

0315实验报告

实验题目：构建主机端开发环境（上）

小组名称：异度侵入组

小组成员：赵宇聪 汪倩 刘迪

一、实验目的

- 建立主机开发环境
- 使用本地gcc编译应用程序
- 使用Makefile管理应用程序
- 通过autotools生成Makefile，学会常用的make操作
- 通过git/github管理团队软件和工作文件

二、实验内容

- 安装ubuntu主机开发环境
- 编写c应用程序，通过本地gcc编译
- 编写Makefile，通过autotools生成makefile，完成常用的make操作
- 创建git仓库，存储小组工作文件

三、实验过程与结果

1. 环境配置

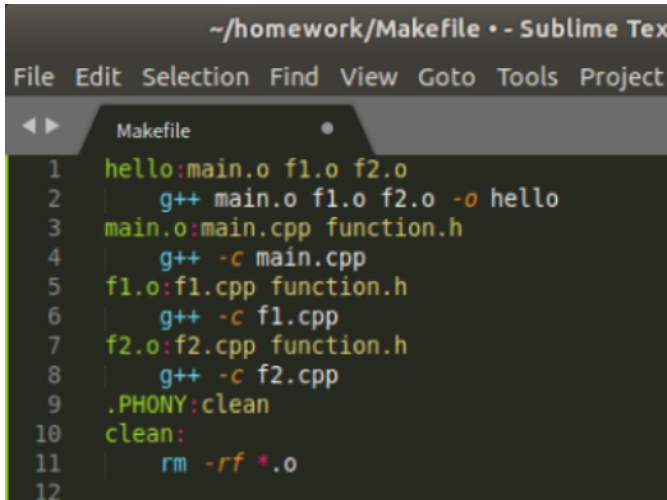
在Ubuntu中安装c语言编译所依赖的环境：

- 在终端输入命令： `sudo apt-get install build-essential`，即安装了gcc，g++和make编译器
- 输入命令 `sudo apt-get install sublime-text`，安装代码文本编辑器sublime-text

2. 编写Makefile

1) 编写了3个c++文件（若是c文件，改用gcc编译即可，其他操作一致），主函数为main.cpp，调用了f1.cpp和f2.cpp两个文件中的函数


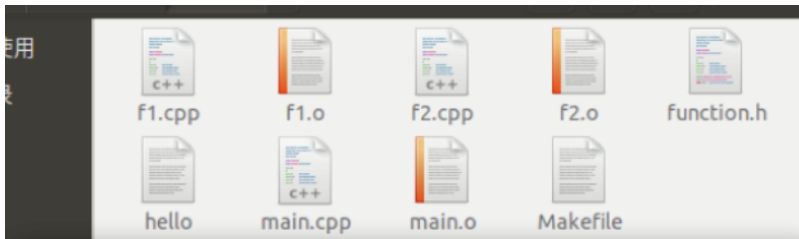
function1和function2。根据此文件依赖规则编写makefile文件如下：



```
~/homework/Makefile • - Sublime Text
File Edit Selection Find View Goto Tools Project

Makefile
1 hello:main.o f1.o f2.o
2   g++ main.o f1.o f2.o -o hello
3 main.o:main.cpp function.h
4   g++ -c main.cpp
5 f1.o:f1.cpp function.h
6   g++ -c f1.cpp
7 f2.o:f2.cpp function.h
8   g++ -c f2.cpp
9 .PHONY:clean
10 clean:
11   rm -rf *.o
12
```

在终端运行make，结果如下：

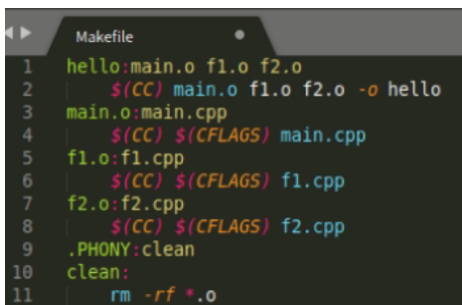


```
pursuit@pursuit-virtual-machine: ~/homework/work0
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

pursuit@pursuit-virtual-machine:~/homework/work0$ subl Makefile
pursuit@pursuit-virtual-machine:~/homework/work0$ Makefile
Makefile: 未找到命令
pursuit@pursuit-virtual-machine:~/homework/work0$ make
g++ -c main.cpp
g++ -c f1.cpp
g++ -c f2.cpp
g++ main.o f1.o f2.o -o hello
pursuit@pursuit-virtual-machine:~/homework/work0$ ./hello
hello,pursuit!
The factorial of 5 is120
```

2) 根据隐式规则，将makefile改写为：

version1：



```
Makefile
1 hello:main.o f1.o f2.o
2   $(CC) main.o f1.o f2.o -o hello
3 main.o:main.cpp
4   $(CC) $(CFLAGS) main.cpp
5 f1.o:f1.cpp
6   $(CC) $(CFLAGS) f1.cpp
7 f2.o:f2.cpp
8   $(CC) $(CFLAGS) f2.cpp
9 .PHONY:clean
10 clean:
11   rm -rf *.o
```

进一步改进，得到version2：

```

Makefile
1  CC=g++
2  CFLAGS=-c
3  OBJS=main.o f1.o f2.o
4
5  hello:$(OBJS)
6      $(CC) $^ -o $@
7  .PHONY:clean
8  clean:
9      rm -rf *.o

```

在终端运行make:

```

pursuit@pursuit-virtual-machine:~/homework/work0$ make clean
rm -rf *.o
pursuit@pursuit-virtual-machine:~/homework/work0$ make
g++ -c -o main.o main.cpp
g++ -c -o f1.o f1.cpp
g++ -c -o f2.o f2.cpp
g++ main.o f1.o f2.o -o hello
pursuit@pursuit-virtual-machine:~/homework/work0$ ./hello
hello,pursuit!
The factorial of 5 is120

```

3. 通过autotools生成Makefile

1) 安装autoconf安装包

```

pursuit@pursuit-virtual-machine:~$ sudo apt-get install autoconf
[sudo] pursuit 的密码:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会同时安装下列软件:
  automake autotools-dev libsigsegv2 m4
建议安装:
  autoconf-archive gnu-standards autoconf-doc libtool m4-doc
下列【新】软件包将被安装:
  autoconf automake autotools-dev libsigsegv2 m4
升级了 0 个软件包，新安装了 5 个软件包，要卸载 0 个软件包，有 374 个软件包未被升级。
需要下载 1,082 kB 的归档。
解压缩后会消耗 3,994 kB 的额外空间。
您希望继续执行吗？ [Y/n] y

```

2) 运行autoscan命令，生成 configure.scan 文件，复制该文件并重命名为 configure.ac文件，对其进行如图修改，目标生成程序为hello，源码文件为 main.cpp

```

~/homework/work01/configure.ac - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Makefile x configure.ac x
1  #                                     -*- Autoco
2  # Process this file with autoconf to produce a configure
   script.
3
4  AC_PREREQ([2.69])
5  AC_INIT(hello,1.0)
6  AM_INIT_AUTOMAKE(hello,1.0)
7  AC_CONFIG_SRCDIR([main.cpp])
8  AC_CONFIG_HEADERS([config.h])
9
10 # Checks for programs.
11 AC_PROG_CXX
12 #AC_PROG_CC
13
14 # Checks for libraries.
15
16 # Checks for header files.
17 AC_CHECK_HEADERS([string.h])
18 # Checks for typedefs, structures, and compiler
   characteristics.
19
20 # Checks for library functions.
21 AC_CONFIG_FILES([Makefile])

```

3) 使用aclocal命令生成 aclocal.m4文件；使用autoconf 命令生成configure脚本；使用autoheader命令生成config.h.in文件

4) 创建Makefile.am文件

```
pursuit@pursuit-virtual-machine:~/homework/work01$ more Makefile.am
AUTOMAKE_OPTIONS=foreign
bin_PROGRAMS=hello
hello_SOURCES=main.cpp f1.cpp f2.cpp function.h
```

5) 使用automake --add-missing命令生成Makefile.in文件

```
pursuit@pursuit-virtual-machine:~/homework/work01$ automake --add-missing
configure.ac:6: warning: AM_INIT_AUTOMAKE: two- and three-arguments forms are deprecated. For more info, see:
configure.ac:6: http://www.gnu.org/software/automake/manual/automake.html#Modernize-AM_005fINIT_005fAUTOMAKE-invocation
pursuit@pursuit-virtual-machine:~/homework/work01$
```

6) 在终端输入./configure，把 Makefile.in变成最终的Makefile文件

7) 运行makefile，此时该文件夹下生成hello程序

```
pursuit@pursuit-virtual-machine:~/homework/work01$ make
make all-am
make[1]: 进入目录"/home/pursuit/homework/work01"
make[1]: 离开目录"/home/pursuit/homework/work01"
```

4. 常用的make操作

1) 进入管理员模式：

```
pursuit@pursuit-virtual-machine:~/homework/work01$ sudo su
[sudo] pursuit 的密码:
root@pursuit-virtual-machine:/home/pursuit/homework/work01# make install
make[1]: 进入目录"/home/pursuit/homework/work01"
/bin/mkdir -p '/usr/local/bin'
/usr/bin/install -c hello '/usr/local/bin'
make[1]: 对"install-data-am"无需做任何事。
make[1]: 离开目录"/home/pursuit/homework/work01"
root@pursuit-virtual-machine:/home/pursuit/homework/work01# ^C
root@pursuit-virtual-machine:/home/pursuit/homework/work01# make uninstall
( cd '/usr/local/bin' && rm -f hello )
root@pursuit-virtual-machine:/home/pursuit/homework/work01#
```

2) 运行make dist、make install、make uninstall

o make dist：生成hello-1.0.tar.gz

```
pursuit@pursuit-virtual-machine:~/homework/work01$ ls
aclocal.m4      config.status  f2.cpp          main.o
autom4te.cache  configure      f2.o            Makefile
autoscan.log    configure.ac   function.h       Makefile.am
compile         configure.scan hello            Makefile.in
config.h        depcomp       hello-1.0.tar.gz missing
config.h.in     f1.cpp        install-sh      stamp-h1
config.log      f1.o          main.cpp
```

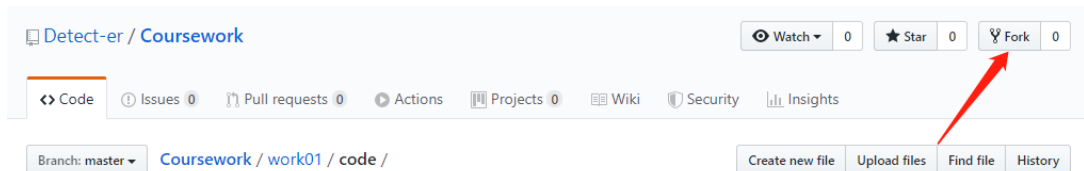
o make install/uninstall:

```
pursuit@pursuit-virtual-machine:~/homework/work01$ sudo su
[sudo] pursuit 的密码:
root@pursuit-virtual-machine:/home/pursuit/homework/work01#
make install
make[1]: 进入目录"/home/pursuit/homework/work01"
/bin/mkdir -p '/usr/local/bin'
/usr/bin/install -c hello '/usr/local/bin'
make[1]: 对"install-data-am"无需做任何事。
make[1]: 离开目录"/home/pursuit/homework/work01"
root@pursuit-virtual-machine:/home/pursuit/homework/work01#
make uninstall
( cd '/usr/local/bin' && rm -f hello )
root@pursuit-virtual-machine:/home/pursuit/homework/work01#
exit
exit
pursuit@pursuit-virtual-machine:~/homework/work01$
```

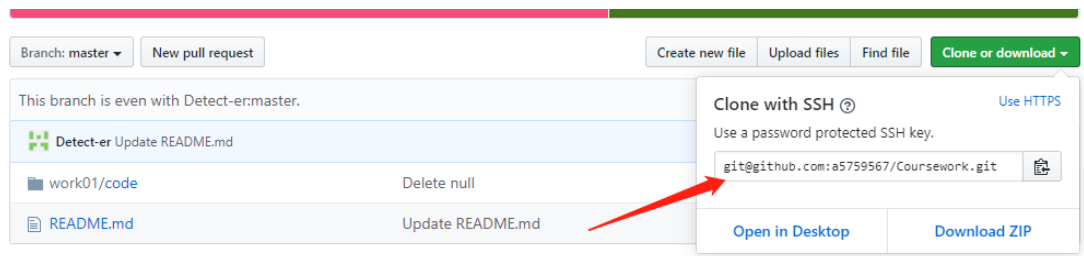
在/usr/bin/install目录下可看到安装/卸载了目标程序hello

5. git

1.从源仓库Detect-er/Coursework中把代码Fork到自己的Git hub账号里。



2.在自己Fork的仓库中获取该仓库的Clone地址。



3.使用git命令将仓库 Clone到本地。

命令：git clone <第二步中复制的仓库地址>

```
45778@LAPTOP-MF9R4DHJ MINGW64 ~ (master)
$ git clone git@github.com:a5759567/Coursework.git
Cloning into 'Coursework'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 24 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (24/24), 4.14 KiB | 707.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.
```

4.进入到Clone下来的仓库目录中。

```
45778@LAPTOP-MF9R4DHJ MINGW64 ~ (master)
$ cd Coursework/
```

5.创建新分支develop

```
45778@LAPTOP-MF9R4DHJ MINGW64 ~/Coursework (master)
$ git checkout -b develop
Switched to a new branch 'develop'
```

6.在本地修改相应内容，如添加实验报告文件



7.查看修改的文件

```
45778@LAPTOP-MF9R4DHJ MINGW64 ~/Coursework (develop)
$ git status
On branch develop
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    work01/doc/

nothing added to commit but untracked files present (use "git add" to track)
```

8.将状态改变的代码提交至缓存

```
45778@LAPTOP-MF9R4DHJ MINGW64 ~/Coursework (develop)
$ git add .
```

9.将代码提交到本地仓库中

10.将代码提交到本地仓库中

四、实验总结

coursework1中，我们学习配置了linux主机开发环境Ubuntu，使用本地g++编译了应用程序hello，并编写了原始版本的Makefile管理应用程序，然后通过隐式规则使得文件更加简洁，得到了version2。然后通过autotools生成Makefile，学会了常用的make dist、install等操作，最后通过git工具建立了小组的作业仓库，来管理团队代码和文件。

这些学习过程让我们对linux的开发环境有了入门的了解，掌握了基本的实验知识，为后续的work打下了基础。

五、实验源码

见<https://github.com/Detect-er/Coursework/tree/master/work01/code>