MAKEFILES

BUILD SYSTEMS

- Used for projects with multiple source files
- You don’t have to remember the full command-line with all flags
- Rebuilds only files that have been changed

MAKE

- One of many *build systems*
  - Comparatively simple
- Can be used for various different tasks / languages
- Suitable for small projects consisting of a few source files
- Checks timestamps for outdated targets
  - `-B` to force a rebuild

```
<target>: <dependencies>
»   <command>
»   <command>
    …
```

```
example: example.c other.c
    gcc -Wall -Wextra -o example example.c other.c
```

- Pitfall: commands must be indented with tabs

```
Makefile:2: *** missing separator.  Stop.
```

File should be named `Makefile` and next to your sources

```
$ make example
gcc -Wall -Wextra -o example example.c other.c
```

First rule is the default

```
$ make
gcc -Wall -Wextra -o example example.c other.c
```

Using *special variables* $@ and $^ to prevent duplication.

```
example: example.c other.c
    gcc -Wall -Wextra -o $@ $^
```

- See Manual: Automatic Variables

- Use conventional variables (CC, CFLAGS, ...)
- These variables are often initialized by Make

```
CFLAGS = -Wall -Wextra

example: example.c other.c
	$(CC) $(CFLAGS) -o $@ $^
```

- See Manual: Implicit Variables

Variables can be set from the command-line

```
$ make CFLAGS=-O2
cc -O2 -o example example.c other.c
```

- Use object files
  - Intermediate files, speeds up rebuilding

```
CFLAGS = -Wall -Wextra

example: example.o other.o
    $(CC) $(LDFLAGS) $^ $(LDLIBS) -o $@

example.o: example.c
    $(CC) $(CFLAGS) -c -o $@ $^

other.o: other.c
    $(CC) $(CFLAGS) -c -o $@ $^
```

```
$ make
cc -Wall -Wextra -c -o example.o example.c
cc -Wall -Wextra -c -o other.o other.c
cc  example.o other.o -o example
```

- Utilize Make’s implicit rules
  - Pattern matching on target and dependencies

```
CFLAGS = -Wall -Wextra

example: example.c other.c
```

```
$ make
cc -Wall -Wextra    example.c other.c   -o example
```

- See Manual: Implicit Rules

```
CFLAGS = -Wall -Wextra

example: example.o other.o
```

```
$ make
cc -Wall -Wextra   -c -o example.o example.c    # compilation of example
cc -Wall -Wextra   -c -o other.o other.c        # compilation of other
cc   example.o other.o   -o example             # linking
```

```
example ← example.o ← example.c
        ↖ other.o   ← other.c
```

- First (default) target should be `all`
- `clean` target should be present

```
CFLAGS = -Wall -Wextra

all: example

clean:
    $(RM) example example.o other.o

example: example.o other.o
```

```
$ make
cc -Wall -Wextra   -c -o example.o example.c
cc -Wall -Wextra   -c -o other.o other.c
cc   example.o other.o   -o example

$ make clean
rm -f example example.o other.o
```

- `LDFLAGS` and `LDLIBS` are used for linking
- Dependencies between files (including headers) must be stated manually
  - Common source of error

Read The Friendly Manual