

# Make and Makefiles

# Make

---

- ▶ Make is a build automation tool
  - ▶ automatically builds programs and/or libraries from source
  - ▶ reads a Makefile that specifies how the program/library should be built
- ▶ the Wikipedia page has numerous interesting facts and quotes from the creator
  - ▶ [https://en.wikipedia.org/wiki/Make\\_\(software\)](https://en.wikipedia.org/wiki/Make_(software))
- ▶ documentation
  - ▶ <https://www.gnu.org/software/make/manual/make.html>

# Makefile

---

- ▶ a Makefile tells Make what to do
- ▶ in a simple Makefile, the contents of the Makefile consist of one or more rules
- ▶ a rule looks like:

```
target : prerequisites ...  
    recipe-step1  
    ...
```

- ▶ every recipe step must begin with a TAB character
  - ▶ not spaces!

# Makefile

---

- ▶ in a Makefile for a simple C program build, the first rule can define how the executable program should be built
  - ▶ Make calls the first target the *default goal*
  - ▶ the prerequisites are the files that are used to create the target
  - ▶ the recipe consists of the commands needed to build the target given the prerequisites

## The complete Makefile for **hello1.c**:

```
# Makefile for Hello, world! program
hello1 : hello1.c
    gcc hello1.c -o hello1
```

**hello1** depends on the file **hello1.c**; if **hello1.c** changes then Make will re-build the target **hello1**

## The complete Makefile for **hello1.c**:

```
# Makefile for Hello, world! program
hello1 : hello1.c
    gcc hello1.c -o hello1
```



The recipe to build the target  
**hello1**

The start of a Makefile for the **list-demo** program:

```
# Makefile for test_list program
list_demo : list.o list_demo.o
    gcc -o list_demo list_demo.o list.o
```

The start of a Makefile for the **list-demo** program:

```
# Makefile for test_list program
list_demo : list.o list_demo.o
    gcc -o list_demo list_demo.o list.o
```

**list-demo** depends on the files **list.o** and **list\_demo.o**;  
if one of these files needs to be re-built, then Make will also re-build **list-demo**



## The start of a Makefile for the **list-demo** program:

```
# Makefile for test_list program
list_demo : list.o list_demo.o
    gcc -o list_demo list_demo.o list.o
```

Recipe for building **list-demo** given the prerequisite files.

This recipe tells gcc to link the two object files to create the executable file **list\_demo**.

# Building the prerequisites

---

- ▶ in most C programs, the prerequisites for the overall target also need to be built
  - ▶ because the prerequisites are usually object files
- ▶ you can define a rule to build each prerequisite if needed

## Rules to build the object files:

```
# Makefile for test_list program
list_demo : list.o list_demo.o
    gcc -o list_demo list_demo.o list.o

list.o : list.c list.h
    gcc -c list.c

list_demo.o : list_demo.c list.h
    gcc -c list_demo.c
```

# Building the default goal

---

- ▶ to build the default goal simply run the command

**make**

- ▶ Make will read the Makefile in the current directory and begin processing the first rule
- ▶ before running the recipe in the first rule, Make will process the rules for the prerequisites of the first rule
- ▶ rules for targets that the goal does not depend on (directly or indirectly) are not processed unless you tell make to do so explicitly

# Building the default goal

---

- ▶ if a target already exists, then Make will rebuild the target only if the prerequisites for the target are newer than the target

# Rules to build the object files:

```
# Makefile for test_list program
```

```
list_demo : list.o list_demo.o
```

... which will cause **list\_demo** to be re-built.

```
gcc -o list_demo list_demo.o list.o
```

```
list.o : list.c list.h
```

If we edit **list.h** then **list.o** will be rebuilt...

```
gcc -c list.c
```

```
list_demo.o : list_demo.c list.h
```

```
gcc -c list_demo.c
```

...and so will **list\_demo.o**...

# Phony targets

---

- ▶ a phony target is one that is not the name of a file; rather it is just a name for a recipe to be executed when you make an explicit request
- ▶ to build a phony target, you tell Make to specifically build that target:

**make** *target*

where *target* is the name of the phony target

## Example of a phony target:

```
# Makefile for test_list program
list_demo : list.o list_demo.o
    gcc -o list_demo list_demo.o list.o

list.o : list.c list.h
    gcc -c list.c

list_demo.o : list_demo.c list.h
    gcc -c list_demo.c

clean :
    rm list_demo *.o
```