嵌入式软件开发技术与工具实验报告

刘一丁 黎炜桁 冉然

一、实验目的

- 在建立交叉编译开发平台之前，首先需要建立主机(虚拟机/PC，或双系统）开发环境；
- 学会使用本地gcc编译应用程序；
- 学会使用Makefile管理应用程序；
- 学会通过autotools生成Makefile，学会常用的make操作；
- 学会通过git/github管理团队软件和工作文件。

二、实验内容

- 安装主机(虚拟机/PC)Linux开发环境，Fedora，Ubuntu，Debian均可；
- 编写c应用程序，通过本地gcc编译应用程序，如果是基于x86的主机，gcc输出的执行文件运行的是x86指令集；
- 编写Makefile管理应用程序，为更好体现Makefile的作用，需编写多个c程序，给出所创建的Makefile的内容；
- 通过autotools生成Makefile，完成常用的make操作(make,make install, make uninstall, make dist)；
- 创建小组git仓库，github账号，用来存储小组工作文件以及小组报告；学习如何构建github文件，如何上传和下载github文件等。

三、实验过程与结果

1. 安装Linux开发环境

采用双系统的方式安装Ubuntu-16.04 LTS系统。

2. GCC编译应用程序

创建一个C语言程序，使用GCC进行编译并运行。C源程序代码如下：

```c
#include <stdio.h>

int main(){
    printf("hello world!");
    return 0;
}
```

使用GCC编译程序`gcc hello.c -o hello`，并运行程序，结果如下：

```
ubuntu@VM-26-77-ubuntu:~/embeded$ gcc hello.c -o hello
ubuntu@VM-26-77-ubuntu:~/embeded$ ./hello
hello world!ubuntu@VM-26-77-ubuntu:~/embeded$
```

3. Makefile管理应用程序

本实验采用已有的XML并行解析程序，共建立四个C源程序文件main.c,step1.c,step2.c,step3.c,一个头文件myXml.h，使用Makefile进行管理和编译。Makefile代码如下：

```
objects = main.o step1.o step2.o step3.o
cc = gcc
hdrs = myXml.h

xml : main.o step1.o step2.o step3.o
    gcc main.o step1.o step2.o step3.o -o xml -lpthread
main.o  : main.c $(hdrs)
    $(cc) -c main.c
step1.o : step1.c $(hdrs)
    $(cc) -c step1.c
step2.o : step2.c $(hdrs)
    $(cc) -c step2.c
step3.o : step3.c $(hdrs)
    $(cc) -c step3.c

.PHONY : clean
clean :
    -rm -f $(objects)
```

执行make命令，编译整个程序

```
-rw-rw-r-- 1 ubuntu ubuntu    18227 Mar 14 17:43 xml
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ make
gcc -c main.c
gcc -c step1.c
gcc -c step2.c
gcc -c step3.c
gcc main.o step1.o step2.o step3.o -o xml -lpthread
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$
```

执行程序，程序成功运行，编译成功

```
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ ./xml

        第一阶段预处理结束！划分文件块3276个！

        第二阶段多线程处理完毕！  并行线程数为10个！
```

4. 通过autotools生成Makefile

(1) 运行autoscan，生成configure.scan文件

```
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ autoscan
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ ls
autoscan.log  configure.scan  main.c  myXml.h  step1.c  step2.c  step3.c  test.xml  xml
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$
```

(2)修改configure.scan，设置程序名称、版本等信息，关联头文件和相关库，并修改为configure.ac

```
#                                               -*- Autoconf -*-
# Process this file with autoconf to produce a configure script.

AC_PREREQ([2.69])
AC_INIT(xml, 1.0)
AM_INIT_AUTOMAKE(program, 1.0)
AC_CONFIG_SRCDIR([main.c])
AC_CONFIG_HEADERS([config.h])

# Checks for programs.
AC_PROG_CC

# Checks for libraries.

# Checks for header files.
AC_CHECK_HEADERS([stdlib.h string.h])

# Checks for typedefs, structures, and compiler characteristics.

# Checks for library functions.
AC_FUNC_MALLOC
AC_CONFIG_FILES([Makefile])
AC_OUTPUT

```

(3)使用aclocal命令扫描configure.ac文件生成 aclocal.m4文件, 获取autoconf创建configure所需的宏定义

```
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ aclocal
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ ls
aclocal.m4  autom4te.cache  autoscan.log  configure.ac  main.c  myXml.h  step1.c  step2.c  step3.c  test.xml  xml
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$
```

(4)使用autoconf命令读取configure.ac文件中的宏，生成configure脚本

```
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ autoconf
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ ls
aclocal.m4  autom4te.cache  autoscan.log  configure  configure.ac  main.c  myXml.h  step1.c  step2.c  step3.c  test.xml  xml
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$
```

(5)使用autoheader命令生成config.h.in文件

```
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ autoheader
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ ls
aclocal.m4  autom4te.cache  autoscan.log  config.h.in  configure  configure.ac  main.c  myXml.h  step1.c  step2.c  step3.c  test.xml  xml
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$
```

(6)创建Makefile.am文件，由于该程序使用了多线程技术，因此需要在模板的基础上加入 `xml_LDADD=-lpthread`引用多线程库，文件内容如下

hello.c Makefile.am

embeded > XMLParser > XMLParallelParsing > Makefile.am

```
1  AUTOMAKE_OPTIONS = foreign
2
3  bin_PROGRAMS = xml
4
5  xml_SOURCES = main.c step1.c step2.c step3.c myXml.h
6
7  xml_LDADD=-lpthread
```

(7)使用automake命令生成Makefile.in文件。--add-missing 选项可以让 automake自动添加必需的模板文件

```
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ 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
configure.ac:11: installing './compile'
configure.ac:6: installing './install-sh'
configure.ac:6: installing './missing'
Makefile.am: installing './depcomp'
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ ls
Makefile.am  aclocal.m4      autoscan.log  config.h.in  configure.ac  install-sh  missing  step1.c  step3.c   xml
Makefile.in  autom4te.cache  compile       configure    depcomp       main.c      myXml.h  step2.c  test.xml
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$
```

(8)生成成功后，运行./configure，生成Makefile

```
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking for stdlib.h... (cached) yes
checking for string.h... (cached) yes
checking for stdlib.h... (cached) yes
checking for GNU libc compatible malloc... yes
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
config.status: executing depfiles commands
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$
```

(9)运行make，并运行生成的程序，观察编译结果

```
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ make
make  all-am
make[1]: Entering directory '/home/ubuntu/embeded/XMLParser/XMLParallelParsing'
gcc -DHAVE_CONFIG_H -I.     -g -O2 -MT main.o -MD -MP -MF .deps/main.Tpo -c -o main.o main.c
mv -f .deps/main.Tpo .deps/main.Po
gcc -DHAVE_CONFIG_H -I.     -g -O2 -MT step1.o -MD -MP -MF .deps/step1.Tpo -c -o step1.o step1.c
mv -f .deps/step1.Tpo .deps/step1.Po
gcc -DHAVE_CONFIG_H -I.     -g -O2 -MT step2.o -MD -MP -MF .deps/step2.Tpo -c -o step2.o step2.c
mv -f .deps/step2.Tpo .deps/step2.Po
gcc -DHAVE_CONFIG_H -I.     -g -O2 -MT step3.o -MD -MP -MF .deps/step3.Tpo -c -o step3.o step3.c
mv -f .deps/step3.Tpo .deps/step3.Po
gcc  -g -O2   -o xml main.o step1.o step2.o step3.o -lpthread
make[1]: Leaving directory '/home/ubuntu/embeded/XMLParser/XMLParallelParsing'
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$
```

程序运行成功

```
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ ./xml

        第一阶段预处理结束！划分文件块3276个！

        第二阶段多线程处理完毕！  并行线程数为10个！
```

(10)运行make install指令，安装程序，注意需要sudo，否则权限不够无法拷贝文件到系统目录

```
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ make install
make[1]: Entering directory '/home/ubuntu/embeded/XMLParser/XMLParallelParsing'
 /bin/mkdir -p '/usr/local/bin'
  /usr/bin/install -c xml '/usr/local/bin'
/usr/bin/install: cannot create regular file '/usr/local/bin/xml': Permission denied
Makefile:330: recipe for target 'install-binPROGRAMS' failed
make[1]: *** [install-binPROGRAMS] Error 1
make[1]: Leaving directory '/home/ubuntu/embeded/XMLParser/XMLParallelParsing'
Makefile:636: recipe for target 'install-am' failed
make: *** [install-am] Error 2
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ sudo make install
make[1]: Entering directory '/home/ubuntu/embeded/XMLParser/XMLParallelParsing'
 /bin/mkdir -p '/usr/local/bin'
  /usr/bin/install -c xml '/usr/local/bin'
make[1]: Nothing to be done for 'install-data-am'.
make[1]: Leaving directory '/home/ubuntu/embeded/XMLParser/XMLParallelParsing'
```

后退到embeded目录，运行程序，程序运行成功

```
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ cp test.xml /home/ubuntu/embeded/
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ cd ..
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser$ cd ..
ubuntu@VM-26-77-ubuntu:~/embeded$ xml

        第一阶段预处理结束！划分文件块3276个！

        第二阶段多线程处理完毕！  并行线程数为10个！

Segmentation fault (core dumped)
ubuntu@VM-26-77-ubuntu:~/embeded$
```

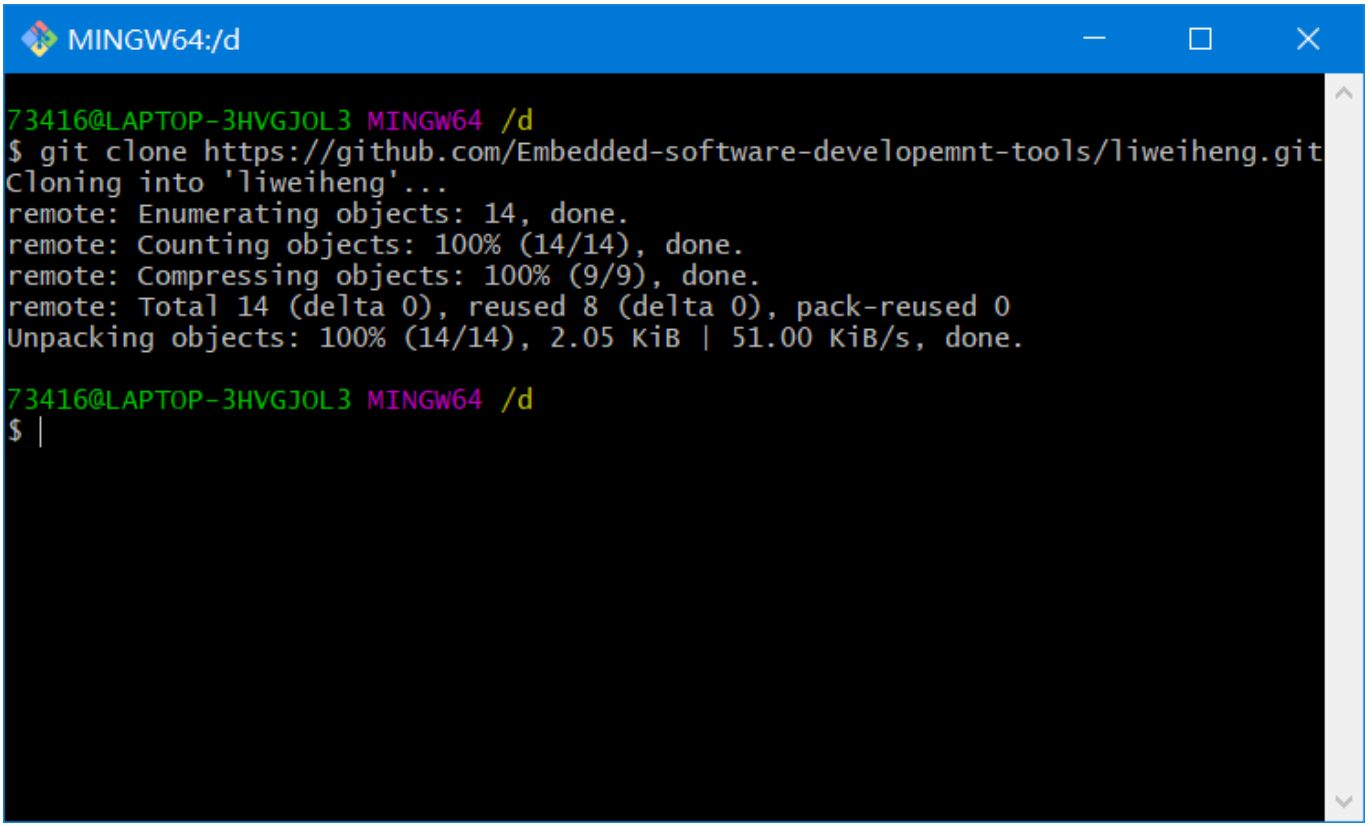
(11)运行make dist指令，打包程序，在目录下形成`xml-1.0.tar.gz`文件

```
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ make dist
make  dist-gzip am__post_remove_distdir='@:'
make[1]: Entering directory '/home/ubuntu/embeded/XMLParser/XMLParallelParsing'
if test -d "xml-1.0"; then find "xml-1.0" -type d ! -perm -200 -exec chmod u+w {} ';' && rm -rf "xml-1.0" || { sleep 5 && rm -rf "xml-1.0"; }; else :; fi
test -d "xml-1.0" || mkdir "xml-1.0"
test -n "" \
|| find "xml-1.0" -type d ! -perm -755 \
        -exec chmod u+rwx,go+rx {} \; -o \
  ! -type d ! -perm -444 -links 1 -exec chmod a+r {} \; -o \
  ! -type d ! -perm -400 -exec chmod a+r {} \; -o \
  ! -type d ! -perm -444 -exec /bin/bash /home/ubuntu/embeded/XMLParser/XMLParallelParsing/install-sh -c -m a+r {} {} \; \
|| chmod -R a+r "xml-1.0"
tardir=xml-1.0 && ${TAR-tar} chof - "$tardir" | GZIP=--best gzip -c >xml-1.0.tar.gz
make[1]: Leaving directory '/home/ubuntu/embeded/XMLParser/XMLParallelParsing'
if test -d "xml-1.0"; then find "xml-1.0" -type d ! -perm -200 -exec chmod u+w {} ';' && rm -rf "xml-1.0" || { sleep 5 && rm -rf "xml-1.0"; }; else :; fi
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$ ls
Makefile     aclocal.m4      compile      config.log     configure.ac  main.c   myXml.h   step1.o  step3.c   xml
Makefile.am  autom4te.cache  config.h     config.status  depcomp       main.o   stamp-h1  step2.c  step3.o   xml-1.0.tar.gz
Makefile.in  autoscan.log    config.h.in  configure      install-sh    missing  step1.c   step2.o  test.xml
ubuntu@VM-26-77-ubuntu:~/embeded/XMLParser/XMLParallelParsing$
```

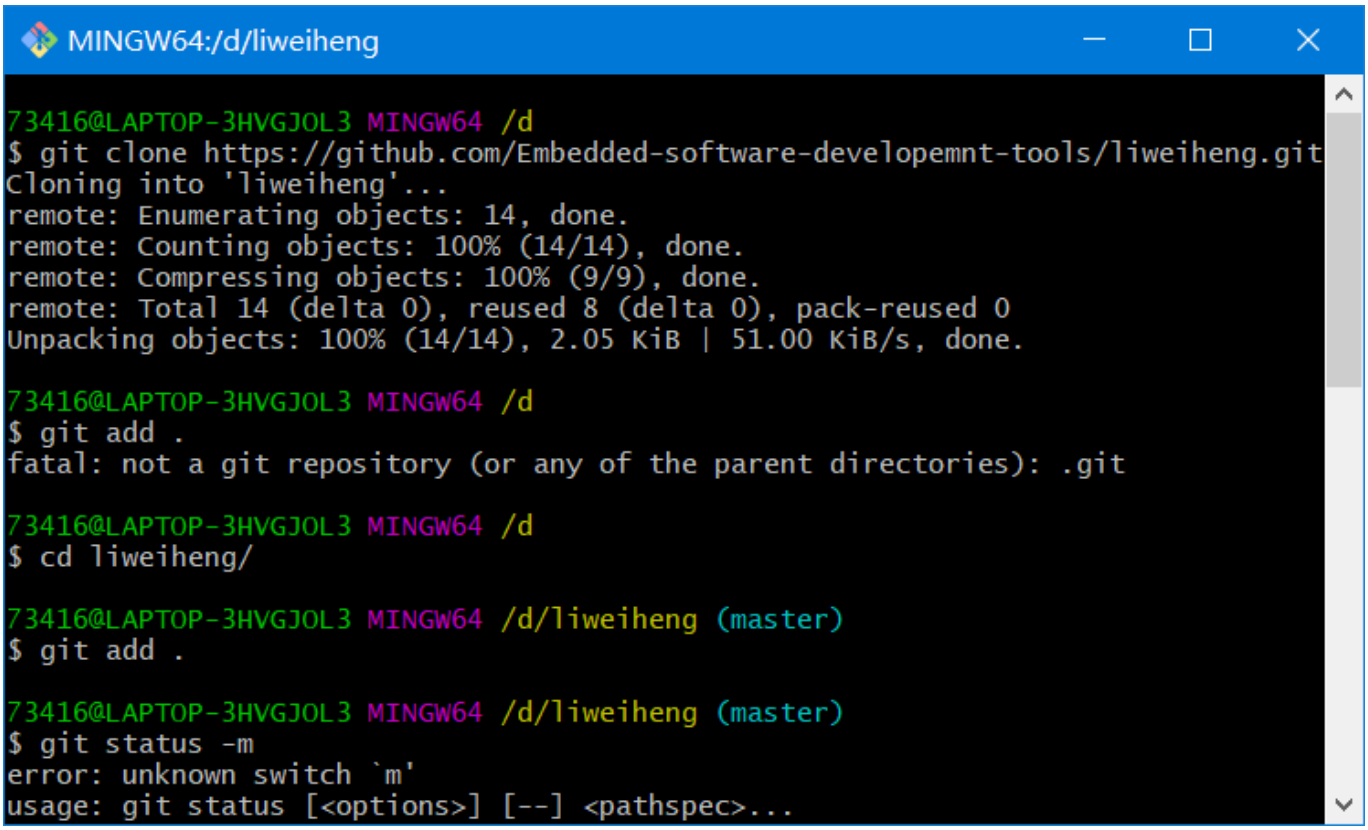
5. 创建小组git仓库并上传文件

创建的github仓库地址为https://github.com/Embedded-software-developemnt-tools/liweiheng

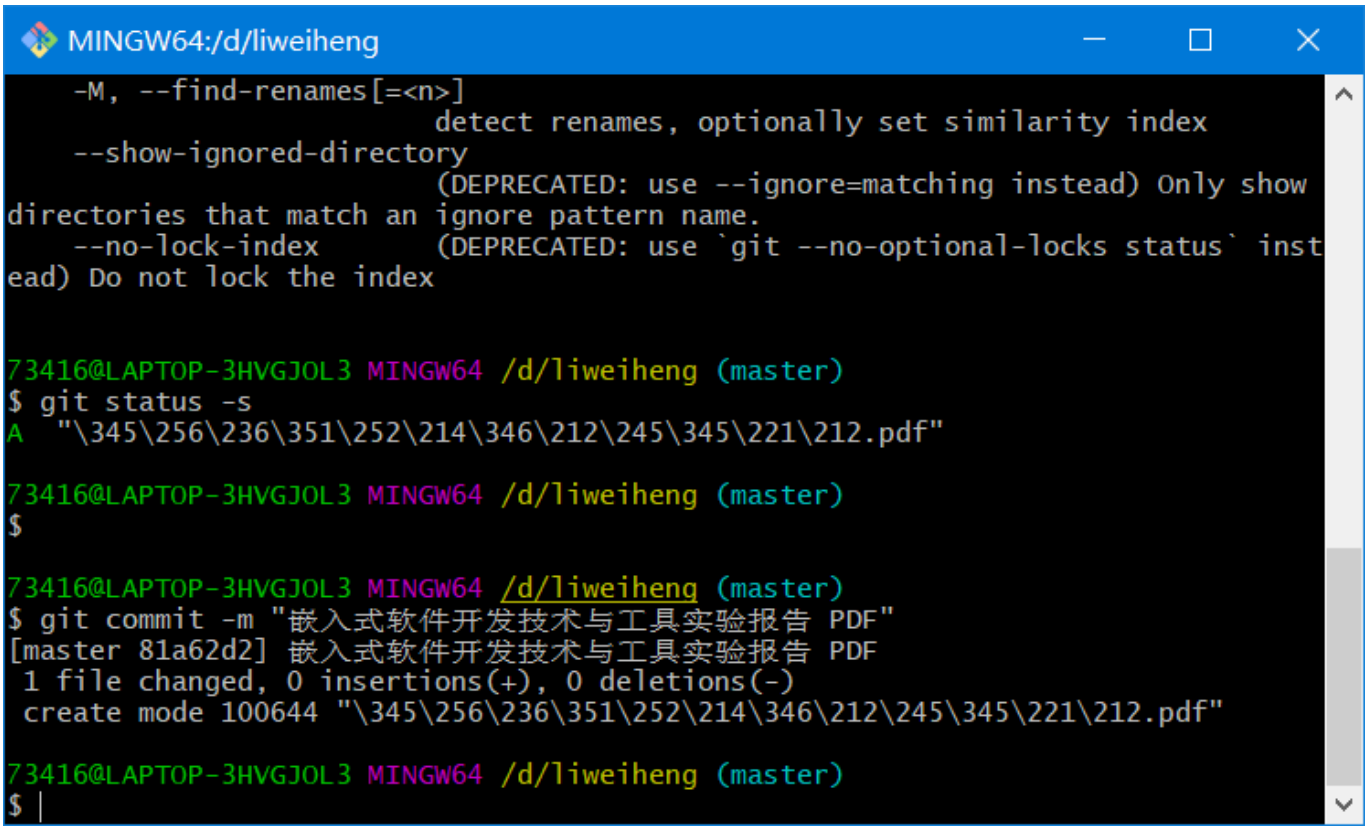
首先克隆到本地，在git命令行输入`git clone https://github.com/Embedded-software-developemnt-tools/liweiheng.git`



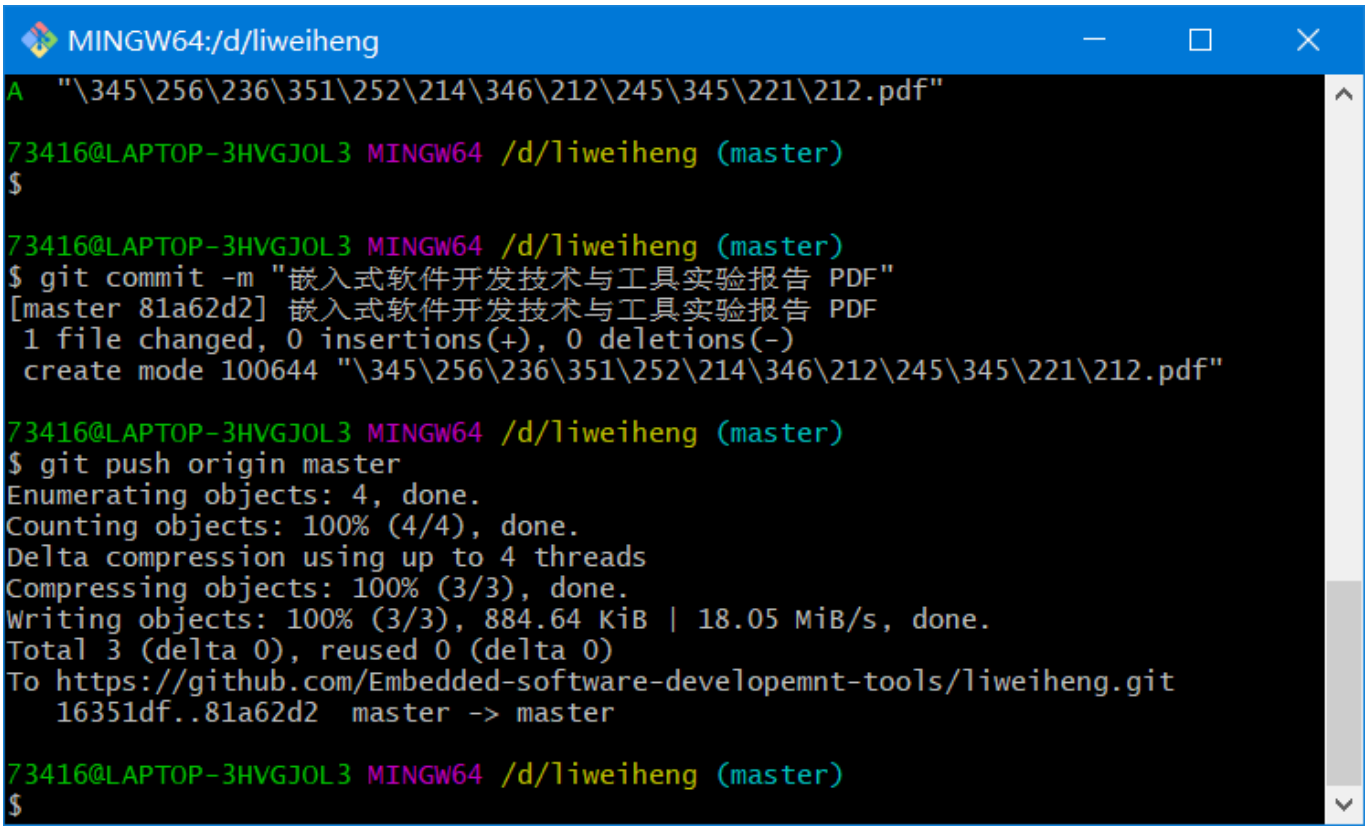
将实验报告复制到目录下，使用`git add .`提交到暂存区



`git commit -m "嵌入式软件开发技术与工具实验报告 PDF"`提交到本地版本库并添加说明



`git push origin master`提交到云端版本库，完成文件上传



网页端可观察到文件上传成功

四、实验总结

本次实验我在笔记本上安装了Ubuntu双系统，并实践了Makefile的使用和编写，利用automake工具自动生成Makefile，最终将文件上传至小组github仓库中。实验过程中也遇到了很多的问题，例如automake工具安装后报错，automake过程中由于示例程序使用了不同的库和头文件导致的Makefile生成失败等问题，在查阅各种资料后最终解决。在实验的过程中我熟练掌握了Makefile的各种用法以及github的用法，受益匪浅。