

Introduction

Introduction to gcc and Makefiles

Stef Nychka

Department of Computing Science
University of Alberta

CMPUT 201, Fall 2006

Overview

- 1 Overview
- 2 GCC
 - Object Files and Executables
 - Flags
- 3 Better to Use Makefiles
- 4 Makefiles
 - Makefile Rules
 - clean Rule and Comments
 - A Sample Assignment 1 Makefile
 - Other Things to Note

-c and -o

You will use gcc to compile your C code.

```
gcc -c draw1.c
```

Compiles draw1.c, creating an object file (machine code) whose default name will be draw1.o.

```
gcc -o draw1 draw1.o memwatch.o
```

- No -c means linking is now done, which results in an executable.
- draw1.o, memwatch.o and any code from library files are linked.
- -o specifies that the executable will be called draw1.

(Other) GCC Flags

```
gcc -Wall -ansi -DMEMWATCH -DMW_STDIO -c draw1.c
```

- -Wall enables all (well, most) warnings, which often catches bugs.
- -ansi ensures your code conforms to the 1990 C ANSI standard that your text book uses.
- -D defines macros, in this case the MEMWATCH and MW_STDIO (yep, for memwatch)

In 201, you must always use these 4 flags.

Better to Use Makefiles

You could type GCC commands at the command prompt, but it's better to put these commands into a Makefile.

- Makefiles automate compilation.
- This is less error prone.
- This allows others to compile your code without understanding it.
- It only compiles what needs to be compiled, which saves time for programs with many source files.

Makefile Rules

The main construct in a Makefile is a rule:

```
target : prerequisites
<TAB>command1
```

example:

```
draw1.o: draw1.c memwatch.h draw1.h
    gcc -Wall -ansi -DMEMWATCH -DMW_STUDIO -c draw1.c
```

- If prerequisites, draw1.c or memwatch.h, are newer than the target, then must create a new target, draw1.o.
- How? Run the command, gcc.
- Must have a Tab before each command

clean Rule and Comments

- You must have a clean rule to remove .o files, your executable and core files:

```
clean:
    -rm -f *.o draw1 core
```

- Because clean is not a file and there are no prerequisites, it can only be invoked by typing `make clean`. It is called a phony rule.
- The - in front of rm means to suppress errors from non-existent files.
- Comments start with a #, and continue to the end of the line:

```
# This is a comment
```

Assignment 1 Makefile

Download the Makefile from online. TA will review the previous concepts, and also will show, in general

- that each `.c` file should be a prerequisite once, with its corresponding `.o` file as the target;
- prerequisites also contain `.h` files that may be modified (like `memwatch.h`). They do not contain C library `.h` files.;
- the first rule has the executable as the target, and all the `.o` files as prerequisites;
- how prerequisites that are also targets are handled.

Other Things to Note

- Invoke your Makefile by typing `make`. It invokes the first rule, which should create the executable.
- Can invoke an individual target by typing `make <target>`.
- Remember to have a rule that creates `memwatch.o`, and remember to link `memwatch.o` into your executable.
- More advanced Makefile concepts online at the tutorial.