

## 实验 7 应用开发

### 一、实验目的

- 1、熟悉 gcc 工具。
- 2、掌握 gcc 编程方法。

### 二、预备知识

- 1、熟悉 Linux 下的 C 语言编程以及 GCC、GDB 工具的使用。
- 2、编辑、编译、运行、调试简易的程序，参考书中例子。
- 3、采用多个文件编写程序，并使用 Makefile 文件和 make 工具。

### 三、实验内容

- 1、解释 Linux 内核源代码的结构，以及根目录下的 Makefile 文件。

内核源代码目录在“/usr/src/kernels/”中（如果装有多内核，这里会有多个文件夹）。主要目录解释如下：

**arch:** 这个子目录包含了此核心源代码所支持的硬件体系结构相关的核心代码。

**ipc:** 此目录包含了核心的进程间通讯代码。

**include:** 这个目录包括了核心的大多数 include 文件。另外对于每种支持的体系结构分别有一个子目录。

**init:** 此目录包含核心启动代码。

**mm** 此目录包含所有的内存管理代码。与具体硬件体系结构相关的内存管理代码位于 arch/\*/mm 目录下，如对应于 X86 的就是 arch/i386/mm/fault.c。

**drivers:** 系统中所有的设备驱动都位于此目录中。它又进一步划分成几类设备驱动，每一种也有对应的子目录，如声卡的驱动对应于 drivers/sound。

**fs:** Linux 支持的文件系统代码。不同的文件系统有不同的子目录对应，如 ext2 文件系统对应的就是 ext2 子目录。

**kernel:** 主要核心代码。同时与处理器结构相关代码都放在 arch/\*/kernel 目录下。

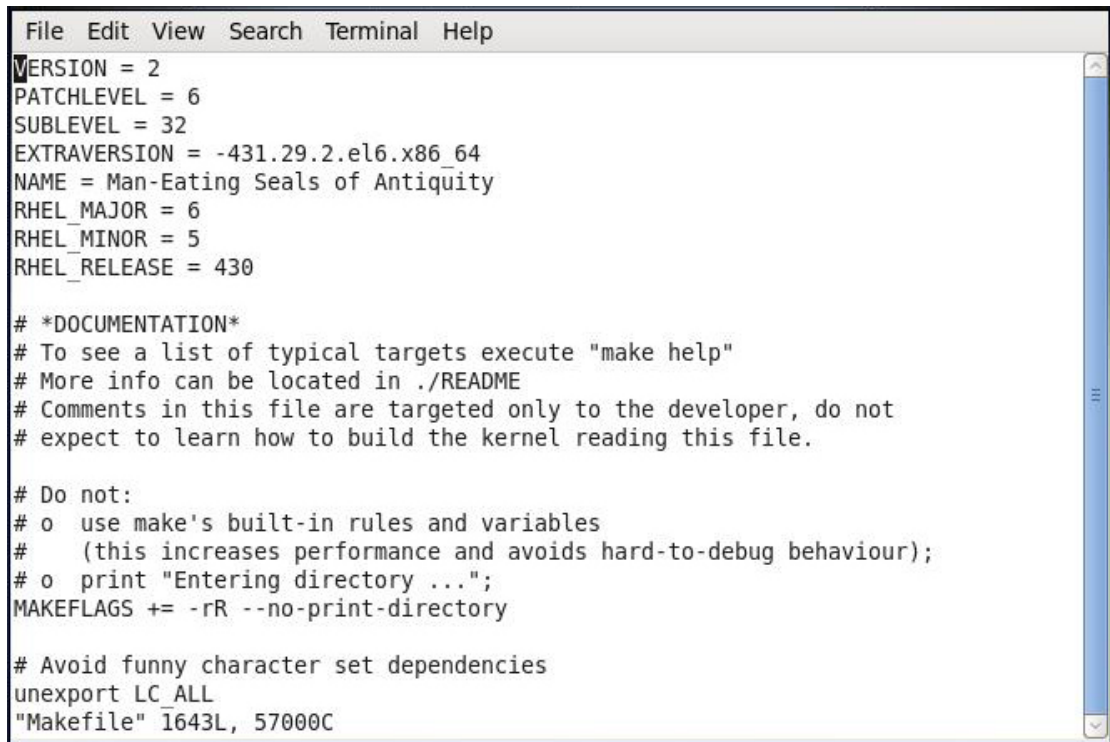
**net:** 核心的网络部分代码。里面的每个子目录对应于网络的一个方面。

**lib:** 此目录包含核心的库代码。与处理器结构相关库代码被放在 arch/\*/lib/目录下。

**scripts:** 此目录包含用于配置核心的脚本文件。

**documentation:** 此目录包含一些文档，起参考作用。

Makefile 文件定义 Linux 内核的编译规则，如图 E7-1 所示。



```
File Edit View Search Terminal Help
VERSION = 2
PATCHLEVEL = 6
SUBLEVEL = 32
EXTRAVERSION = -431.29.2.el6.x86_64
NAME = Man-Eating Seals of Antiquity
RHEL_MAJOR = 6
RHEL_MINOR = 5
RHEL_RELEASE = 430

# *DOCUMENTATION*
# To see a list of typical targets execute "make help"
# More info can be located in ./README
# Comments in this file are targeted only to the developer, do not
# expect to learn how to build the kernel reading this file.

# Do not:
# o use make's built-in rules and variables
#   (this increases performance and avoids hard-to-debug behaviour);
# o print "Entering directory ...";
MAKEFLAGS += -rR --no-print-directory

# Avoid funny character set dependencies
unexport LC_ALL
"Makefile" 1643L, 57000C
```

图 E7-1 Makefile 文件示例

VERSION=2, PATCHLEVEL=6, SUBLEVEL=32, EXTRAVERSION=-431.29.2.el6.x86\_64:  
它们共同构成内核的发行版本 KERNELRELEASE: 2.6.32-431.29.2.el6.x86\_64。

NAME =Man-Eating Seals of Antiquity : 给变量 NAME 赋值。

操作符“+=”是给变量 (“+=前面的 MAKEFLAGS”) 追加值。

如果变量 (“+=前面的 MAKEFLAGS”) 之前没有定义过, 那么 (“+=”) 会自动变成  
“=”。

如果变量 (“+=前面的 MAKEFLAGS”) 之前定义过, 那么“+=”会继承前次操作的赋值  
符。

如果前一次是“: =”, 那么“+=”会以“: =”作为赋值符。

在执行 make 时的命令行选项参数被通过变量 MAKEFAGS 传给子目录下的 make 程  
序。

对于这个变量除非使用指示符“unexport”对它们进行申明, 它们在整个 make 执行过程  
中始终被自动的传递给所有子 make。

还有个特殊变量 SHELL 和 MAKEFLAGS 一样, 默认情况下在整个 make 执行过程中被  
自动传递给所有子 make。

#Avoid funny character set dependencies

Unexport LC\_ALL: #非传递环境变量, 不会传递到下一级 Makefile。

## 2、查找 GCC 使用的库文件和头文件存放的路径。

使用 dpkg -S gcc 查找, 结果如图 7-2 所示, 其中的.h 文件即为头文件, .so 等文件即为  
库文件, .gz 文件则为压缩包文件。它们的存放路径如图 E7-2 所示。

```
File Edit View Search Terminal Help
jsj2018@donghua:~$ dpkg -S gcc
linux-headers-4.15.0-39-generic: /usr/src/linux-headers-4.15.0-39-generic/scripts/gcc-x86_32-has-stack-protector.sh
libgcc-5-dev:amd64: /usr/lib/gcc/x86_64-linux-gnu/5/include/avx512ifmavlintrin.h
libgcc-5-dev:amd64: /usr/lib/gcc/x86_64-linux-gnu/5/include/avx512ifmaintrin.h
linux-headers-4.15.0-29: /usr/src/linux-headers-4.15.0-29/scripts/gcc-plugins/gcc-generate-ipa-pass.h
libgcc-5-dev:amd64: /usr/lib/gcc/x86_64-linux-gnu/5/include/backtrace-supported.h
linux-headers-4.15.0-29: /usr/src/linux-headers-4.15.0-29/include/dt-bindings/clk/qcom,gcc-mdm9615.h
libgcc-5-dev:amd64: /usr/lib/gcc/x86_64-linux-gnu/5/libasan.so
libgcc-5-dev:amd64: /usr/lib/gcc/x86_64-linux-gnu/5/include/avx512cdintrin.h
libgcc-5-dev:amd64: /usr/lib/gcc/x86_64-linux-gnu/5/libitm.so
libgcc-5-dev:amd64: /usr/lib/gcc/x86_64-linux-gnu/5/include/clwbintrin.h
gcc-5: /usr/bin/x86_64-linux-gnu-gcc-ranlib-5
gcc-5: /usr/share/doc/gcc-5-base/sanitizer
libgcc1:amd64: /usr/share/lintian/overrides/libgcc1
libgcc-5-dev:amd64: /usr/lib/gcc/x86_64-linux-gnu/5/include/tmmintrin.h
libstdc++-5-dev:amd64, g++-5: /usr/share/doc/gcc-5-base/C++
language-pack-en-base: /usr/share/locale-langpack/en_GB/LC_MESSAGES/gcc-4.7.mo
gcc-5: /usr/share/doc/gcc-5-base/quadmath/changelog.gz
libgcc-5-dev:amd64: /usr/lib/gcc/x86_64-linux-gnu/5/crtprec32.o
language-pack-en-base: /usr/share/locale-langpack/en_AU/LC_MESSAGES/gcc-4.7.mo
```

图 E7-2 GCC 库文件与头文件存放路径

- 3、编写一个 C 程序，其功能是用户向系统输入一个简单的问候，然后系统以字母反序将该问候显示出来。要求：编写 Makefile 文件。

(1) 新建文件夹 demo1，在其中新建 xx.c 文件，如图 E7-3 所示。

```
File Edit View Search Terminal Help
#include <stdio.h>
#include <string.h>

int main()
{
    char str[10]="\0";
    gets(str);
    int n=0;
    n = strlen(str);

    for(;n>=0;n--)
    {
        printf("%c",str[n]);
    }
    return 0;
}
~
~
```

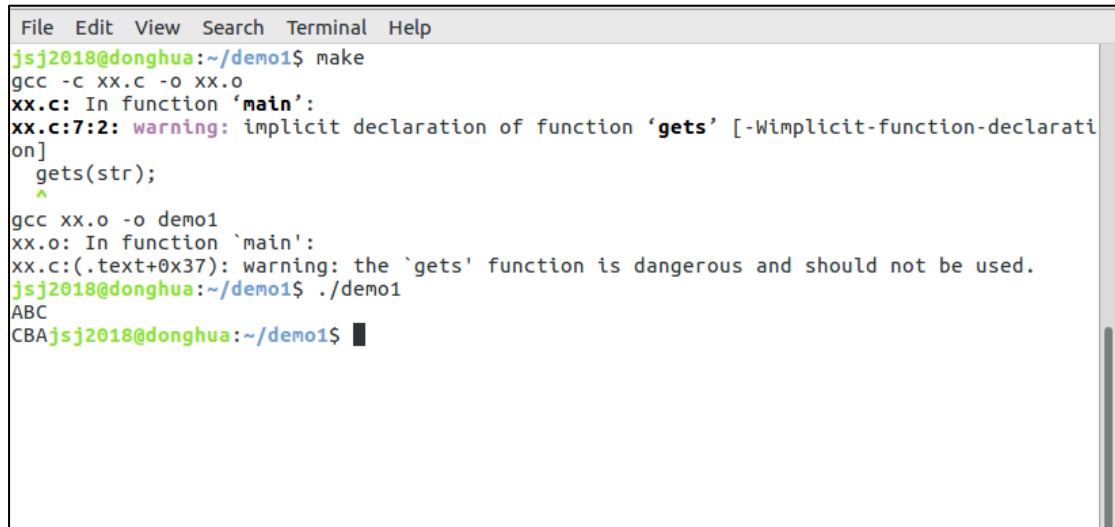
图 E7-3 C 程序示例之一

(2) 新建 Makefile 文件，vi Makefile，如图 E7-4 所示。

```
File Edit View Search Terminal Help
demo1 : xx.o
        gcc xx.o -o demo1
xx.o : xx.c
        gcc -c xx.c -o xx.o
clean:
        rm -fr *.o demo1
~
~
~
~
~
~
~
```

图 E7-4 编写 Makefile 文件

(3) 编译, 测试结果如图 E7-5 所示。

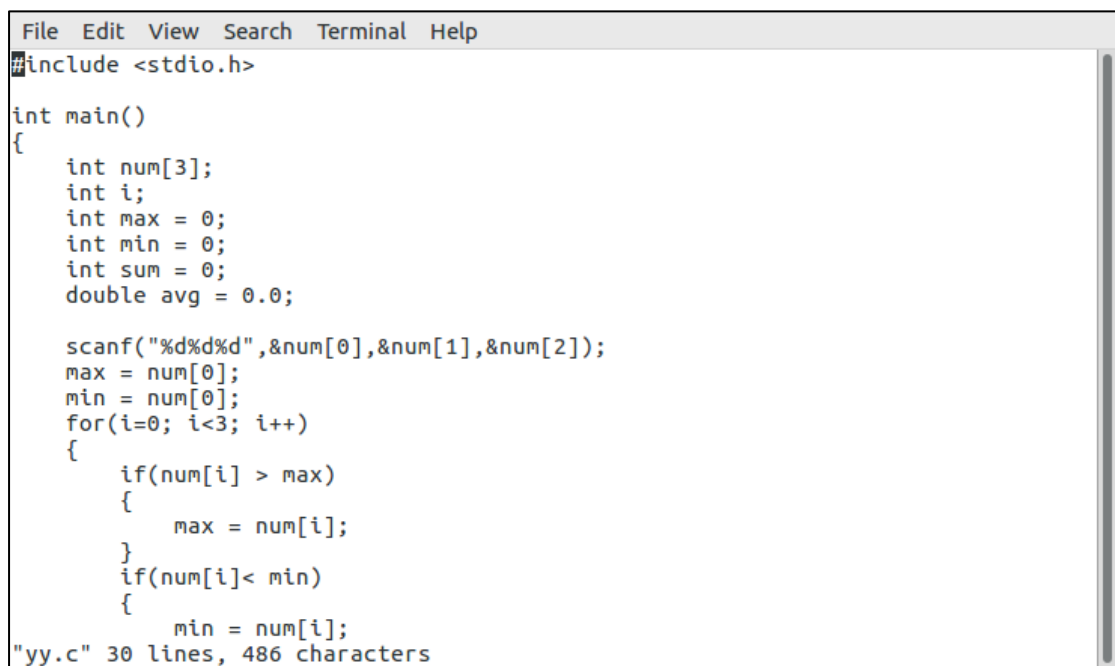


```
File Edit View Search Terminal Help
jsj2018@donghua:~/demo1$ make
gcc -c xx.c -o xx.o
xx.c: In function 'main':
xx.c:7:2: warning: implicit declaration of function 'gets' [-Wimplicit-function-declara
on]
  gets(str);
  ^
gcc xx.o -o demo1
xx.o: In function 'main':
xx.c:(.text+0x37): warning: the 'gets' function is dangerous and should not be used.
jsj2018@donghua:~/demo1$ ./demo1
ABC
CBAjsj2018@donghua:~/demo1$
```

图 E7-5 程序编译与运行

4、编写一个 C 程序, 其功能是从键盘输入三个整数, 然后系统输出最大值、最小值和平均值 (保留两位小数)。要求: 使用函数功能及 Makefile 文件。

(1) 新建 demo2 文件夹, 新建 yy.c 文件, 如图 E7-6 所示。



```
File Edit View Search Terminal Help
#include <stdio.h>

int main()
{
    int num[3];
    int i;
    int max = 0;
    int min = 0;
    int sum = 0;
    double avg = 0.0;

    scanf("%d%d%d",&num[0],&num[1],&num[2]);
    max = num[0];
    min = num[0];
    for(i=0; i<3; i++)
    {
        if(num[i] > max)
        {
            max = num[i];
        }
        if(num[i]< min)
        {
            min = num[i];
        }
    }
}
```

"yy.c" 30 lines, 486 characters

图 E7-6 C 程序示例之二

(2) 新建 Makefile 文件, vi Makefile, 如图 E7-7 所示。

```
File Edit View Search Terminal Help
demo2 : yy.o
        gcc yy.o -o demo2
yy.o : yy.c
        gcc -c yy.c -o yy.o
clean:
        rm -fr *.o demo2

~
~
~
~
~
~
```

图 E7-7 编写 Makefile 文件

(3) 编译，测试结果如图 E7-8 所示。

```
File Edit View Search Terminal Help
jsj2018@donghua:~/demo2$ vi yy.c
jsj2018@donghua:~/demo2$ vi makefile
jsj2018@donghua:~/demo2$ make
gcc -c yy.c -o yy.o
gcc yy.o -o demo2
jsj2018@donghua:~/demo2$ ./demo2
23
34
12
最大值：34
最小值：12
平均值：23.00
jsj2018@donghua:~/demo2$ █
```

图 E7-8 程序编译与运行

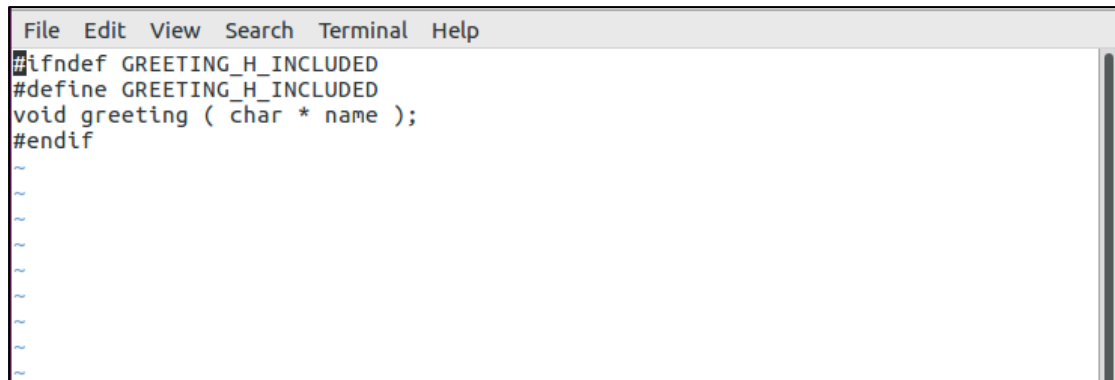
5. 编写一个多个文件的 c 程序, 其功能是从键盘输入一个字符串 name, 然后系统输出 Hello name[]!. 要求: 使用函数功能及 Makefile 文件。

(1) 新建 mypro 文件夹, 进入该文件夹, 如图 E7-9 所示。

```
File Edit View Search Terminal Help
root@donghua:~# mkdir mypro
root@donghua:~# cd mypro
root@donghua:~/mypro# █
```

图 E7-9 新建文件夹并进入

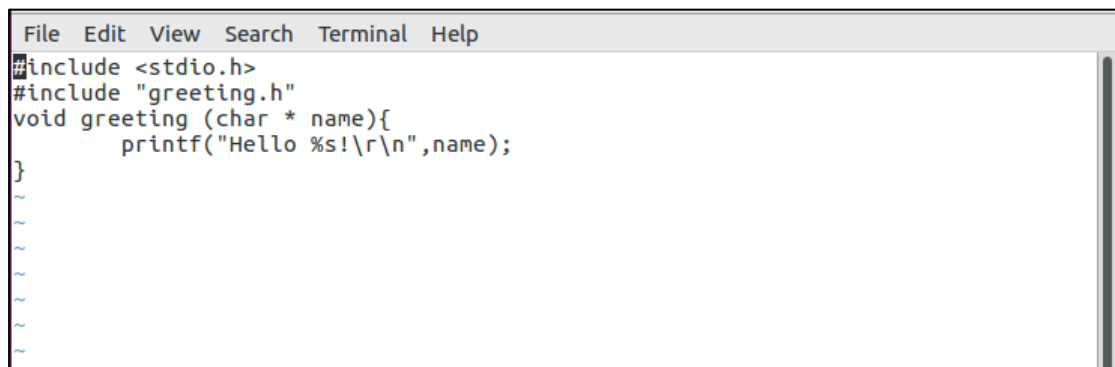
(2) 新建 greeting.h 文件, vi greeting.h, 如图 E7-10 所示。



```
File Edit View Search Terminal Help
#ifndef GREETING_H_INCLUDED
#define GREETING_H_INCLUDED
void greeting ( char * name );
#endif
~
~
~
~
~
~
~
```

图 E7-10 编写 greeting.h 文件

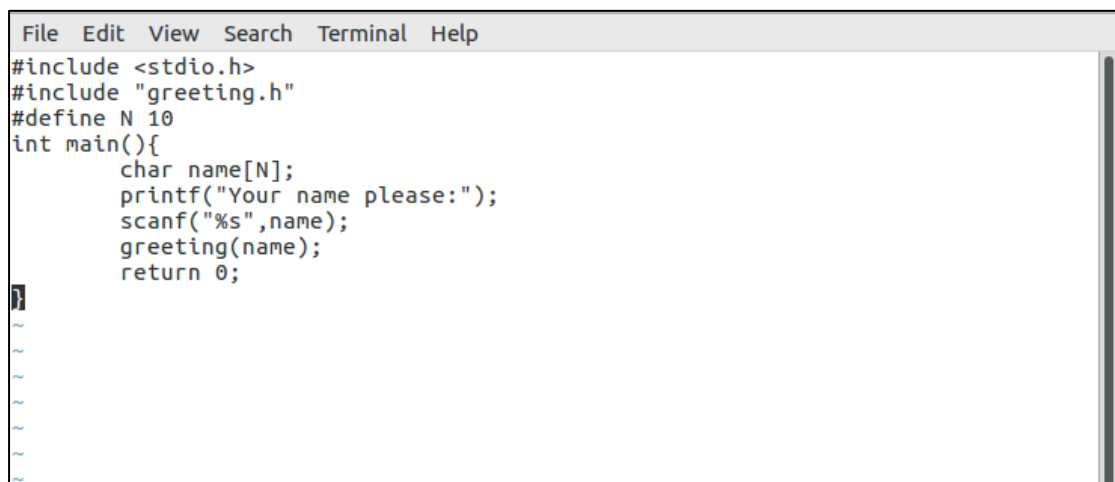
(3) 新建 greeting.c 文件, vi greeting.c, 如图 E7-11 所示。



```
File Edit View Search Terminal Help
#include <stdio.h>
#include "greeting.h"
void greeting (char * name){
    printf("Hello %s!\r\n",name);
}
~
~
~
~
~
~
~
```

图 E7-11 编写 greeting.c 文件

(4) 新建 my\_app.c 文件, 如图 E7-12 所示。



```
File Edit View Search Terminal Help
#include <stdio.h>
#include "greeting.h"
#define N 10
int main(){
    char name[N];
    printf("Your name please:");
    scanf("%s",name);
    greeting(name);
    return 0;
}
~
~
~
~
~
~
~
```

图 E7-12 编写 my\_app.c 文件

(5) 新建 Makefile 文件, 如图 E7-13 所示。

```
File Edit View Search Terminal Help
OBJS = my_app.o greeting.o
CC = gcc
CFLAGS = -Wall -O -g
my_app : $(OBJS)
    $(CC) $(OBJS) -o my_app
greeting.o: greeting.c greeting.h
    $(CC) $(CFLAGS) -c greeting.c
my_app.o : my_app.c greeting.h
    $(CC) $(CFLAGS) -c my_app.c -Ifunctions
clean:
    rm -f *.o my_app
```

图 E7-13 编写 Makefile 文件

(6) 编译，测试结果如图 E7-14 所示。

```
File Edit View Search Terminal Help
jsj2018@donghua:~/mypro$ make
gcc -Wall -O -g -c my_app.c -Ifunctions
my_app.c: In function 'main':
my_app.c:7:2: warning: ignoring return value of 'scanf', declared with attribute
warn_unused_result [-Wunused-result]
    scanf("%s",name);
    ^
gcc -Wall -O -g -c greeting.c
gcc my_app.o greeting.o -o my_app
jsj2018@donghua:~/mypro$ ./my_app
Your name please:turbo
Hello turbo!
jsj2018@donghua:~/mypro$
```

图 E7-14 程序编译与运行

#### 四、拓展

1、安装最新版的 JDK 和 Eclipse。

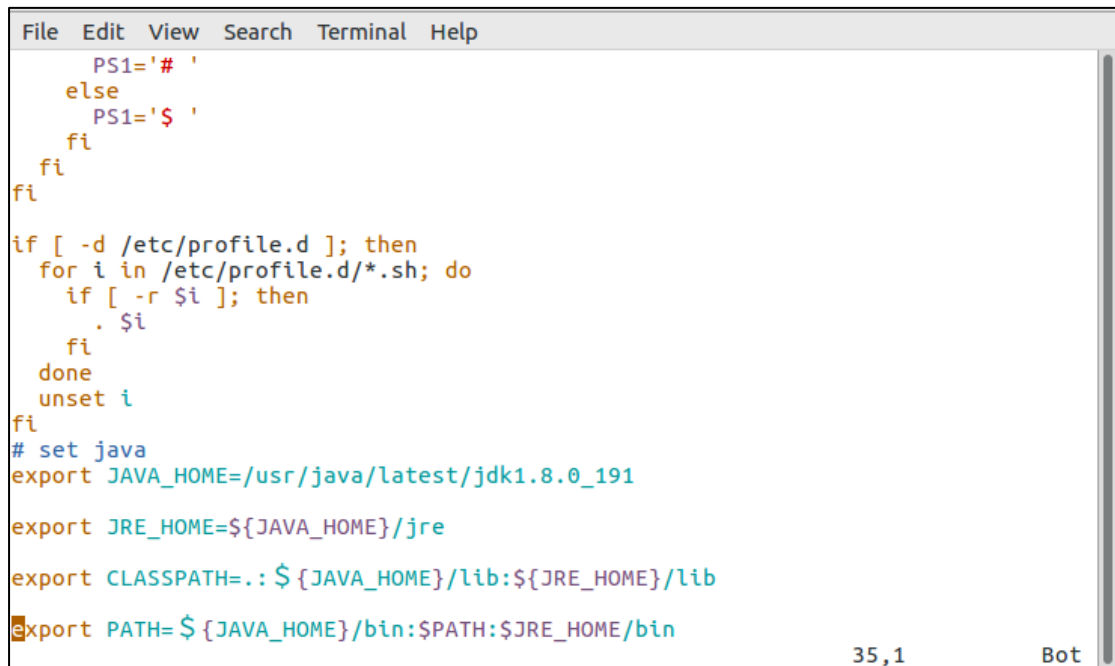
(1) 安装 JDK，网上下载 jdk 的 Linux 包，选择需要的 jdk 版本，网址如下：

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

下载 jdk-7u71-linux-x64.tar.gz 版本的 jdk，解压后放入/usr/java/latest 中。

(2) 配置 jdk 的环境变量，命令为# vi /etc/profile。

在文件末尾添加如图 E7-15 所示的信息。

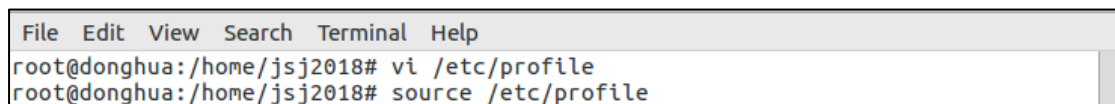


```
File Edit View Search Terminal Help
PS1='# '
else
PS1='$ '
fi
fi
fi
if [ -d /etc/profile.d ]; then
for i in /etc/profile.d/*.sh; do
if [ -r $i ]; then
. $i
fi
done
unset i
fi
# set java
export JAVA_HOME=/usr/java/latest/jdk1.8.0_191
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.: ${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH= ${JAVA_HOME}/bin:$PATH:$JRE_HOME/bin
```

35,1 Bot

图 E7-15 Java 配置

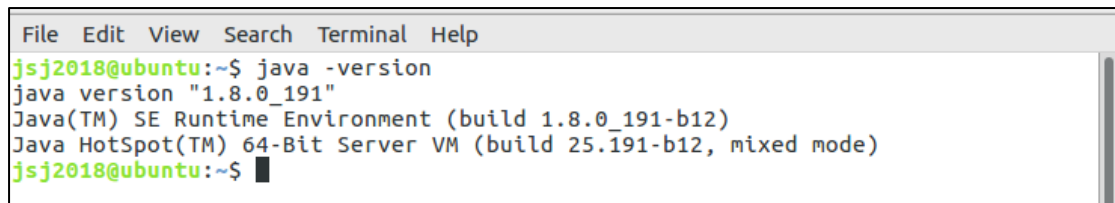
修改完后保存关闭，并输入以下命令使环境变量立即生效。



```
File Edit View Search Terminal Help
root@donghua:/home/jsj2018# vi /etc/profile
root@donghua:/home/jsj2018# source /etc/profile
```

图 E7-16 环境变量设置

(3) 输入命令 `java -version` 查看是否安装成功。



```
File Edit View Search Terminal Help
jsj2018@ubuntu:~$ java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)
jsj2018@ubuntu:~$
```

图 E7-16 查看 Java 版本

(4) 安装 Eclipse，从官网下载 <https://www.eclipse.org/downloads/> 下载需要的 Eclipse。解压后放入 `/usr/local/eclipse` 中。本实验选择的是 Eclipse IDE for Java Developers，下载完成，在 `/home/jsj2018/eclipse/java-2018-09/eclipse`，运行 `eclipse`。

2、在 Eclipse 中创建一个 Java 项目，利用版本管理器进行管理。

(1) 创建一个 Java 项目，打开 Eclipse，选择 `File > New > Other > Web > Dynamic Web Project` 即可。

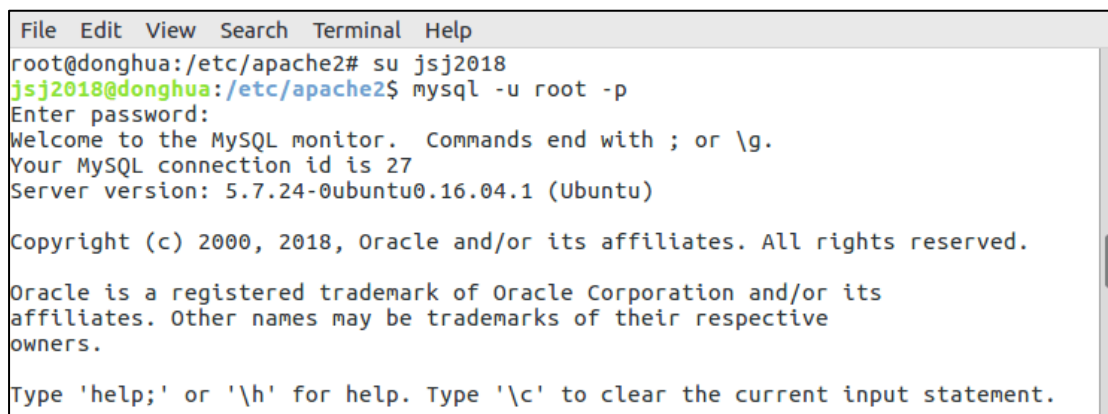
(2) 当前较为常用的版本管理有 Git 和 SVN，这里使用 SVN，在 eclipse 中安装 SVN 插件。可以直接在 Eclipse Marketplace 中找到 SVN 插件进行安装，也可以单独下载安装。推荐使用第一种方法。点击 `Help --> Eclipse Marketplace -->` 在 Find 中输入 `subclipse` 搜索 --> 找到 `subclipse` 点击 `install`，安装完成。

3、搭建 LAMP 平台，创建一个网站，可以实现学生信息的插入删除。

(1) 搭建一个 LAMP 平台，需要安装 PHP、MySQL、Apache，本实验使用 Ubuntu16.04，可以执行以下命令安装：



安装 Apache: `sudo apt-get install apache2`。  
安装 MySQL: `sudo apt-get install mysql-server mysql-client`。  
安装 PHP: `sudo apt-get install php7.0`。  
安装其他模块: `sudo apt-get install libapache2-mod-php7.0`。  
`sudo apt-get install php7.0-mysql`  
重启服务: `service apache2 restart`  
`service mysql restart`  
修改权限: `sudo chmod 777 /var/www`  
查看是否修改成功: `ls -al /var/www`  
安装 phpMyAdmin: `sudo apt-get install phpMyAdmin`  
启用 Apache `mod_rewrite` 模块: `sudo a2enmod rewrite`  
配置 Apache: `vim /etc/apache2/apache2.conf`  
重启服务: `service apache2 restart`  
(2) 连接数据库, 实现插入功能。  
切换到 jsj2018 用户 : `su jsj2018`  
进入 mysql: `mysql -u root -p`

A terminal window with a menu bar (File, Edit, View, Search, Terminal, Help). The user is at the prompt 'root@donghua:/etc/apache2#' and runs 'su jsj2018'. The prompt changes to 'jsj2018@donghua:/etc/apache2\$'. The user runs 'mysql -u root -p'. The terminal shows the MySQL login sequence: 'Enter password:', 'Welcome to the MySQL monitor. Commands end with ; or \g.', 'Your MySQL connection id is 27', 'Server version: 5.7.24-0ubuntu0.16.04.1 (Ubuntu)', 'Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.', 'Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.', and 'Type \'help;\' or \'h\' for help. Type \'c\' to clear the current input statement.'

```
File Edit View Search Terminal Help
root@donghua:/etc/apache2# su jsj2018
jsj2018@donghua:/etc/apache2$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 27
Server version: 5.7.24-0ubuntu0.16.04.1 (Ubuntu)

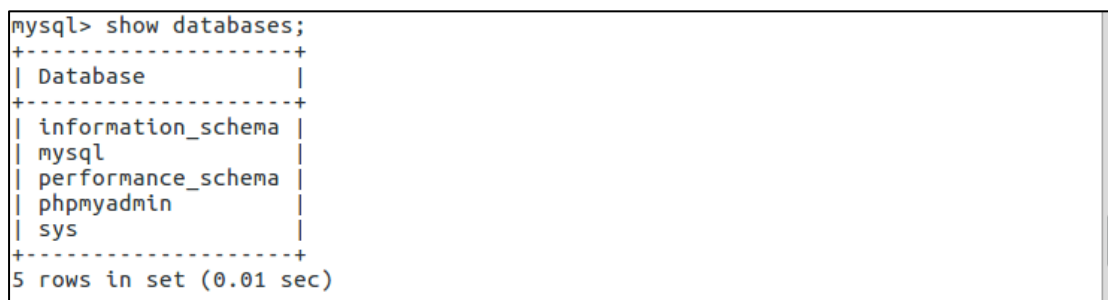
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or 'h' for help. Type 'c' to clear the current input statement.
```

图 E7-17 进入 mysql 数据库

查看当前数据库: `show databases`


A terminal window showing the output of the 'show databases;' command. The output is a table with one column 'Database' and five rows: 'information\_schema', 'mysql', 'performance\_schema', 'phpmyadmin', and 'sys'. The terminal also shows '5 rows in set (0.01 sec)' at the bottom.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| sys |
+-----+
5 rows in set (0.01 sec)
```

图 E7-18 显示数据库信息

在 test 数据库下查看表: `use test;`

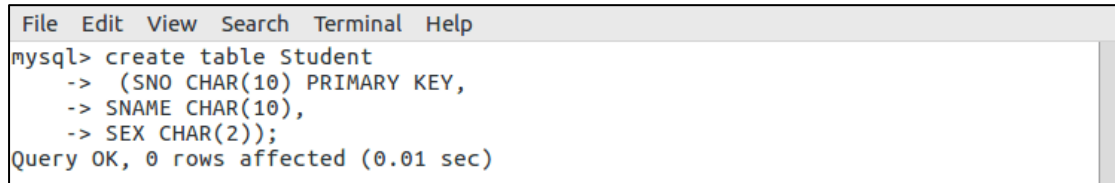
`Show tables;`

A terminal window showing the output of 'use test;' and 'show tables;'. The output shows 'Database changed' and 'Empty set (0.00 sec)' respectively.

```
File Edit View Search Terminal Help
mysql> use test;
Database changed
mysql> show tables;
Empty set (0.00 sec)
```

图 E7-19 查看表

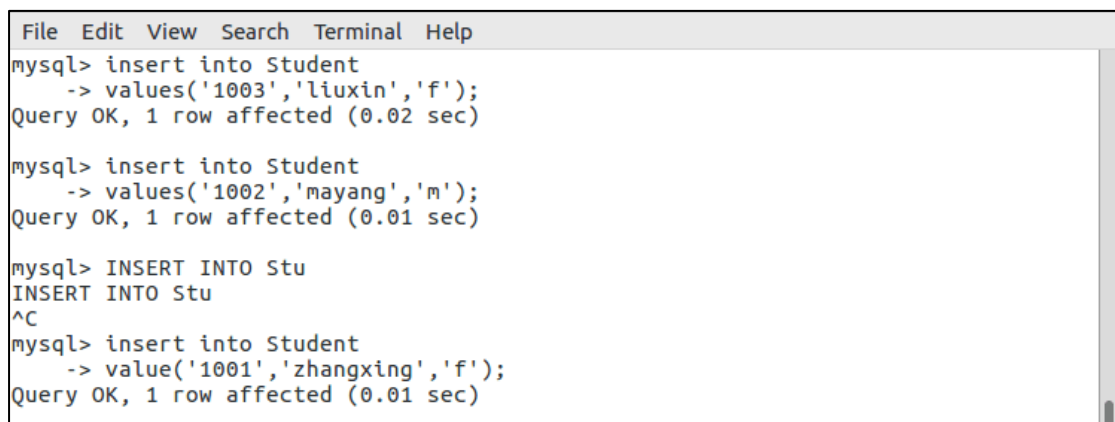
建立新表 (student) :create table student  
(SNO CHAR(10) PRIMARY KEY,  
SNAME CHAR(10),  
SEX CHAR(2));



```
File Edit View Search Terminal Help
mysql> create table Student
-> (SNO CHAR(10) PRIMARY KEY,
-> SNAME CHAR(10),
-> SEX CHAR(2));
Query OK, 0 rows affected (0.01 sec)
```

图 E7-20 新建表

插入数据如图 E7-21 所示: insert into student  
Values( '1003' , ' liuxing' , ' f' );




```
File Edit View Search Terminal Help
mysql> insert into Student
-> values('1003','liuxin','f');
Query OK, 1 row affected (0.02 sec)

mysql> insert into Student
-> values('1002','mayang','m');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Stu
INSERT INTO Stu
^C
mysql> insert into Student
-> value('1001','zhangxing','f');
Query OK, 1 row affected (0.01 sec)
```

图 E7-21 在表中插入数据

编写 PHP 页面代码, 如图 E7-22 所示。



```
aa.php
/var/www/html
Save

<p>
<center>
<?php

/* Connect to a MySQL server 连接数据库服务器 */
$link = mysqli_connect(
    'localhost', /* The host to connect to 连接MySQL地址 */
    'root', /* The user to connect as 连接MySQL用户名 */
    '123456', /* The password to use 连接MySQL密码 */
    'test'); /* The default database to query 连接数据库名称*/

if (!$link) {
    printf("Can't connect to MySQL Server. Errorcode: %s ", mysqli_connect_error());
    exit;
}else
    echo '数据库连接上了!'. "<br/>";

if ($result = mysqli_query($link, 'SELECT * FROM Student')) {
echo('SNO    SNAME    SEX '). "<br/>";
```

图 E7-22 编写数据库连接代码

执行结果如图 E7-23 所示。



图 E7-23 在 PHP 页面中显示数据库的数据