

作业二：创建一个应用，使用命令行 `qmake`、`nmake` 命令完整创建一个 Qt 工程，编译应用等。

姓名：徐力行 学号：2022213386

步骤一：确保 Qt 和命令行工具安装好了

1. 安装 Qt 和 qmake

推荐方法是通过 Homebrew 安装 Qt：

```
brew install qt
```

安装完成后，Qt 一般会被安装在 `/opt/homebrew/opt/qt` 目录下。

2. 添加 Qt 到环境变量

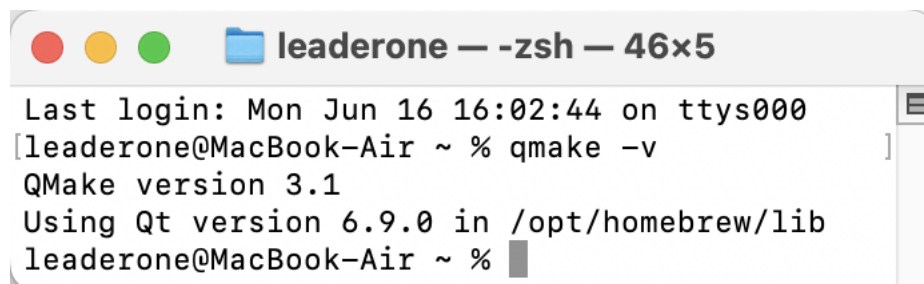
```
export PATH="/opt/homebrew/opt/qt/bin:$PATH"
```

我将这行加入 `~/.zshrc` 或 `~/.bash_profile` 中让它永久生效。

测试是否成功：

```
qmake -v
```

输出中包含 Qt 版本就表示成功。

A screenshot of a macOS terminal window titled "leaderone — -zsh — 46x5". The terminal shows the output of the command `qmake -v`. The output text is: "Last login: Mon Jun 16 16:02:44 on ttys000", "[leaderone@MacBook-Air ~ % qmake -v", "QMake version 3.1", "Using Qt version 6.9.0 in /opt/homebrew/lib", and "leaderone@MacBook-Air ~ %". The cursor is at the end of the last line.

```
Last login: Mon Jun 16 16:02:44 on ttys000
[leaderone@MacBook-Air ~ % qmake -v
QMake version 3.1
Using Qt version 6.9.0 in /opt/homebrew/lib
leaderone@MacBook-Air ~ %
```

步骤二：创建一个简单的 Qt 项目

我以一个最简单的 Qt 窗口应用为例：

1. 创建目录和文件

```
mkdir -p "/Users/leaderone/Desktop/study/企业实训/徐力行/HelloQt"
```

```
cd "/Users/leaderone/Desktop/study/企业实训/徐力行/HelloQt"
```

```
touch main.cpp
touch HelloQt.pro
```

2. 写入源代码

main.cpp

```
#include <QApplication>

#include <QLabel>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    QLabel label("Hello, Qt on macOS!");

    label.resize(200, 100);

    label.show();

    return app.exec();
}
```

3. 编写 Qt 项目文件

HelloQt.pro

```
QT += core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

TARGET = HelloQt

TEMPLATE = app

SOURCES += main.cpp
```

步骤三：使用命令行构建项目

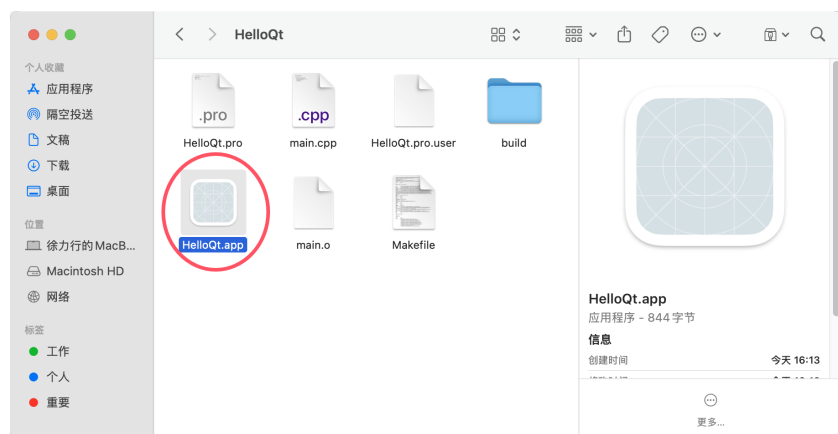
1. 运行 qmake 生成 Makefile。

2. 使用 make 编译

编译完成后，生成了一个可执行文件 HelloQt。

```

HelloQt --zsh -- 80x31
Last login: Mon Jun 16 16:09:40 on ttys000
leaderone@MacBook-Air HelloQt % qmake
Info: creating stash file /Users/leaderone/Desktop/study/企业实训/徐力行/HelloQt/.qmake.stash
leaderone@MacBook-Air HelloQt % make
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/clang++ -c -pipe -stdlib=libc++ -O2 -std=gnu++1z -arch arm64 -isysroot /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX15.5.sdk -mmacosx-version-min=14.0 -Wall -Wextra -DQT_NO_DEBUG -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_CORE_LIB -I. -I/opt/homebrew/lib/QtWidgets.framework/Headers -I/opt/homebrew/lib/QtGui.framework/Headers -I/opt/homebrew/lib/QtCore.framework/Headers -I. -I/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX15.5.sdk/System/Library/Frameworks/OpenGL.framework/Headers -I/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX15.5.sdk/System/Library/Frameworks/AGL.framework/Headers -I/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX15.5.sdk/System/Library/Frameworks/OpenGL.framework/Headers -I/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX15.5.sdk/System/Library/Frameworks/AGL.framework/Headers -I/opt/homebrew/share/qt/mkspecs/macx-clang -F/opt/homebrew/lib -o main.o main.cpp
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/clang++ -stdlib=libc++ -headerpad_max_install_names -arch arm64 -isysroot /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX15.5.sdk -mmacosx-version-min=14.0 -Wl,-rpath,@executable_path/../Frameworks -Wl,-rpath,/opt/homebrew/lib -o HelloQt.app/Contents/MacOS/HelloQt main.o -F/opt/homebrew/lib -framework QtWidgets -framework QtGui -framework AppKit -framework ImageIO -framework Metal -framework QtCore -framework IOKit -framework DiskArbitration -framework UniformTypeIdentifiers -framework AGL -framework OpenGL
leaderone@MacBook-Air HelloQt %
```



步骤四：运行应用

./HelloQt

看到一个包含 “Hello, Qt on macOS!” 的窗口。

