

2021 Fall: CPSC 240

Practice Assignment 1 The Hello World Program

Preface

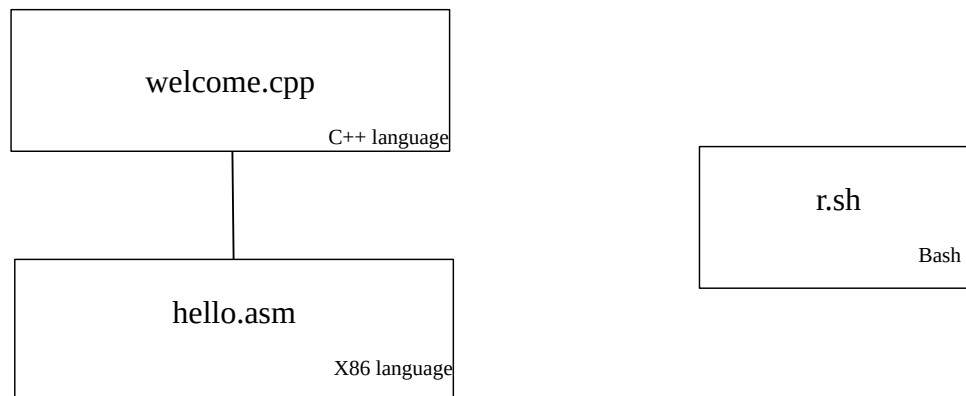
This assignment welcomes you to the world of programming called “X86 Assembly Programming”.

Basic requirements

Make a program satisfying the structure given in the diagram that follows. The application purpose of the program is that you input your name and the program says “hello” to you.

The academic purpose of this program is that you experience how to handle strings in X86. Notice that there are no numbers in this first assignment of the semester. There will be many future assignments that use both float and integer numbers.

Calling structure



The system structure has three files as shown in the diagram above. Each rectangle represents one file. Inside of each rectangle is the name of the file and the language of the rectangle. The name of the language is usually in smaller font.

The top level file begins the execution of the system. Its job is to welcome the user in a friendly manner. Typically the top level file does not know what the file below it is doing.

You may substitute C language for C++ language if you prefer. Later in the semester it will be required that C language be used. For this assignment you may choose either one.

The file named hello.asm is written in X86 assembly language, which is the language you will learn this semester.

The file r.sh is written in bash. Bash is one example of a script language. You can learn enough bash for this course by downloading and studying the example programs posted at the class website.

You should know that the class website does not require a password in order to view open source knowledge. Knowledge (facts) should be shared freely on this earth. That is why a password is not required.

Sample dialog with the program

Welcome to this friendly 'Hello' program created by Jerry Long.

Please enter your first and last names: Santa Claus

Please enter your title (Ms, Mr, Engineer, Programmer, Mathematician, Genius, etc): Programmer

Hello Programmer Santa Claus. How has your day been so far? Pretty good.

Pretty good is really nice.

This concludes the demonstration of the Hello program written in x86 assembly.

Stay away from viruses Santa Claus.

Bye. The integer zero will now be returned to the operating system.

Legend. The text in light blue is output from the welcome module (C++ or C).

The text in yellow is output from the hello module (X86).

The text in pink is input from the keyboard.

The requirements of this program

The calling structure diagram must be preserved. That means welcome must call hello.

The source code in the hello.asm file must follow an organized internal structure. Assembly language is extremely difficult to read. To alleviate this problem organized your source files in the way the sample assembly files are organized. Even when you do the maximal amount of structural organization of the source code the file may still be impossible to understand by any one except the original programmer. Apart from this fact you still have to try to organize the X86 source file.

The program must run to completion without crashing. The assembly term for a runtime crash is "seg fault".

The two source files must carry the proper correct open source license.

The two source files must carry the professional identification in areas of author, program, and module.

The one X86 file must be block structured.

Blocks must be properly identified.

There must be no evidence of any file having been copied from a previous program. If comments made by the the professor or any other third party are found in the source files then the entire program is invalidated.

It is ok to copy executable source code from other programs found on the web, but don't copy the comments found on the web. You must create your own comments. You must personalize the program to make it your own.

Suggested date of completion

September 6, 2021

If you finish this practice assignment by the above date you are making normal progress.

FAQ: Why should I do this assignment? Answer: The midterm is open notes. You may want to refer to this practice assignment in order to find an answer to a question on the test.

Where to get help

Search through the posted X86 programs to find one that shows how to accomplish what you want to do in your program.

Form a question and place it in the search box of an online search engine. It will lead you to example showing how to do what you need to do.

Go ask one of the free tutors provided by the CPSC department. Tutors will announce specific hours for tutoring each week. You need to log in by zoom at the proper time. At the time of this writing (Aug 8) the hours for tutoring have not yet been announced.

The word 'tutor' means teacher. The tutor is there to teach you something. Be a smart student who can ask the right question. Don't ask "Show me the instruction that will output the welcome message in X86." That's bad. Do ask "Teach me what I need to know about string-IO in X86."

Go to the discord server dedicated to assembly programming. Post your question for others to see. Sooner or later another student who has already encountered the same issue will post an answer to your question. Check back later to find the solution you wanted.

Go to the weekly SI sessions. Here the agenda is already set by the SI leader. The SI leader is there to teach everyone an aspect of the 240 course. However, some SI leaders are willing to accept questions from the audience after the teaching session has concluded. Go to the SI sessions and find out if there are answers for your important question.

Use Zoom and go to the professor office hour. This is a public hour. Other student will be present to listen to your question. The first thing the professor will do is ask you what answers did you receive from the 5 avenues of help listed above on this page. Gather your records that document what happened in each of the 5 resources.

- Sample programs

- Search engine

- Tutor

- Discord

- SI meeting

Show that you diligently tried everything to obtain an answer, and all those answer failed.

Addendum: Suggestion for enhancing your employability.

This page has nothing to do with assignment 1. It is related to getting a job.

All potential employers are interested in what you have done during your university career. You should prepare for that question two years in advance of searching for a job.

The question “What did you do as a CS major?” may occur during the initial online contact or it may occur later during a job interview. All job searching is done online in the modern day. Don’t be surprised if you encounter the question when you are first submitting your resume.

Prepare for the question by saving a copy of your best program or best project in every CS class you take. From every course save your best performance not matter what it is: program, term paper, project report, user guide, design charts, etc.

Initially you save all of these products in a folder in your own computer. I still have one called Portfolio although I don’t use it much any more.

Later you need to post your accomplishments online. Most professional people and most students prefer github for this purpose. But, you can use something if you prefer.

In your resume include a link to the location where your best products from each class are stored. That link is the answer to that question “What have you done during your university years?”

Good luck

Software. This document was created by “Libre Office Writer” word processor. The file is saved in an open source format known as “odt”. If you open this document using Microsoft software then in most cases any diagram like the one on page 1 will be distorted. Yes, Google docs do a fine job of replacing Microsoft Office. I like Google docs but I don’t like to depend on the cloud for everything I do. That is why you should have Libre Office, which is available as a free download at <https://www.libreoffice.org/download/download/> . There are versions available for Macs, Windows, and Linux.