中国科学技术大学

University of Science and Technology of China

Make & CMake

中国科学技术大学

王宇航

2025年3月13日

汇报提纲

为什么要用构建工具?

- 在写代码时，特别是 C/C++ 这类编译型语言，我们需要把源代码转换成可执行文件。如果只有一个源文件，直接运行：

```
gcc main.c -o myprogram
```

- 就行了。但是当项目变大（几十上百个文件），手动敲命令就变得非常麻烦：
  - 每次修改一个文件，都要重新编译所有文件（很慢）
  - 需要记住复杂的编译参数
  - 跨平台时命令可能不同

Make 和 Makefile

❑是什么?

- ❖Make 是一个经典的构建工具，诞生于 1977 年，几乎在所有 Unix/Linux 系统上都预装了。
- ❖它通过读取一个名为 Makefile 的文本文件，来决定怎么构建你的程序。

❑怎么用?

- ❑假设你有两个 C 文件：main.c 和 utils.c，以及一个头文件 utils.h。你想编译成 myapp。
- ❑手动编译需要：

```
gcc -c main.c          # 生成 main.o
gcc -c utils.c         # 生成 utils.o
gcc main.o utils.o -o myapp
```

Make 和 Makefile

❑现在，创建一个名为 Makefile 的文件，内容如下：

makefile

```
myapp: main.o utils.o
	gcc main.o utils.o -o myapp

main.o: main.c utils.h
	gcc -c main.c

utils.o: utils.c utils.h
	gcc -c utils.c

clean:
	rm -f *.o myapp
```

```
make          # 构建 myapp
make clean    # 清理
```

解释：

- 目标: 依赖。例如 myapp 依赖于 main.o 和 utils.o，如果这两个文件比 myapp 新（或不存在），就执行下面的命令（注意：命令前必须有一个 Tab 缩进）。
- clean 是一个伪目标，用来清理中间文件。

CMake 和 CMakeLists.txt

❑是什么?

- ❖CMake 是一个更现代化的“元构建”工具，它不直接构建程序，而是生成适合你当前平台的构建文件（比如 Makefile、Visual Studio 解决方案、Xcode 项目等）。
- ❖你只需要写一份 CMakeLists.txt 配置文件，CMake 会根据你的系统环境自动生成对应的构建脚本。

❑怎么用?

- ❖还是上面的例子，创建一个 CMakeLists.txt 文件：

```
cmake_minimum_required(VERSION 3.10)
project(MyApp)

set(SOURCES main.c utils.c)

add_executable(myapp ${SOURCES})
```

CMake 和 CMakeLists.txt

- 然后，在项目目录下新建一个 build 文件夹（推荐外部构建，避免污染源代码目录）：

```
mkdir build
cd build
cmake ..          # 根据上级目录的 CMakeLists.txt 生成 Makefile
make              # 使用生成的 Makefile 进行编译
```

- 运行后，在 build 目录下就会得到 myapp 可执行文件。
- 你也可以用 cmake --build . 代替 make，这样 CMake 会自动调用合适的构建工具。

Make 与 CMake 的关系

- 简单来说：
  - CMake 负责生成 Makefile（或其他构建文件）
  - Make 负责执行构建
  - CMake 让 Make 的使用变得跨平台且更易管理。

```
源代码 + CMakeLists.txt
        |
        v
    [ CMake ]  (根据当前系统生成)
        |
        v
   Makefile  (或其他，如 .sln)
        |
        v
    [ make ]  (调用真正的编译器)
        |
        v
    可执行程序
```

总结与建议

- Make：适合小型项目、快速原型，或者你希望精确控制编译过程。

- CMake：适合中大型项目、跨平台开发，也是现在许多开源项目（如 LLVM、OpenCV）的首选。