Lectures and Assignments

Compilation Pipeline

Lecture Notes

Lecture 1: Compilation Pipeline (PDF - 1.0MB)

Lab Exercises

The primary goal of this lab period is to get your C compiler up and running.

We have 2 “Hello, World!” examples for you to practice compiling to make sure that everything is working.

Lab 1 files (ZIP) (This ZIP file contains: 3 .c files and 1 .h file.)

Compile `hello1` with:

```
gcc hello.c -o hello1
```

Compile `hello2` with:

```
gcc main.c hello.c -o hello2
```

Assignment 1

Setup

Assignment 1 files (ZIP) (This ZIP file contains: 3 .c files and 2 .h files.)

The zip contains 3 C files:

1. fibeverse.c
2. fibonacci.c
3. reverse.c

And 2 header files (.h):

1. fibonacci.h
2. reverse.h

You can compile them with this command (though it won't work at first; see Problem 1):

```
gcc -Wall -std=c99 _fibeverse.c reverse.c fibonacci.c_ -o **fibeverse**
```

You can run the resulting program with two arguments: a number, then a string (in quotes):

```
./fibeverse 6 'what a trip that was!'  8  was! that trip a what
```

The first line it prints is the 6th fibonacci number. The second line is the string you provided, with the words reversed.

Problem 1

Unfortunately, the code doesn't compile as-is! Fix the compile errors and warnings. `gcc` should produce no output with the above command when you are done.

Problem 2

I can't decide whether I want a program that computes Fibonacci numbers or a program that reverses strings! Let's modify fibeverse so that it can be compiled into either.

Use the preprocessor macros we taught in class to make it so that I can choose which program it is at compile time.

When I compile it with this command, it should compute the Fibonacci number but not reverse the second argument:

```
gcc -Wall -std=c99 **-DFIBONACCI** fibeverse.c reverse.c fibonacci.c -o
**fibonacci**
```

Then I can run it like this:

```
./fibonacci 8
```

When I use this command, it should reverse the string I provide as the first argument, and not do any fibonacci calculation:

```
gcc -Wall -std=c99 **-DREVERSE** fibeverse.c reverse.c fibonacci.c -o **reverse**
```

Then I can run it like this:

```
./reverse 'a brave new world'
```

It should work as it originally did when I provide both compiler flags:

```
gcc -Wall -std=c99 **-DFIBONACCI -DREVERSE** fibeverse.c reverse.c fibonacci.c -o
**fibeverse**
```

Solutions

Assignment 1 solution (PDF)