Programming in C++

Make Update

Curtis Larsen

Utah Tech University—Computing

Spring 2025

Objectives

Objectives:

- Configure Build Options
- Correct Build Dependencies

Configure Build Options

Make Variables

Define a variable with `VAR := value`. Use a variable with `$(VAR)`.

Common C++ make variables are:

- `CXX` - defines the C++ compiler, usually `g++`
- `CXXFLAGS` - defines C++ compiler flags
  - `-std=c++20` - Use C++ 20
  - `-Wall` - Enable all normal warnings
  - `-Wpedantic` - Enable all strict C++ warnings
  - `-Werror` - Enable all C++ warnings as errors
  - `-O0` - Disable optimization
  - `-g` - Enable debug

You should use these in all of your `Makefile`s, that compile/link code.

Make Variables Example

```
# Note the define once, use many times benefit.
CXXFLAGS := -std=c++20 -g -O0 -Wall -Wpedantic -Werror
CXX := g++

all: main
main: main.o Point2D.o
        $(CXX) $(CXXFLAGS) -o $@ $^
main.o: main.cpp Point2D.h
        $(CXX) $(CXXFLAGS) -o $@ -c $<
Point2D.o: Point2D.cpp Point2D.h
        $(CXX) $(CXXFLAGS) -o $@ -c $<

clean:
        -rm -f *.o main
        -rm -f *~
```

Correct Build Dependencies

Build Dependencies

Remember the format of a `Makefile` build rule:

```
target: dependency1 dependency2 ... dependencyN
<tab-indent>command
```

Configuration issues:

- Missing dependencies
- Extra dependencies

What problems do they cause?

Compiler Dependencies

`.o` file depends on one `.cpp` and zero or more `*.h` files.

- `.cpp` file is listed first.
- Zero or more `.h` files that are `#included` by the `.cpp`.
- Zero or more `.h` files that are `#included` by other `.h` files.

Configuration issues:

- Missing dependencies
- Extra dependencies

What problems do they cause?

Library Dependencies

`.a` file depends on one or more `.o` files.

- One or more `.o` files that are expected in the library.

Configuration issues:

- Missing dependencies
- Extra dependencies

What problems do they cause?

Linker Dependencies

Executable depends on `.o` files and `lib*.a` files.

- One `.o` file with `main` in it.
- Zero or more `.o` or `lib*.a` files with functions called by `main`.
- Zero or more `.o` files with functions called by other `.o` or `lib*.a` files.
- Zero or more `lib*.a` files with functions called by other files.

Configuration issues:

- Missing dependencies
- Extra dependencies

What problems do they cause?

Make Variables Example

```
# Note the define once, use many times benefit.
CXXFLAGS := -std=c++20 -g -O0 -Wall -Wpedantic -Werror
CXX := g++

all: main
main: main.o Point2D.o
        $(CXX) $(CXXFLAGS) -o $@ $^
main.o: main.cpp Point2D.h
        $(CXX) $(CXXFLAGS) -o $@ -c $<
Point2D.o: Point2D.cpp Point2D.h
        $(CXX) $(CXXFLAGS) -o $@ -c $<

clean:
        -rm -f *.o main
        -rm -f *~
```