

---

# GNU Make

---

# What Is GNU Make?

---

Make is a tool which controls:

- the generation of executables
- and other non-source files

of a program

- from the program's source files.

# Capabilities of Make

---

- Enables the end user to build and install your package
  - without knowing the details of how that is done
- Automatically build and intelligently rebuild
  - Figures out which files it needs to update, based on which source files have changed and determines the proper order for updating files, in case one non-source file depends on another non-source file.
  - if you change a few source files it does not recompile all of your program.
- Make is not limited to any particular language.
  - the makefile calls the shell commands you need (e.g. runs a compiler to produce an object file, the linker etc...)
- Make is not limited to building a package.
  - You can also use Make to control installing or deinstalling a package, build the docs etc..

# Writing a Makefile

---

- A rule in the makefile tells Make how to execute a series of commands

*target ... : dependencies ...*

*command*

*...*

*...*

- A target is usually the name of a file that is generated by a program (can also be the name of an action to carry out, such as `clean').
- A dependency is a file that is used as input to create the target. A target often depends on several files.
- A command is an action that make carries out.
  - put a tab character at the beginning of every command

# Building a cake...

---

cake: cake\_mix eggs water icing

mix cake\_mix eggs water

bake cake

cool cake

apply\_icing cake icing

cake\_mix: money

buy cake\_mix

money: job

go to work

job: ...

# A simple Makefile

*edit : main.o kbd.o command.o display.o \*

*insert.o : insert.c defs.h buffer.h*

*insert.o :*

*cc -o e*

*d*

*u*

*main.o : ma*

*cc -c m*

*kbd.o : kbd.*

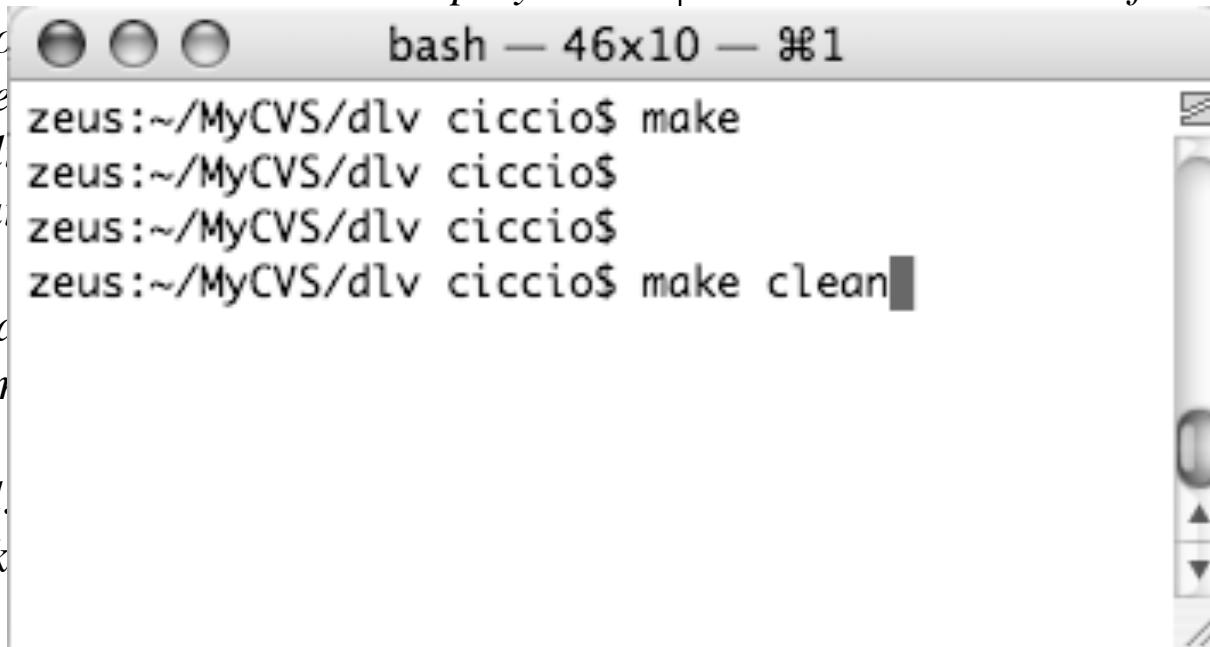
*cc -c k*

*command.o : command.c defs.h command.h*

*cc -c command.c*

*display.o : display.c defs.h buffer.h*

*cc -c display.c*

A terminal window titled "bash — 46x10 — 1" showing the execution of a Makefile. The user enters 'make' and the terminal shows the compilation of 'main.o', 'kbd.o', 'command.o', 'display.o', 'insert.o', and 'search.o'. The user then enters 'make clean' and the terminal shows the removal of the object files and executables.

```
zeus:~/MyCVS/dlv ciccio$ make
zeus:~/MyCVS/dlv ciccio$
zeus:~/MyCVS/dlv ciccio$
zeus:~/MyCVS/dlv ciccio$ make clean
```

*clean :*

*rm edit main.o kbd.o command.o \*  
*display.o insert.o search.o \*  
*files.o utils.o*

# Variables

---

A variable is defined with the syntax

*var\_name = definition*

and is expanded with *\$(var\_name)*

*objects = main.o kbd.o command.o display.o \  
insert.o search.o files.o utils.o*

*...*

*edit : \$(objects)*

*cc -o edit \$(objects)*

# Comments and Pattern Rules

---

- “#” identifies comments
  - Everithing follows “#” is considered to be a comment

- Pattern rules

*# Assemble the program.*

*t1.%.o : t1.%.dlx dlxasm*

*commands*

- \$\* The stem with which an implicit rule matches.
- \$@ The file name of the target of the rule.
- \$< The name of the first dependency.
- \$? The names of all the dependencies that are newer than the target, with spaces between them.
- \$^ The names of all the dependencies, with spaces between them.



# Special target names

---

*.PHONY : clean*

*clean :*

*-rm edit \$(objects)*

*.SILENT*

make will not the print commands to remake those particular files before executing them.

...and more (see the manual)

# What Makefiles Contain

---

- Explicit rules says when and how to remake one or more files, called the rule's targets.
- Implicit rule says when and how to remake a class of files based on their names.
- A variable definition is a line that specifies a text string value for a variable
- A directive is a command for make to do something special while reading the makefile.
- ``#'` in a line of a makefile starts a comment.

# For further info:

---

- On Unix/Linux write “man make”
- On the Internet:
  - <http://www.gnu.org/software/make/>
  - <http://www.gnu.org/software/make/manual/>
  - <http://www.student.cs.uwaterloo.ca/~isg/res/unix/make/tutorial/index.html>

---

# A working example

....the DLV Makefile