UTSA
The University of Texas at San Antonio™

EE-3233, Lab Lecture 9

Python3, Assembly Language and C with Makefile

`Makefile`

- **`Makefile`**s are a special format of script files that facilitate the creation and compilation of big projects.
- **`Makefile`**s include *rules*, which indicate what commands or actions should be run by **`make`**.
- The *target* of a rule may be the name of a file, or simply the name of a list of commands.
- The list of commands included in a rule is called a *recipe*.
- Each rule may have *prerequisite* files.

Why Makefile?

- Compiling the source code files for a big project can be chaotic
  - Lots of files to compile
  - File dependency
  - Avoid recompiling every file after a small change
  - Manually rebuilding only what is needed is error prone
  - Linking
- Indispensable for big and complex projects

Targets

- Targets are normally the name of a file (such as an executable) that should be created by the commands part of the recipe. For example, to compile an executable called app, you can write:

```
app: source.c
	gcc $^ -o $@
```

This reads as: Run **gcc** to compile **source.c** (**$^**) and set the output to be an executable target file called **app** (**$@**).

Example

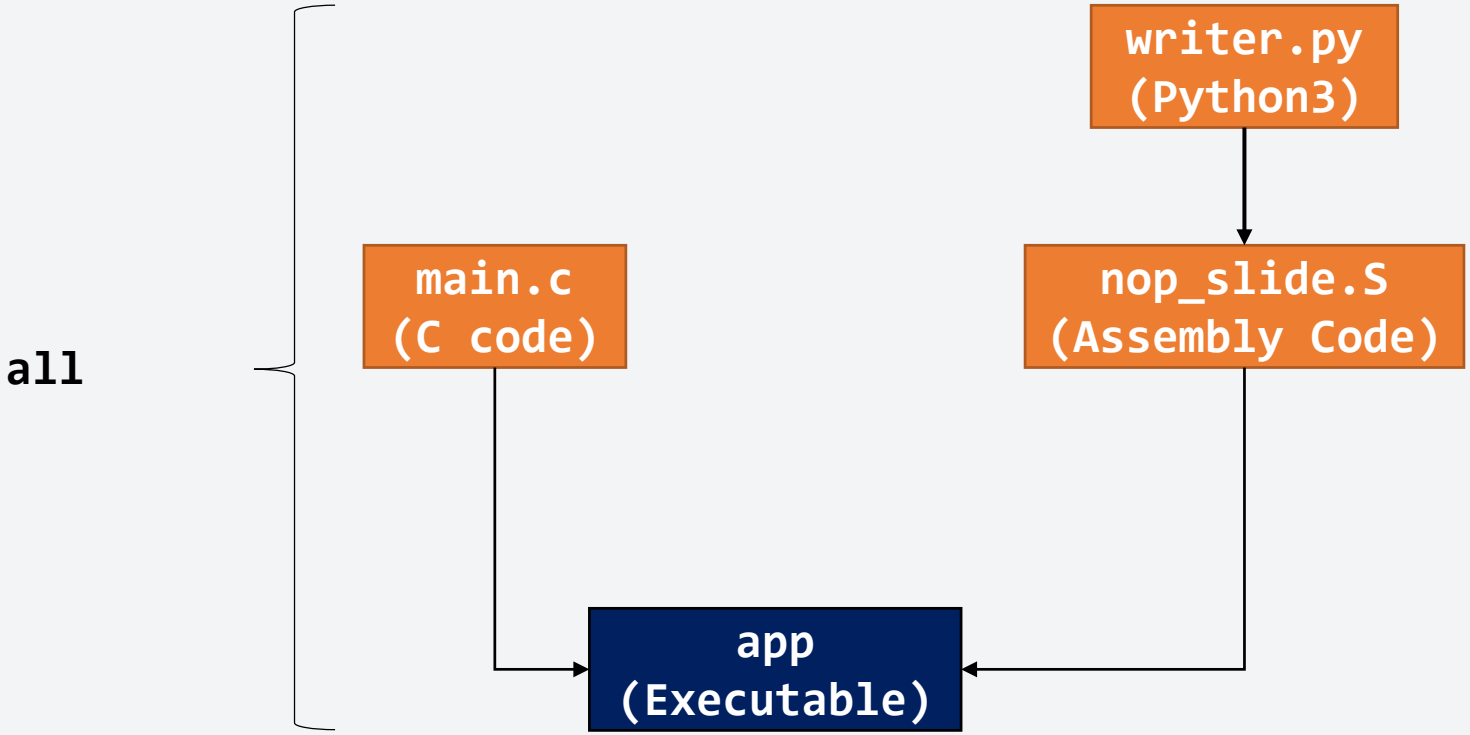
You can use variables in a Makefile (e.g., CC, PY).

.PHONY indicates that **`all`** and **`clean`** are only target names, not target files.

Reads as: **`all`** depends on **`main.c`** and **`nop_slide.S`**. If any of these files doesn't exist, automatically check if there exists an additional rule to create it.

```
1  CC = gcc
2  PY = python3
3
4  .PHONY: all clean
5
6  all: main.c nop_slide.S
7    $(CC) $^ -o app
8
9  nop_slide.S: writer.py
10   $(PY) $^ 20
11
12 clean:
13   rm -f app
14   rm -f nop_slide.S
15
```

Example


all
writer.py
(Python3)
main.c
(C code)
nop_slide.S
(Assembly Code)
app
(Executable)

utsa.edu