

Ben Langmead

ben.langmead@gmail.com

www.langmead-lab.org



Source markdown available at github.com/BenLangmead/c-cpp-notes

Compiling and linking

We've seen:

- How to compile with gcc, creating an executable a.out
- How to use libraries by adding `#include` to the source
- How to use libraries by adding `-l` to the gcc command
 - E.g. `-lm` for math library
- The difference between *declaring* and *defining* a variable

These ideas come into sharper focus when we think about all that happens when we run gcc

What gcc does can be divided into three phases:

1. *Preprocessing*: bring all the relevant code together
2. *Compiling*: turn the human-readable source code into machine-readable object files
3. *Linking*: bring all relevant object files together into a binary executable