Xiao Qin

qin39@purdue.edu

CS 240

lab1

Problem 1

Where is the file gcc located? *usr/*bin/gcc

What is contained in the file gcc? Gcc contains program for pre-processing, translating, and linking.

What does gcc do when it generates a new file a.out? It pre-processing,translates and links.

Where is a.out located? /homes/qin39/Desktop/240/lab1/v1

To view the content of main.c, you can use software such as cat, more, or vim.

What happens when you run cat, more, and vim with a.out as input?

Cat :Weird symbols show up.

More: a.out: not a text file

vim: wield symbols like @,^

As a file, how is a.out different from main.c ?

a.out is composed of machine code after pre-processing, translating and indexing, and main.c is composed of c language.

(ignore the fact that main.c contains a C program)?

a.out show the output and prompt input, but main.c is the source code.

How does gcc react if you change the type of main() to void? It still yieds a.out.

How does gcc react if you omit the type declaration of main()? warning: return type defaults to ‘int’

Does gcc generate a.out? Yes

What happens if you declare the type of main() as char? It still yields a.out

Problem 2

Where is the header file stdio.h located? usr/lib/nodejs/npm/man/man7

Open stdio.h (e.g., using vim) and find printf(). Does stdio.h contain the C code of printf()? no

If not, what information about the function printf() is contained in stdio.h?
The return type and the parameters and parameters' types.

What does it mean for gcc to link the function main() with the library function printf()?

So that when main function run, it can use the printf() function from library without writing the function.

By default, why does gcc not compile printf() during the linking process?

Since we have the object file already, which is machine code, we do not need to compile printf();

What happens if you run gcc with option -c? Main.o will be created instead of a.out.

What is the name of file generated by gcc? Main.o

Try executing the output generated by gcc. What happens and why?
Permission denied.ain.o contains compiled contents of main.c, not linked into an executable yet**.**

What part of the string "result of %d * %d equals %d\n" is literal (just characters to be printed as is) and what part is reserved, i.e., part of formatting control of C (i.e., its syntax) that determines how the values of variables is output?

"result of" and "equals" are literal. "%d" is reserved.

What is \n and what impact does it have?

\n means next line.

What happens if you omit \n in the string argument of printf()?

If you omit \n, your answer will stick to the prompt text next line.

Problem 3

what is the role of & (i.e., ampersand) in scanf("%d %d",&x,&y) and why is scanf("%d %d",x,y) incorrect in C?

& means the address of. Without &, information is not stored to the right address.

Compile the code as is, and run it to verify that it works correctly.

Modify the code by removing the two ampersands and compile main.c using gcc.

What does gcc say after the modification?

It gives warning:

```
warning: format ‘%d’ expects argument of type ‘int *’, but argument 2 has type ‘int’ [-Wformat=]
   scanf("%d %d",x,y);
```

Why does gcc still generate a.out (beyond "it does what it does and I didn't code gcc") even though we can easily tell the code has a bug?

It is just a warning, the system set it by default.

What happens when you run a.out and input two integers 5 and 3? What is going on?

Segmentation fault. So the system will not save the formation and will discard the value input.

Problem 4

When passing the two arguments to mult2(), why do we not use ampersands, that is, mult2(&x,&y)? Because mul2() function's parameter are x and y, whose value has been stored using scanner.

Why does providing &z as the third argument of mult2() enable the callee to update the value of variable z which belongs to main()? Because & means "address of" so that the value of z's address is updated.

Why does providing z as the third argument not work? Because the function mul2() does not have third argument.

Make the above modifications, store the code in a different file main1.c, and compile the modified C code in main1.c. What does gcc say?

```
main.c: In function 'main':
main.c:17:7: error: too many arguments to function 'mult2'
   z = mult2(x,y,&z);
       ^~~~~
main.c:7:7: note: declared here
 float mult2(float, float);
       ^~~~~
main.c: At top level:
main.c:30:5: error: conflicting types for 'mult2'
 int mult2(float a, float b,float *c )
     ^~~~~
main.c:7:7: note: previous declaration of 'mult2' was here
 float mult2(float, float);
       ^~~~~
main.c: In function 'mult2':
main.c:32:7: error: 'c' redeclared as different kind of symbol
 float c;
       ^
main.c:30:35: note: previous definition of 'c' was here
 int mult2(float a, float b,float *c )
                                   ^
main.c:36:3: error: invalid type argument of unary '*' (have 'float')
   *c = a * b;
```

^~

What additional modifications do you need to make to have a correct code?

I added an argument for mult2 declaration *c, and remove the return type for the function.

Figure it out on your own. If you get stuck, ask during the PSOs and office hours and we will help.

Since mult2() does not return a value, what type declaration should we assign mult2() to reflect/highlight this feature? No return type but a float *c parameter.

Problem 5

1)What modification to v5 has been performed in v6?

Mult2 function has been moved to a separate file.

2)How should the code of v6 be compiled?

gcc main.c mult2.c

3)What final modification has been carried out in v7?

General header has been removed to a separate file.

4)Explain what the difference between

#include "generalheader.h"
and
#include <generalheader.h>
is.

In practice, the difference is in the location where the preprocessor searches for the included file.

For #include <filename> the preprocessor searches in an implementation dependent manner, normally in search directories pre-designated by the compiler/IDE. This method is normally used to include standard library header files.

For #include "filename" the preprocessor searches first in the same directory as the file containing the directive, and then follows the search path used for the #include <filename> form. This method is normally used to include programmer-defined header files.

5)What happens if you replace the double quotes of generalheader.h by angle brackets and compile the code in v7?

```
fatal error: generalheader.h: No such file or directory
 #include <generalheader.h>
          ^~~~~~~~~~~~~~~~~
compilation terminated.
```

6)Suppose you were a Purdue system administrator with super user (or root) privilege to the Linux PCs in our lab. What action could you undertake so that the code in v7 with

```
#include <generalheader.h>
```

compiles correctly when running gcc?

I would make generalheader.h in search directories pre-designated by the compiler/IDE.

7)Why can't you or I undertake such an action?

Because I’m not having super user privilege to the Linux PCs in lab.

Bonus problem

Modify the code in v3 discussed in Problem 3 so that it compiles and runs correctly (i.e., reads two integers from standard input, multiplies them, and prints the result to standard output) when using

```
scanf("%d %d",x,y)
```

to read the two integers to be multiplied. Explain the changes you have made and why it works without using ampersands in scanf().

I added two more int variables a,b and change x, y to be the address.

Once input are store in address x,y, we make address x,y to be address of a and b. so that a and b contain the input to be used for manipulation