

Quick notes

- ❖ Check Piazza

- ❖ Reminders:

- Weekly quiz 3 is due today
- Assign3 is due friday

- ❖ Autograder

- If you get a high score, that'll be your grade
- If you don't, I'll look at your code + lab log

- ❖ Assignment 4 won't be as difficult

- Python and diff/patch
- Due one week after assign3
- We'll talk about this stuff today/wednesday

- ❖ Questions?

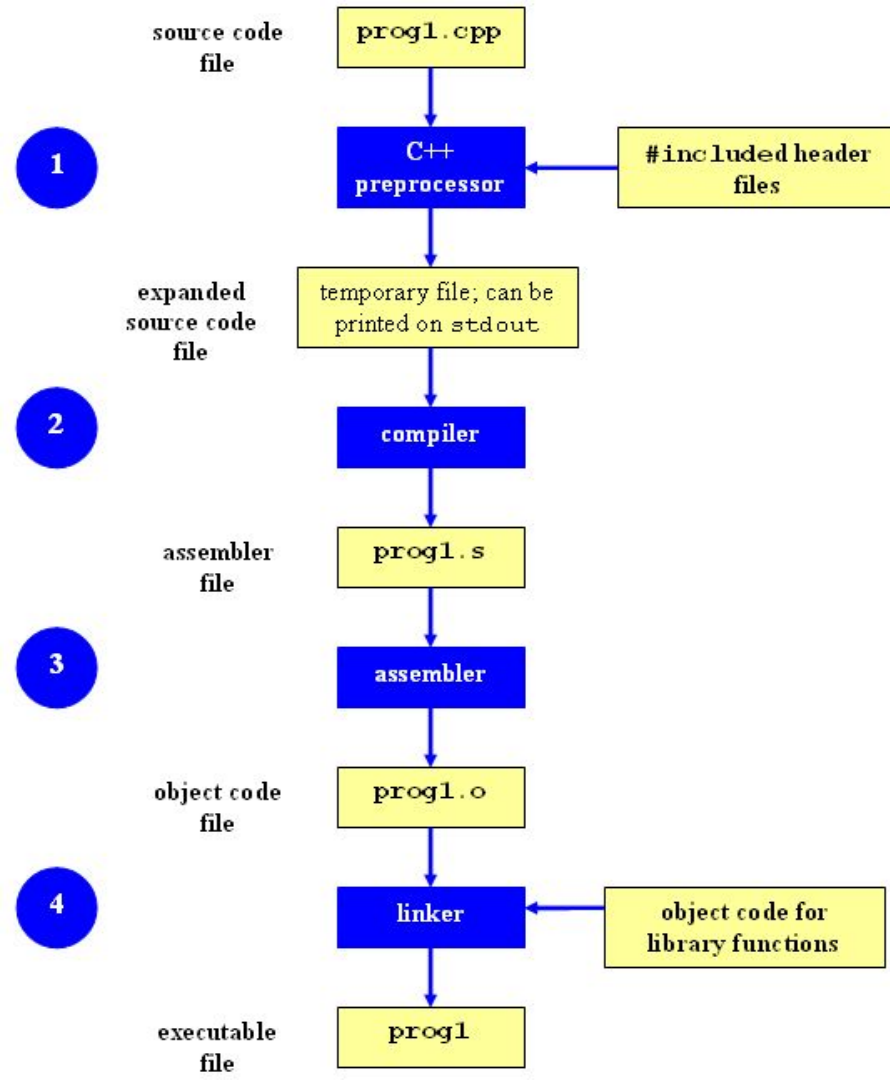
Feedback / Office Hours

❖ Tameez Latib

- tameezlatib@gmail.com, please add “CS35L” to the subject line
- Office Hours: Monday **4pm-6pm** (or by appointment)
- Feedback: <https://forms.gle/6kcJ2aJtzAzFMhHQ7> (anonymous google form)

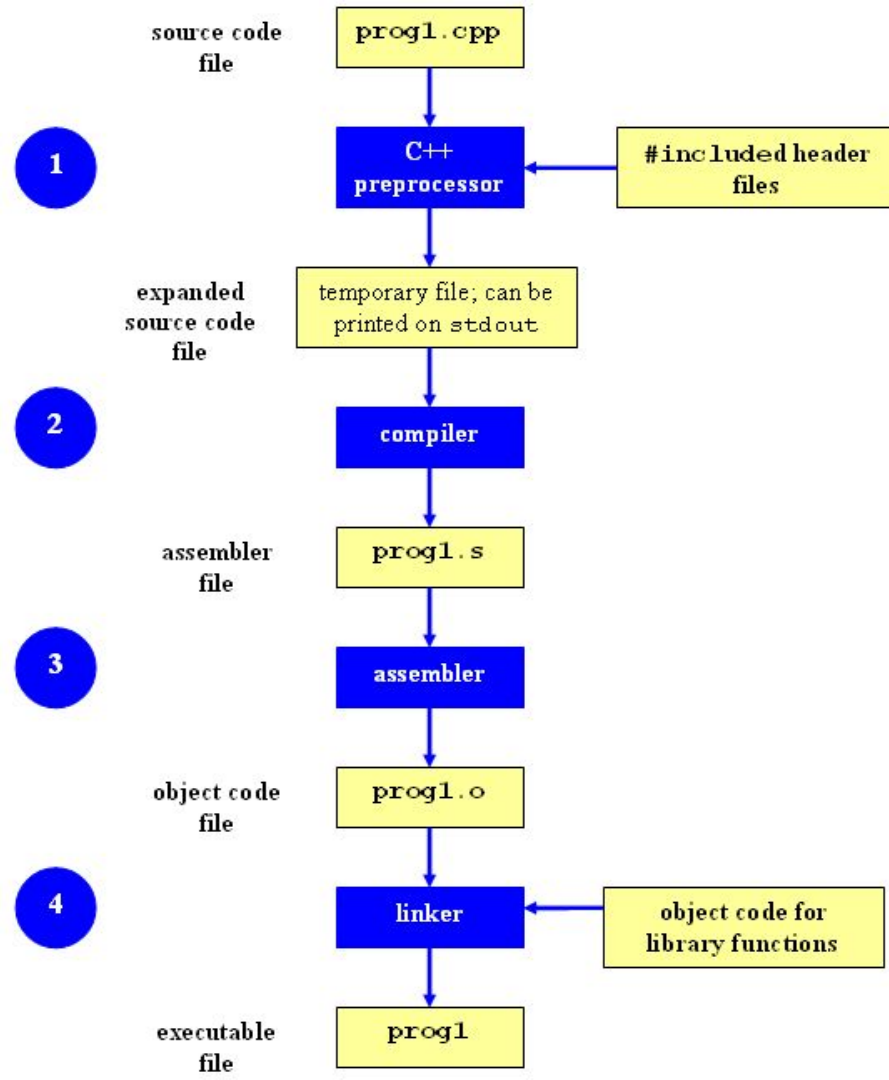
Compilation (and steps)

- ❖ Suppose you have c code, prog.c
- ❖ You want to run this code
 - ./prog.c ??
- ❖ Need to compile first
- ❖ gcc -o prog prog.c ; ./prog
 - This does all the steps for you



The steps

- ❖ Preprocessor (expand)
 - Headers, macros, #define, comments
- ❖ Compiler
 - Convert your code to assembly language
- ❖ Assembler
 - Assembly code to object file / machine code
 - Code in 0/1 only now
- ❖ Linker
 - If other libraries are used, link them
 - Produces final executable



Linking

❖ Static

- Copies code used from other libraries into your executable, at compile-time
- With static linking, only require executable file
 - No longer need .a (static library) files
- Bulkier file

❖ Dynamic

- Linked code loaded at run-time
- Require .dll (dynamic linked library) or .so (shared object) files
- Modular

The makefile

- ❖ Especially for large projects, need to link / compile lots of files
 - Sometimes need to change one small file, don't want to recompile everything
- ❖ Makefile
 - Create scripts (syntax like bash) to 'make' project
 - Uses dependencies to figure out which files need to be re-compiled
- ❖ Target: prereqs
 - Compile command
- ❖ Running make
 - Checks dependency hello.exe
 - Checks dependency hello.o
 - Runs `gcc -c hello.c`
 - Runs `gcc -o hello.exe hello.o`
- ❖ Compiling someone else's code
 - Download code + make

```
all: hello.exe

hello.exe: hello.o
        gcc -o hello.exe hello.o

hello.o: hello.c
        gcc -c hello.c

clean:
        rm hello.o hello.exe
```

The makefile

- ❖ This file MUST be called Makefile
- ❖ make all, make build, make clean, etc
- ❖ Configure
 - Configures file paths, directories, shell stuff, etc.
 - Ran before make, (./configure) so that everything is set up for make
- ❖ Some useful resources, if you want to further investigate
- ❖ <https://thoughtbot.com/blog/the-magic-behind-configure-make-make-install>
- ❖ https://www3.ntu.edu.sg/home/ehchua/programming/cpp/gcc_make.html

How to send out updates

- ❖ Large project, needs small update
 - Only download + install what you need
- ❖ How can we accomplish this:
- ❖ Diff
 - Gets the difference between files
 - Updated file and old file
- ❖ Patch
 - Takes (old) file and output of diff
 - 'Patches' in parts of diff output so that old file gets updated
- ❖ If user is on version 2, current build is version 3:
 - Send out patch(2->3) and user can use to patch their version
- ❖ Check your favorite github -> commits -> (click on any)

Diff + patch example

- ❖ Have words file, want to update from version 1 to version 2
- ❖ `diff -u dirA/words1 dirB/words2 > words_patch`
- ❖ `patch -p1 dirA/words1 words_patch`
 - `-p1` -> strip first leading directory off
 - `dirA` and `dirB` removed from `words_patch`

```
$ cat words1
WORDS1
ant
bark
dog
```

```
$ cat words2
WORDS2
dog
```

- ❖ Reading diff output:
 - `-` indicates in `words1`, not in `words2`. `+` is opposite
 - `" "` = in both files
 - `-1,4` = 1st line of `words1`, 4 lines total
 - `+1,2` = 1st line of `words2`, 2 lines total

```
tamee@LAPTOP-GUPJBPCG MINGW64 ~/Desktop/test
$ cat words_patch
--- dirA/words1 2020-10-25 19:56:00.882716300 -0700
+++ dirB/words2 2020-10-25 19:56:05.111751700 -0700
@@ -1,4 +1,2 @@
-WORDS1
-ant
-bark
+WORDS2
dog
```

Getting started, lab4

- ❖ Download Coreutils 8.29
- ❖ Extract via tar
 - Check man tar (tar -xvfJ coreutils-8.29.tar.xz)
- ❖ ./configure --prefix=your/directory
- ❖ Test this ls command with (./your/directory/ls)
- ❖ If you get an error when 'make'-ing after patch
 - Use autoreconf -fi
 - Check man autoreconf for more on this
- ❖ Gpg verify

Questions?

❖ Assign3?

❖ Assign4?

➤ Wednesday we talk about more 'fun' stuff

- Python

- HW