Lecture 5: Makefiles, intro to lab02, and review of loops and function mechanics

Monday, April 17, 2017 12:31 PM

INTRO TO LAB02, MAKEFILES, REVIEW LOOPS/FUNCTIONS

Problem Solving with Computers-I

https://ucsb-cs16-sp17.github.io/

Clickers out – frequency AB

How do you feel about frequency of hws

A. Too much

B. Too little

C. Okay

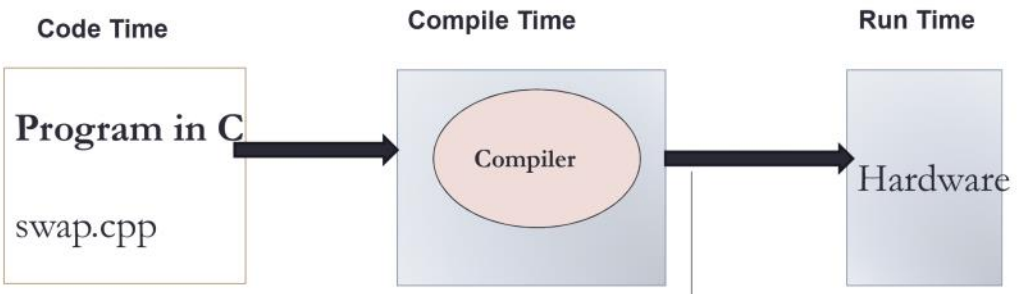
Monday, April 17, 2017 12:31 PM

Demo

- Basics of code compilation in C++ (review)
- Makefiles (used to automate compilation of medium to large projects) consisting of many files
- We will start by using a makefile to compile just a single program
- Extend to the case where your program is split between multiple files
- By the end of this you should know what each of the following are and how they are used in program compilation
  - Header file (.h)
  - Source file (.cpp)
  - Object file (.o)
  - Executable
  - Makefile
  - Compile-time errors
  - Link-time errors

Monday, April 17, 2017 12:31 PM

Steps in program translation



Program:
Text file stored on computers hard disk or some secondary storage

Executable:
Program in machine code
+Data in binary

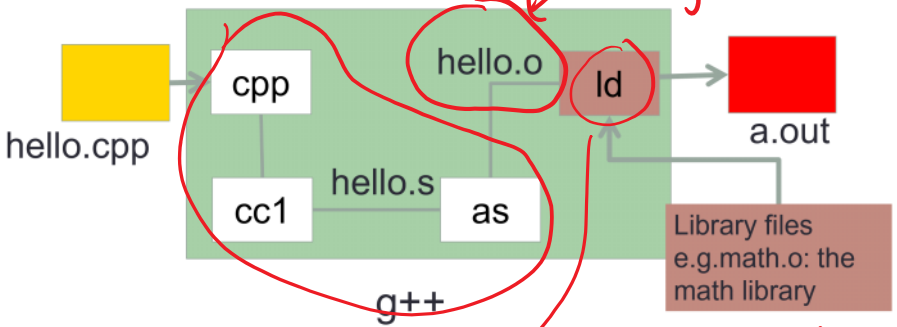
```
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
```

Monday, April 17, 2017 12:31 PM

4

g++ is composed of a number of smaller programs

- Code written by others (libraries) can be included
- ld (linkage editor) merges one or more object files with the relevant libraries to produce a single executable



obj (file)

only contains a translation of your code

linker

g++ hello.cpp -c → hello.o

functions.h

void SayHello();
void PrintData();

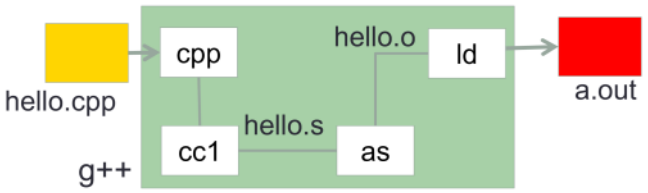
A. Declarations
B. Definitions
C. Call

SayHello →

Steps in gcc

- Ask compiler to show temporary files:

```
$ g++ –S hello.cpp
$ g++ –c hello.o
$ g++ –o hello hello.cpp
$ g++ functions.o main.o –o myhello
```



Makefiles (Example covered in class)

target dependencies

```
all : hello myhello #rule for target all
# make looks for a rule for each dependency
hello : hello.o
    g++ hello.o -o hello
hello.o : hello.cpp # if hello.cpp is modified target hello.o is regenerated by running the g++ command
    g++ hello.cpp -c
myhello : main.o functions.o
    g++ main.o functions.o -o myhello
main.o : main.cpp
    g++ main.cpp -c
functions.o : functions.cpp
    g++ functions.cpp -c
clean:
    rm *.o hello myhello
```

Monday, April 17, 2017 12:31 PM

Control Flow: while and do while loops

```
while(Boolean expression){
    //statement 1
    //statement 2
}

do{
    //statement 1
    //statement 2
  }while(Boolean expression);
```

Monday, April 17, 2017 12:31 PM

Identify the code that is not equivalent to the other two?

Assume 'n' is an integer that has already been declared (may be positive or negative)

A.

```
for( int x = 0; x < n; x++ ) {
  cout<<x <<endl;
}
```

B.

```
int x = 0;
while(x < n) {
  cout<< x << endl;
  x++;
}
```

C.

```
int x = 0;
do{
  cout<< x<< endl;
  x++;
} while(x < n);
```

D. They are ALL equivalent

n = -1