MAIN
PREVIOUS
NEXT
HELP
Follow

Exercise 1: Dust Off That Compiler

Here is a simple first program you can make in C:

```
int main(int argc, char *argv[])
{
    puts("Hello world.");

    return 0;
}
```

You can put this into a `ex1.c` then type:

```
$ make ex1
cc     ex1.c   -o ex1
```

Your computer may use a slightly different command, but the end result should be a file named `ex1` that you can run.

What You Should See

You can now run the program and see the output.

```
$ ./ex1
Hello world.
```

If you don't then go back and fix it.

How To Break It

In this book I'm going to have a small section for each program on how to break the program. I'll have you do odd things to the programs, run them in weird ways, or change code so that you can see crashes and compiler errors.

For this program, rebuild it with all compiler warnings on:

```
$ rm ex1
$ CFLAGS="-Wall" make ex1
cc -Wall    ex1.c   -o ex1
ex1.c: In function 'main':
ex1.c:3: warning: implicit declaration of function
'puts'
$ ./ex1
Hello world.
$
```

Now you are getting a warning that says the function "puts" is implicitly declared. The C compiler is smart enough to figure out what you want, but you should be getting rid of all compiler warnings when you can. How you do this is add the following line to the top of `ex1.c` and recompile:

```
#include <stdio.h>
```

Now do the make again like you just did and you'll see the warning go away.

Extra Credit

- Open the `ex1` file in your text editor and change or delete random parts. Try running it and see what happens.
- Print out 5 more lines of text or something more complex than hello world.
- Run `man 3 puts` and read about this function and many others.