CS530 Assignment #1

Purpose

This assignment is designed to familiarize you with ELF files.

Requirements

This assignment consists of one major requirements.

1) Developing solutions to the problems below.
2) Your assignments will be checked, submitted and graded electronically.

Problem

1) PLEASE READ INSTRUCTIONS BEFORE MAKING ACCOUNT!
   You must signup for an account at gitlab.com:
   IMPORTANT
   You will be sharing your username with me. Please make it school appropriate.

   http://www.gitlab.com

   We are going to use gitlab for this class because it is both a standard fully functional git management webportal and has unlimited private repositories for you store your assignments. We could use the more standard github.com, however you would not be able to make private repositories (and this makes your code available for all students). All of your projects/repositories for this class should be set of private when you create them. Please note that any members you add to the repository is tracked and recorded (thus if you give another student access to a repository I will know about it and it will be recorded as cheating).

2) Setup a Debian[1] based linux system. If you do not have an additional computer or do not want to permanently install linux on one of your machines you can setup a virtual computer for your linux system to run on. I recommend VirtualBox[2]. You can find a number of tutorials online with the correct search terms. If you wish to use another linux distribution or virtualization software you may. I will be testing on a Debian distribution with default packages.

---

1 Recommended. Mint: https://www.linuxmint.com/ or Ubuntu: http://www.ubuntu.com/

2 VirtualBox: https://www.virtualbox.org/

3) Setup a project named “CS530Assignment1” under your account. Then you can add a new project.
4) Clone[3] this repository to your computer (or your virtual machine, or both). I recommend the software SourceTree for a graphical user interface on windows or osx. In your linux installation on virtualbox you will need to learn how to use the command line interface (Learning the git command line tools is essential to being developer with hireable skills in the future). After this repository is cloned to your local machine (you should probably do this in your virtualbox), create a file called README.md and put your Full Name and Email (each on their own line, as they appear in blackboard) in the file. Commit the file with an appropriate commit message (don’t forget to push the commit to the server). You will also complete the readme to contain the information required by the assignment later.

5) Create the following C programs/libraries. Your programs will follow the naming scheme specified: progX_Y.zzz Where X is the assignment number, Y is the program/library number for the assignment and zzz is the file extension for the particular language we will be writing that code in. For example, the first program will be in the file prog1_1.c (Capitalization matters!). Every driver you write for this class will start by printing a standardized header. The header will look like this:

```
Assignment #X-Y, <NAME>, <EMAIL>
```

Only programs that have entry points will print the header. In this assignment part 1 (prog1_1.c will print a header, but prog1_2.c/prog1_2.h will not, prog1_3.c will).

Your program must use the strings used in the examples provided. You do not have any creative leeway in the prompts or the responses. Most of the grading in this course is done automatically and the autograder is *extremely* unforgiving.

Please ensure that you commit and push your submission files as well.

Not all of your programs will be a driver (have an entry point). Some of the programs you write will be libraries for other programs.

---

3 How to use GIT: https://rogerdudler.github.io/git-guide/ We will be creating the repositories on the server through the web interface, so you do not need to do that step on your local computer. If you are using sourcetree I recommend: https://github.com/GSoft-SharePoint/Dynamite/wiki/Getting-started-with-SourceTree,-Git-and-git-flow

prog1_1

Create a C program that takes a single command line argument FILE. The FILE will be a target elf file that you will read as a binary file in its entirety. Your program will then print that files CRC32 checksum as an 8 digit hex number.

Example compilation:
```
gcc prog1_1.c -o prog1_1
```

Example execution:
```
./prog1_1 target_file
```

Example run (User input to STDIN highlighted with yellow):
```
Assignment #1-1, Scott Lindeneau, slindeneau@sdsu.edu
FA13E1C4
```

prog1_2

Create a C program that takes a single command line argument FILE. The FILE will be a target elf file. Your program will then print the CRC32 checksum of ONLY the program header table (not the file header or the section headers or any sections).

Example compilation:
```
gcc prog1_2.c -o prog1_2
```

Example execution:
```
./prog1_2 target_file
```

Example run (User input to STDIN highlighted with yellow):
```
Assignment #1-2, Scott Lindeneau, slindeneau@sdsu.edu
0013AE34
```

prog1_3

Create a C program thakes two command line arguments, FILE and SECTION_NAME. The FILE will be a target elf file, SECTION_NAME will be a string identifying which sections for which your program will produce CRC32 checksums. If there are multiple sections that match the provided SECTION_NAME your program will produce CRC32 checksums for each of them

Example compilation:
```
gcc prog1_3.c -o prog1_3
```

Example execution:
```
./prog1_3 target_file .rodata
```

Example run (User input to STDIN highlighted with yellow):

```
Assignment #1-3, Scott Lindeneau, slindeneau@sdsu.edu
5FCE304C
6A84EFAB
```

6) After you have committed your files you will have to grant me developer access to your repository. To do this you must open the project on gitlab.com, and open the Members page that is under the settings menu on the right hand side of the page (It looks like a gear). Add me to your project by typing in my username: slindeneau and make sure that the project access is set to developer. When I grade your project I will push a new branch called “Grade” to your repository with the autograder output and any notes associated with your grade.

   You will need to do this for <u>every</u> assignment.

7) The last step involves verifying that everything is working correctly. Please go to:
   http://cs530.lindeneau.com
   Input your first and last name (as it appears on blackboard), your gitlab username and select the assignment you would like to grade. Verify that all of the parts of your program work as expected.

<u>DO NOT ENTER ANY PASSWORDS</u>

Passwords are not required for cs530.lindeneau.com to function. If your project does not get graded, you have not given correct access to the correct developer account on gitlab.

Additional Details

- You should be able to investigate any issues you have on your own and spend the time necessary to understand what is happening overall.

Late Policy

Programs will be given 120% if turned in by the date specified, otherwise the program will be worth 100% up to one week later. After that time no late work is accepted.

Assignment #1
Due 10/10 11:59pm

Cheating Policy

There is a zero tolerance policy on cheating in this course. You are expected to complete all programming assignments on your own. Collaboration with other students in the course is not permitted. You may discuss ideas or solutions in general terms with other students, but you must not exchange code. (Remember that you can get help from me. This is not cheating, but is in fact encouraged.) I will examine your code carefully. Anyone caught cheating on a programming assignment or on an exam will result in a zero for the class or assignment.