

Problem Set 1

Due before midnight on Tuesday, January 22, 2008.

1 Assignment Goals

1. To familiarize yourself with the tools needed for this course: the Zoo computer facility, your Zoo course account, a good text editor, the C compiler and linker (gcc), a Linux command shell, and basic Linux commands.
 2. To learn the basic file types that comprise a C program: .c source files, .h header files, .o object files, and binary executable commands (generally with no file extension).
 3. To write your first complete C program.
 4. To learn how to manually compile and link a multi-module program.
 5. To learn how to run a binary executable command.
 6. To learn how to use the Submit script to submit your assignment.
-

2 Wisdom!

Congratulations on your purchase of Wisdom! software. Built to the highest standards of quality, your satisfaction is assured.

Some assembly is required before use. Please follow these simple instructions and your product will give you a lifetime of satisfaction.

Parts List You should find the following listed parts in the Zoo course directory:

`/c/cs223/assignments/ps1`

1. The core wisdom object file, part name “pearls.o”.
2. The core wisdom header file, part name “pearls.h”.

Do not continue if these files cannot be located or if you do not have a Zoo account. Assembly can only be performed on the Zoo. A Windows or Mac machine cannot be used for this purpose.

Tools Required You will need to use the following tools:

1. The C compiler `gcc`.
2. The linker, also called `gcc`.
3. A good text editor such as `emacs`.
4. A Linux command shell such as `sh`, `bash`, `csh`, or `tcsh` (standard on all accounts).

Assembly Overview In assembling your Wisdom! software, use `emacs` to create a command source file `wisdom.c`. Apply `gcc` in compile mode to `wisdom.c` and the furnished `pearls.h` to produce the command object file `wisdom.o`. Carefully mix together `wisdom.o` with `pearls.o` using `gcc` in link mode. Give your executable command the name `wisdom`. Type `wisdom` at a shell prompt to get Wise!

Detailed Assembly Instructions

1. Using `emacs`, create a C program file `wisdom.c` that uses `#include` to include the header file `pearls.h` and contains only a main program. The program should first print a welcome line with your name and the number of this course. It should then call the function whose prototype is contained in `pearls.h`.
2. Compile your program using `gcc` with the following command line options:

<code>-c</code>	Puts <code>gcc</code> into compile mode.
<code>-o wisdom.o</code>	Specifies the name of the object module to be created by the compiler.
<code>-std=c99</code>	Tells the compiler to accept the C99 dialect of C that we will be using in this course.
<code>-pedantic</code>	Says not to accept language extensions not provided by the C99 standard.
<code>-Wall</code>	Tells the compiler to warn of possible source code errors.
<code>-I path</code>	Tells the compiler an additional place to look for header files.
<code>-g</code>	Says to include the symbol table for later use by the debugger
<code>-O1</code>	Tells the compiler to use optimization level 1

For the *path* argument to the `-I` switch, you need to specify the directory that contains the header file `pearls.h`, which is in `/c/cs223/assignments/ps1`. It should be used from there and *not* copied to your course directory.

If you get any warnings from the compiler, go back to step 1, correct the error in the source code, and repeat step 2.

3. Link your program using `gcc` in the default linker mode. You will only need to specify one option:

<code>-o wisdom</code>	Specifies the name of the executable command file to be created by the linker.
------------------------	--

In addition to the `-o` option, you will need to give `gcc` as command line arguments the names of the two compiled object files to be linked together: `wisdom.o` and `pearls.o`. Because `pearls.o` is not in your working directory, you will need to give its full path name `/c/cs223/assignments/ps1/pearls.o`

4. Test your new program by typing its name on the command line. If this fails, try preceding the name with `./`.
5. Now that your program works, install it in the submissions directory where it can be admired and used by the course instructor and TA's. Do this using the `/c/cs223/bin/submit` script on the Zoo. (Documentation is available here or in the text file `/c/cs223/doc/submit.txt`.) In order to properly admire your code, they will want to see not only the command file `wisdom` but also the intermediate files you produced: `wisdom.c` and `wisdom.o`, as well as a file containing sample output from two or three runs of your program. You can cut and paste text from the shell window into a text editor in order to prepare the output file for submission. *Please do not forget to submit all required files.*

Satisfaction Guarantee Our operators are on duty 7 days a week to help you with any problems you may encounter. Go to the bottom of the course web page for contact information and help desk hours.