the homework is to learn the Unix programming environment. Optionally read textbook Section 2.2.3, the tutorials, and the Unix tutorial in text Appendix B.2.

3.1 Use a text editor to enter the program on textbook page 47 (the code is given here for your convenience). Use GNU GCC to compile, debug (find and fix any syntax errors), and execute the program and fix any semantic errors, for example, by adding "break" statements in the required places and by fixing incorrect characters copied into the programming environment, if any, and by type-changing to print a floating point number. The code is supposed to perform one math operation in each switch-case. The ‘ch’ variable is assigned a new math operator before each switch-case. Submit the corrected program as hw01q3.c [8 points]

```
/* This C program demonstrates the switch statement without using breaks. */
#include <stdio.h>
 main() {
        char ch = '+';
        int a = 10, b = 20;
       double f;
        printf("ch = %c\n", ch);
        switch (ch) {
       case '+': f = a + b; printf("f = %d\n", f);
       case '-': f = a - b; printf("f = %d\n", f);
       case '*': f = a * b; printf("f = %d\n", f);
       case '/': f = a / b; printf("f = %d\n", f);
       default: printf("invalid operator\n");
        }
        ch = '-';
        printf("ch = %c\n", ch);
        switch (ch) {
       case '+': f = a + b; printf("f = %d\n", f);
       case '-': f = a - b; printf("f = %d\n", f);
       case '*': f = a * b; printf("f = %d\n", f);
       case '/': f = a / b; printf("f = %d\n", f);
       default: printf("invalid operator\n");
        }
        ch = '*';
        printf("ch = %c\n", ch);
        switch (ch) {
       case '+': f = a + b; printf("f = %d\n", f);
       case '-': f = a - b; printf("f = %d\n", f);
```

```
        case '*': f = a * b; printf("f = %d\n", f);
        case '/': f = a / b; printf("f = %d\n", f);
        default: printf("invalid operator\n");
        } ch =
        '/';
        printf("ch = %c\n", ch);
        switch (ch) {
        case '+': f = a + b; printf("f = %d\n", f);
        case '-': f = a - b; printf("f = %d\n", f);
        case '*': f = a * b; printf("f = %d\n", f);
        case '/': f = a / b; printf("f = %d\n", f);
        default: printf("invalid operator\n");
        } ch =
        '%';
        printf("ch = %c\n", ch);
        switch (ch) {
        case '+': f = a + b; printf("f = %d\n", f);
        case '-': f = a - b; printf("f = %d\n", f);
        case '*': f = a * b; printf("f = %d\n", f);
        case '/': f = a / b; printf("f = %d\n", f);
        default: printf("invalid operator\n"); }
}
```

4. You are given a file named “hw01q4.c”. All instructions are given in the form of comments in the file. You should correct the errors and identify which error type they are. Please read all instructions carefully, then complete and submit the updated file. [20 points]

What to Submit?

This homework assignment has multiple files. You are required to submit your answers in a compressed format (.zip). Make sure your compressed file is labeled correctly: lastname_firstname**1**.zip. (All lowercase, do not put anything else in the name like "hw1".)

The compressed (zip) file MUST contain the following:

hw01q1.pdf (BNF practice)
hw01q2.pdf (screenshots from questions 2.4 through 2.7)
hw01q3.c (program)
hw01q4.c (program)

No other files should be in the compressed folder.

If multiple submissions are made, the most recent submission will be graded.

Submission preparation notice: The assignment consists of multiple files. You must copy these files into a single folder for Canvas submission. We suggest that you double check your submission to Canvas: if you submitted an empty folder, an incomplete folder, or a wrong folder, you cannot resubmit after the submission linked is closed! We grade only what you submitted in the canvas. We cannot grade the assignment on your computer or any other storage, even if the modification date from your computer indicated that the files were created before the submission due dates. The contents of Canvas are what counts.