桂林电子科技大学

**实验三 服务器的配置与管理**  实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | **实验3 服务器的配置与管理** | | | | | | | |  | 辅导员意见：  成绩 辅导员  签 名 |
| 院 系 | 计算机与信息安全学院 | | | 专业 | | 计算机科学与技术 | | |
| 学 号 | 1600300128 | | | 姓名 | | 王涛 | | |
| 实验日期 | 2018 | 年 | 12 | | 月 | | 1 | 日 |
|  |  | | | | | | | |

## 实验目的

1、 掌握OPENSSH网络服务的安装、配置和使用。

2、 掌握Apache Web服务的安装、配置和使用。

## 实验要求

1、 要求采用源码安装OPENSSH服务；要求理解源码编译选项含义并正确使用合适的选项。

2、 要求能够阅读源码中的INSTALL文件或README等相关文档，并初步了解Makefile文件作用。

3、 在线安装Apache服务器程序，并进行配置。

## 实验步骤

### 一、SSH服务的源码安装、启动和退出

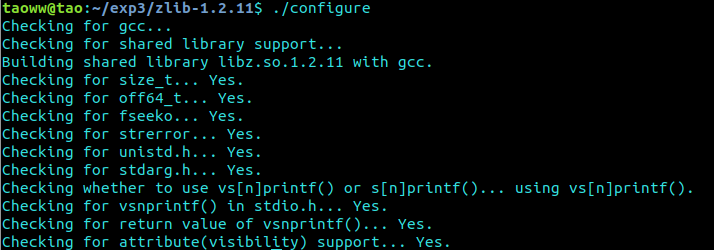
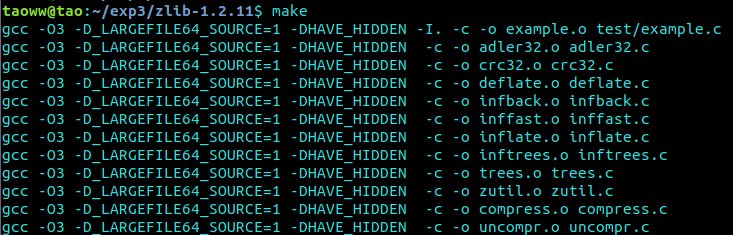
1、源码包安装的一般步骤

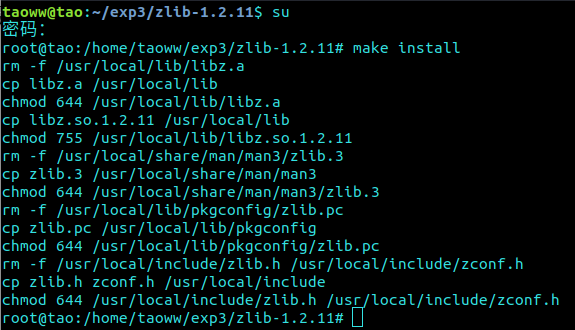
* 1. 解压源码包至一个单独目录
  2. 阅读INSTALL及README文件，了解安装步骤
  3. 运行./configure或config脚本对安装环境进行检测（运行./configure–help可查看./configure脚本的相关参数）
  4. 如果c步骤没有问题，则可运行make对源码进行编译；如果c步骤提示出现依赖关系则需要解决相关依赖关系。
  5. 如果d步骤没有问题，则可切换到root账户（注使用su命令时，不需要加-参数），并运行make install命令安装相应的库

2、本节需要的源码包括三个压缩包，首先都要进行解压，并阅读openssh的INSTALL。

openssh-7.7p1.tar； openssl-1.0.2g.tar ；zlib-1.2.11.tar

1. 编译安装zlib



1. 编译安装openssl

配置选项

./config --prefix=/usr/local/openssl --openssldir=/usr/local/openssl --shared



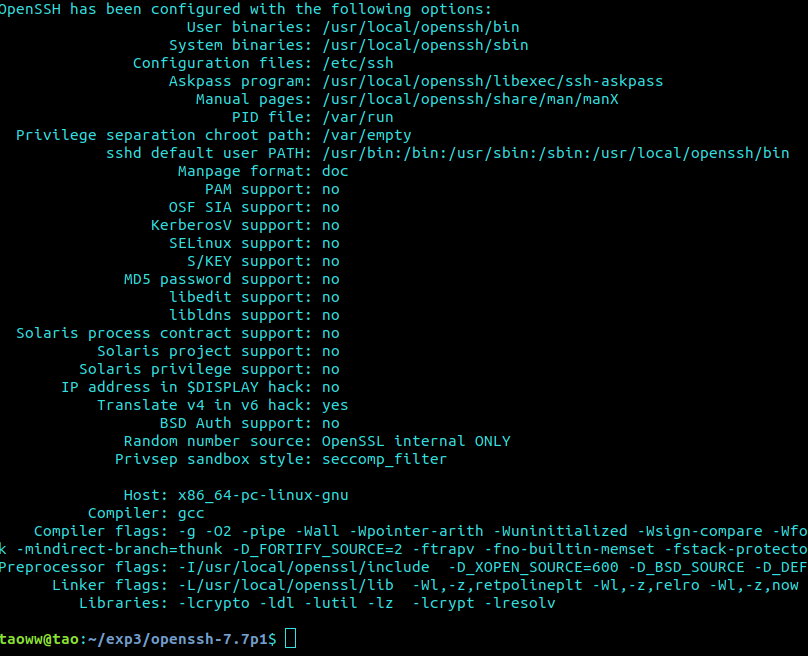




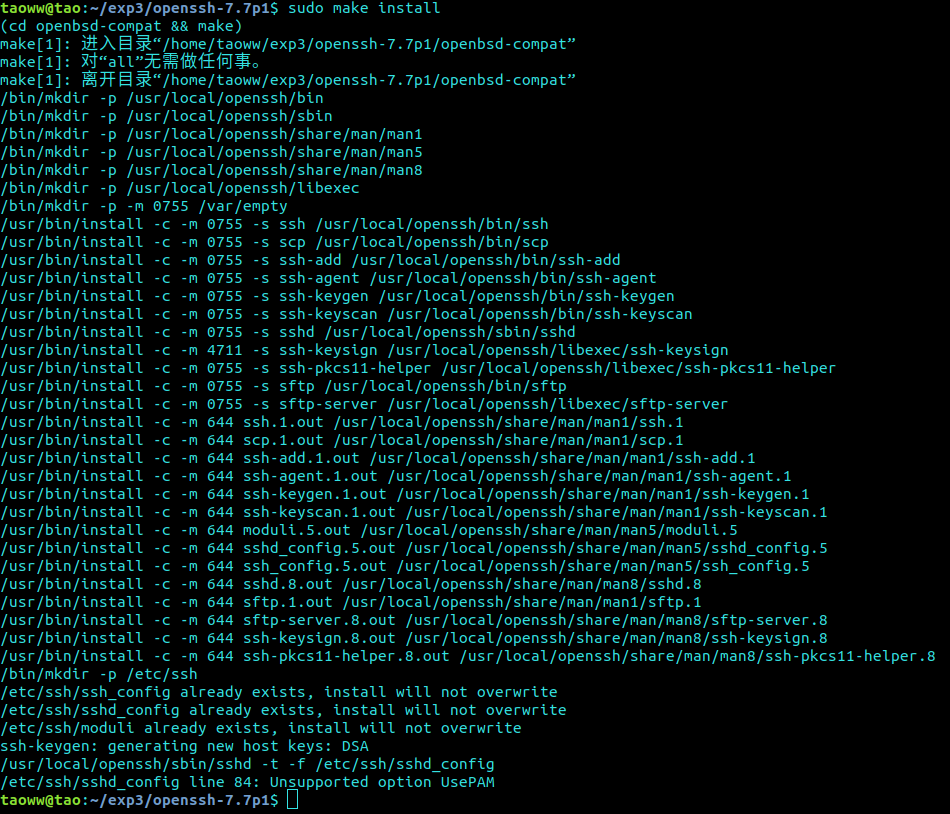


1. 编译安装openssh

./configure --prefix=/usr/local/openssh --sysconfdir=/etc/ssh --with-ssl-dir=/usr/local/openssl --without-openssl-header-check







如果有以下错误提示，

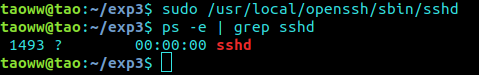
Privilege separation user sshd does not exist

解决办法：编辑/etc/passwd ，添加一项：

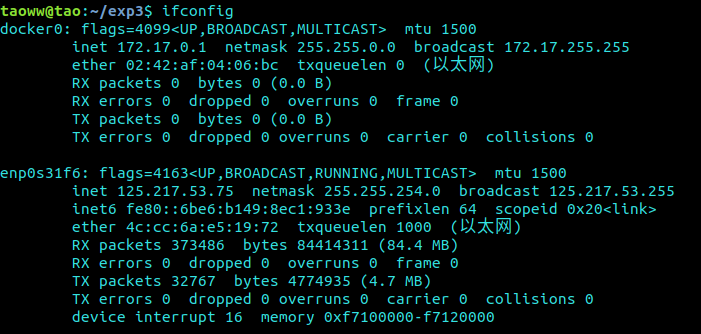
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin

3、启动：sudo /usr/local/openssh/sbin/sshd&（需要绝对路径）

查看ssh服务是否启动：sudo ps –e | grep ssh



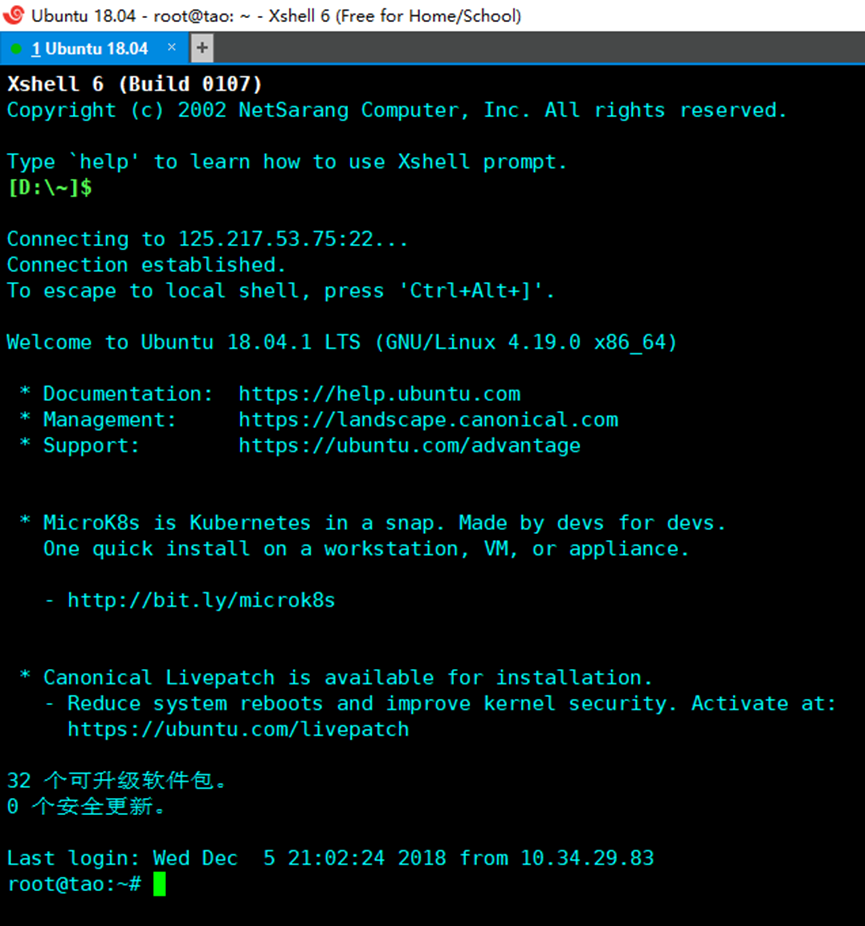
4、通过ifconfig命令查看linux服务器ip地址：ifconfig



IP为125.217.53.75

5、windows客户端使用SSH登录

使用Xshell软件登录linux

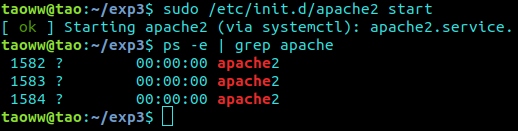


### 二、Apache服务安装和配置

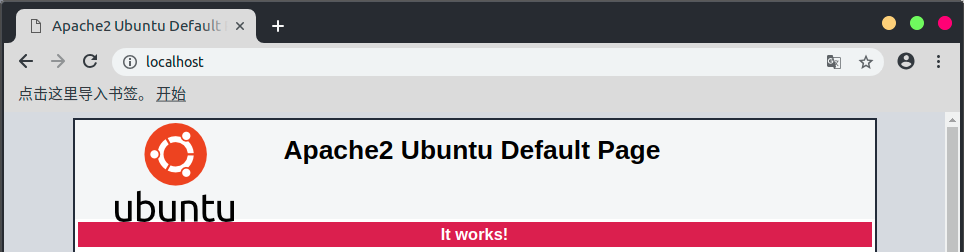
* 安装Apache2，安装命令：sudo apt-get install apache2
* 使用vi修改配置文件：sudo vim /etc/apache2/apache2.conf
* 进入编辑界面，在最后一行加上一句话：ServerName localhost:80



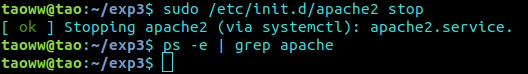
* 启动apache服务，输入指令：sudo /etc/init.d/apache2 start



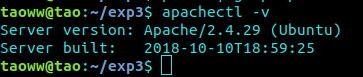
* 测试，在浏览器中输入：[http://localhost](http://localhost/)



* 停止服务，输入命令：sudo /etc/init.d/apache2 stop



* 查看安装的版本：apachectl -v



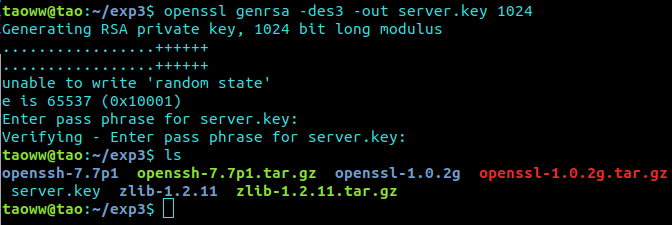
* 增加ssl支持
  + 安装openssl，一般默认安装了，位置 /usr/bin/openssl，如果openssl命令不执行，则安装：sudo apt-get install openssl
  + apache2.conf中设置SSL模块加载, 添加一行内容：

LoadModule ssl\_module /usr/lib/apache2/modules/mod\_ssl.so



* + 使用openssl创建自签名证书，在当前目录下生成私钥文件server.key，des3加密私钥

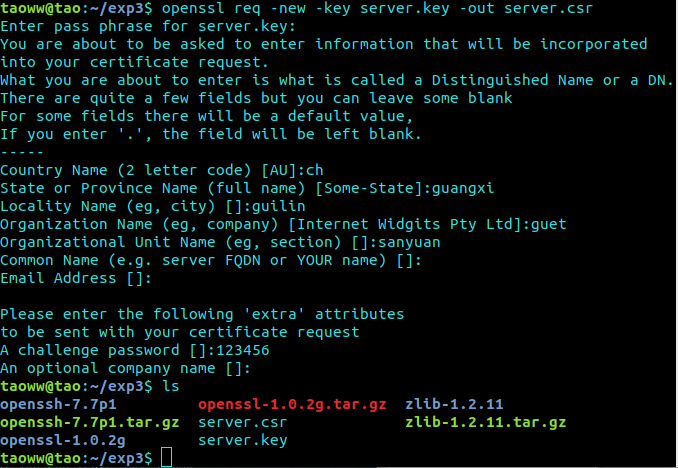
openssl genrsa -des3 -out server.key 1024



* + 生成证书请求文件CSR（Certificate Signing Request）

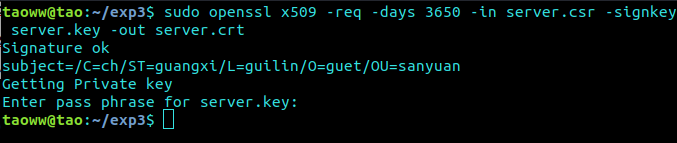
openssl req -new -key server.key -out server.csr

接下来输入一系列信息



* + 自己签发证书：

sudo openssl x509 -req -days 3650 -in server.csr -signkey server.key -out server.crt



* + 复制到SSL目录

cp server.crt /etc/ssl/certs

cp server.key /etc/ssl/private



* + 修改配置文件

ln -s /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-enabled/default-ssl.conf



vim /etc/apache2/sites-enabled/default-ssl.conf

添加：ServerName zrx.me主机名

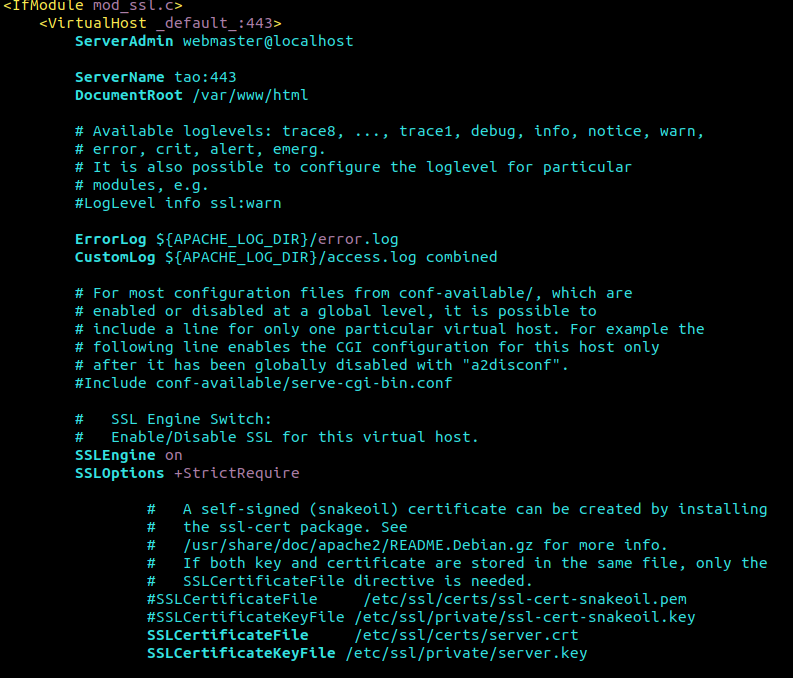
* + 在DocumentRoot中加入内容

SSLEngine On

SSLOptions +StrictRequire

SSLCertificateFile /etc/ssl/certs/server.crt

SSLCertificateKeyFile /etc/ssl/private/server.key

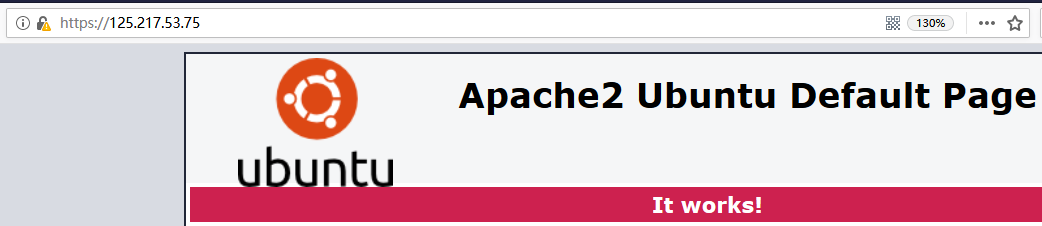


* + 重启apache

sudo /etc/init.d/apache2 start



https访问主机IP结果如下：

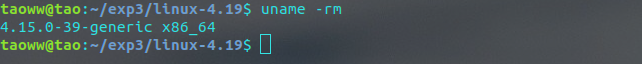


参考连接：<https://www.cnblogs.com/yongpan/p/8000725.html>

### 三、实验扩展：linux内核编译

参考步骤如下：

1. 使用 **uname –rm** 或 **uname –a** 命令查看当前系统的Linux内核版本。



2、下载内核压缩包并解压到当前目录。

3、**make mrproper** 把原来编译产生的垃圾删除

4、配置内核可以根据需要与爱好使用下面命令中的一个：

make config （基于文本的最为传统的配置界面）

**make menuconfig**（基于文本选单的配置界面）

make xconfig （基于图形窗口模式的配置界面）

make oldconfig （如果只想在原来内核配置的基础上修改一些小地方，会省去不少麻烦）

进行配置时，大部分选项可以使用其缺省值，只有小部分需要根据用户不同的需要选择。例如，如果需要内核支持DOS分区的文件系统，则要在文件系统部分选择FAT或DOS系统支持；系统如果配有网卡、PCMCIA卡等，需要在网络配置中选择相应卡的类型。

选择相应的配置时，有三种选择，它们分别代表的含义如下：

“Y”－ 将该功能编译进内核

“N”－ 不将该功能编译进内核

“M”－ 将该功能编译成可以在需要时动态插入到内核中的模块。

将与核心其它部分关系较远且不经常使用的部分功能代码编译成为可加载模块，有利于减小内核的长度，减小内核消耗的内存，简化该功能相应的环境改变时对内核的影响。许多功能都可以这样处理，例如像上面提到的网卡的支持、对FAT等文件系统的支持。

5、 **make dep** (确保关键文件在正确的位置)

这一项在Linux内核2.5.\*-2.6.0都不需要了。其实2.4.\*某些内核也不需要了。

6、 **make clean** (确保所有有关文件都处于最新版本状态)

7、 make zImage (编译压缩形式的内核)

在需要内核支持较多的外设和功能时，内核可能变得很大，此时可以编译大内核：

**make bzImage**

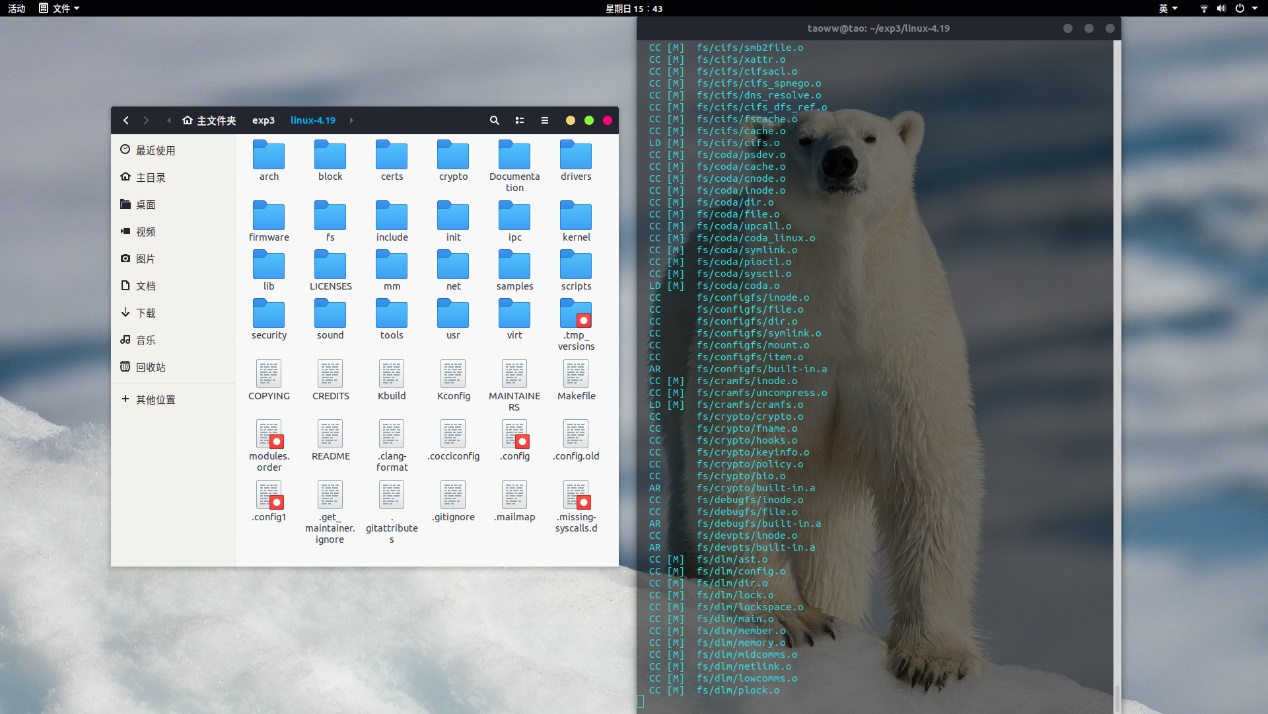
编译的时间与机器的硬件条件及内核的配置等因素有关，所获得的内核的位置在**./arch/i386/boot**目录下，当然这里假设用户的CPU是x86型的。

8、如果选择了可加载模块，编译完内核后，要对选择的模块进行编译：

**make modules** (编译选择的模块)

**make module\_install** (将编译后的模块转移到系统标准位置)

9、 **make install**



上面的命令"make install"将:

(1)把压缩内核映象拷贝到/boot目录下, 并创建相应的System.map符号链接;

(2)修改bootloader的配置文件;

(3)调用mkinitrd程序创建内核的initrd映象.

10、 用编辑器查看/boot/grub/grub.conf配置文件，将其中的**hiddenmenu**行加上注释标志**#**

11、 重新启动Linux系统，注意观察启动后出现的菜单选择项，选择刚编译好的新内核启动。

12、启动完成后，使用 **uname –rm** 或 **uname –a** 命令查看当前系统的Linux内核版本。



## 问题记录和实验总结

问题记录：

1.源码安装缺少依赖。

解决：通常是因为缺少gcc引起的编译失败，使用sudo apt-get install gcc安装gcc即可。另外实验中源码安装的三个包安装顺序不能变，需先安装依赖包，不然会出现错误，在源码安装过程先config或configure检查然后make最后make install，其中make install需要往往root权限，要加上sudo。

2.使用root远程登陆被拒绝。

解决：这个问题是因为安装后配置文件默认拒绝root登陆，打开/etc/ssh/sshd\_config查看是否有PermitRootLogin yes，没有加上即可。

3.Apache配置ssl之后使用https无法访问。

解决：补充如下命令：

（1）开启ssl模块：sudo a2enmod ssl；

（2）启动ssl站点：sudo a2ensite default-ssl；

（3）加入监听端口 443：打开/etc/apache2/ports.conf ，Listen 80改为Listen 80 443；

（4）配置 HTTPS 强制跳转

打开/etc/apache2/sites-available/000-default.conf

在<VirtualHost \*:80></VirtualHost>标签内加入以下三行

<VirtualHost \*:80>

RewriteEngine on

RewriteCond %{HTTPS} !=on

RewriteRule ^(.\*) https://%{SERVER\_NAME}$1 [L,R]

</VirtualHost>

（5）sudo a2enmod rewrite；

（6）重启Apache服务。

4.Linux内核编译缺少依赖。

解决：缺少的依赖包，会在报错信息中提示，例如缺少libncurses5-dev、libssl-dev使用apt安装即可。另外虚拟机linux系统安装内核时要确保内容够大，不然可能会崩溃。

本次实验进行了SSH服务、Apache服务的配置以及linux内核的编译。不同于简单的apt安装，使用码源安装SSH让我知道了安装的更具体的细节，以及包之间依赖关系的重要性。在操作过程中遇到了很多问题，不断的搜索解决之后对安装流程步骤基本熟悉。安装之前一定要看安装文档知道安装所需依赖以及相关配置信息。这样可以减少很多不知名的错误。安装完SSH服务之后便可以用远程连接控制主机，也需要配置一些网络，简单的可以同时连校园网就可以进行远程控制。