<https://github.com/vampy/Makefile>

Day 5

### Example Makefile

Here’s a simple example to demonstrate the basic structure of a Makefile. Assume you have a C program with the following files: main.c, helper.c, and helper.h.

makefile

Copy code

# Define variables

CC = gcc

CFLAGS = -Wall -g

# Define the target executable

TARGET = myprogram

# List of object files

OBJS = main.o helper.o

# Default targetall: $(TARGET)

# Rule to build the target$(TARGET): $(OBJS)

$(CC) $(CFLAGS) -o $(TARGET) $(OBJS)

# Rule to build main.omain.o: main.c helper.h

$(CC) $(CFLAGS) -c main.c

# Rule to build helper.ohelper.o: helper.c helper.h

$(CC) $(CFLAGS) -c helper.c

# Clean up build filesclean:

rm -f $(TARGET) $(OBJS)

### Detailed Explanation

**Variables**:

* 1. CC: The compiler to use (gcc).
  2. CFLAGS: Compiler flags (-Wall for all warnings and -g for debugging information).
  3. TARGET: The name of the executable.
  4. OBJS: List of object files.

**Default Target (**all**)**:

* 1. all: The default target, which depends on the $(TARGET). When you run make, it will build this target by default.

**Building the Target**:

* 1. $(TARGET): The target depends on $(OBJS). The command to build the target is $(CC) $(CFLAGS) -o $(TARGET) $(OBJS).

**Building Object Files**:

* 1. main.o: Depends on main.c and helper.h. The command compiles main.c to produce main.o.
  2. helper.o: Depends on helper.c and helper.h. The command compiles helper.c to produce helper.o.

**Clean Up**:

* 1. clean: A target that removes the built executable and object files. Run this with make clean.

### Makefile for Shell Scripts

You can also use Makefiles to manage tasks related to shell scripts. Here’s an example Makefile for a project with shell scripts.

makefile

Copy code

# Variables

SHELL\_SCRIPTS = script1.sh script2.sh

# Default targetall: check\_permissions run\_scripts

# Check if scripts are executablecheck\_permissions:

@for script in $(SHELL\_SCRIPTS); do \

if [ ! -x $$script ]; then \

echo "Making $$script executable"; \

chmod +x $$script; \

fi \

done

# Run the shell scriptsrun\_scripts: check\_permissions

@for script in $(SHELL\_SCRIPTS); do \

./$$script; \

done

# Clean up generated filesclean:

@echo "No clean-up required for shell scripts"

.PHONY: all check\_permissions run\_scripts clean

### Explanation

**Variables**:

* 1. SHELL\_SCRIPTS: List of shell scripts in the project.

**Default Target (**all**)**:

* 1. Runs check\_permissions and run\_scripts.

**Check Permissions**:

* 1. Ensures all scripts in $(SHELL\_SCRIPTS) are executable. If not, it makes them executable.

**Run Scripts**:

* 1. Executes each script in $(SHELL\_SCRIPTS).

**Clean Up**:

* 1. A placeholder for clean-up tasks. In this example, no specific clean-up is required.

**Phony Targets**:

* 1. .PHONY: Declares targets that do not represent files, ensuring make always executes them.

### Using the Makefile

To use the Makefile, place it in your project directory and run the following commands:

* **Build the project**: make
* **Check permissions**: make check\_permissions
* **Run scripts**: make run\_scripts
* **Clean up**: make clean