内核编译

（1）实验目的：正确的合理地设置内核编译配置选项，进一步理解Linux 系统的启动过程。

（2）实验内容：

编译Linux2.6.29内核，使系统可以正常加载.

1. 实验提示：

一、内核简介   
　　内核，是一个操作系统的核心。它负责管理系统的进程、内存、设备驱动程序、文件和网络系统，决定着系统的性能和稳定性。   
　　Linux的一个重要的特点就是其源代码的公开性，所有的内核源程序都可以在/usr/src/linux下找到，大部分应用软件也都是遵循GPL而设计的，你都可以获取相应的源程序代码。全世界任何一个软件工程师都可以将自己认为优秀的代码加入到其中，由此引发的一个明显的好处就是Linux修补漏洞的快速以及对最新软件技术的利用。而Linux的内核则是这些特点的最直接的代表。   
　　想象一下，拥有了内核的源程序对你来说意味着什么？首先，我们可以了解系统是如何工作的。通过通读源代码，我们就可以了解系统的工作原理，这在Windows下简直是天方夜谭。其次，我们可以针对自己的情况，量体裁衣，定制适合自己的系统，这样就需要重新编译内核。在Windows下是什么情况呢？相信很多人都被越来越庞大的Windows整得莫名其妙过。再次，我们可以对内核进行修改，以符合自己的需要。这意味着什么？没错，相当于自己开发了一个操作系统，但是大部分的工作已经做好了，你所要做的就是要增加并实现自己需要的功能。在Windows下，除非你是微软的核心技术人员，否则就不用痴心妄想了。

二、内核版本号

　　由于Linux的源程序是完全公开的，任何人只要遵循GPL，就可以对内核加以修改并发布给他人使用。Linux的开发采用的是集市模型（bazaar，与cathedral--教堂模型--对应），为了确保这些无序的开发过程能够有序地进行，Linux采用了双树系统。一个树是稳定树（stable tree），另一个树是非稳定树（unstable tree）或者开发树（development tree）。一些新特性、实验性改进等都将首先在开发树中进行。如果在开发树中所做的改进也可以应用于稳定树，那么在开发树中经过测试以后，在稳定树中将进行相同的改进。一旦开发树经过了足够的发展，开发树就会成为新的稳定树。开发数就体现在源程序的版本号中；源程序版本号的形式为x.y.z：对于稳定树来说，y是偶数；对于开发树来说，y比相应的稳定树大一（因此，是奇数）。到目前为止，稳定树的最高版本是2.4.18；开发树的最新版本是2.5.10。下载内核版本请访问[http://www.kernel.org](http://www.kernel.org/)。

三、为什么重新编译内核

　　Linux作为一个自由软件，在广大爱好者的支持下，内核版本不断更新。新的内核修订了旧内核的bug，并增加了许多新的特性。如果用户想要使用这些新特性，或想根据自己的系统度身定制一个更高效，更稳定的内核，就需要重新编译内核。   
　　通常，更新的内核会支持更多的硬件，具备更好的进程管理能力，运行速度更快、 更稳定，并且一般会修复老版本中发现的许多漏洞等，经常性地选择升级更新的系统内核是Linux使用者的必要操作内容。   
　　为了正确的合理地设置内核编译配置选项，从而只编译系统需要的功能的代码，一般主要有下面四个考虑：   
　　l 自己定制编译的内核运行更快（具有更少的代码）   
　　l 系统将拥有更多的内存（内核部分将不会被交换到虚拟内存中）   
　　l 不需要的功能编译进入内核可能会增加被系统攻击者利用的漏洞   
　　l 将某种功能编译为模块方式会比编译到内核内的方式速度要慢一些

四、内核编译模式

　　要增加对某部分功能的支持，比如网络之类，可以把相应部分编译到内核中（build-in），也可以把该部分编译成模块（module），动态调用。如果编译到内核中，在内核启动时就可以自动支持相应部分的功能，这样的优点是方便、速度快，机器一启动，你就可以使用这部分功能了；缺点是会使内核变得庞大起来，不管你是否需要这部分功能，它都会存在，这就是Windows惯用的招数，建议经常使用的部分直接编译到内核中，比如网卡。如果编译成模块，就会生成对应的.o文件，在使用的时候可以动态加载，优点是不会使内核过分庞大，缺点是你得自己来调用这些模块。

五、新版本内核的获取和更新

　　Linux内核版本发布的官方网站是[http://www.kernel.org](http://www.kernel.org/)。新版本的内核...问剑?恢质荈ull Source版本，另外一种是patch文件，即补丁。完整的内核版本比较大，一般是tar.gz或者是.bz2文件，二者分别是使用gzip或者bzip2进行压缩的文件，使用时需要解压缩。patch文件则比较小，一般只有几十K到几百K,但是patch文件是针对于特定的版本的，你需要找到自己对应的版本才能使用。   
　　编译内核需要root权限，以下操作都假定你是root用户。请把你需要升级的内核拷贝到/usr/src/下（下文中以2.4.18的内核的linux-2.4.18.tar.gz为例），命令为

#cp linux-2.4.18.tar.gz /usr/src

　　让我们先来查看一下当前/usr/src的内容，注意到有一个linux-2.4的符号链接，指向一个linux-2.4.7-10（以REDHAT7.2为例）的目录。这就是你所装linux的kernel源代码，删除这个链接。

　　现在解压我们下载的源程序文件。如果所下载的是.tar.gz（.tgz）文件，请使用下面的命令：

#tar -zxvf linux-2.4.18.tar.gz.tar.gz

　　如果你所下载的是.bz2文件，例如linux-2.4.0test8.tar.bz2，请使用下面的命令

#bzip2 -d linux-2.4.18.tar.bz2   
#tar -xvf linux-2.4.18.tar

　　文件将解压到/usr/src/linux目录中,我们把它稍作修改：

#mv linux linux-2.4.18   
#ln -s linux-2.4.18 linux

　　如果下载的是patch文件，就可以进行patch操作（下面假设patch-2.4.18已经位于/usr/src目录下了，否则你需要先把该文件拷贝到/usr/src下）：

#patch -p0 < patch-2.4.18

六、内核编译

　　通常要运行的第一个命令是：

＃cd /usr/src/linux   
＃make mrproper

　　该命令确保源代码目录下没有不正确的.o文件以及文件的互相依赖。由于我们使用刚下载的完整的源程序包进行编译，所以本步可以省略。而如果你多次使用了这些源程序编译内核，那么最好要先运行一下这个命令。   
　　确保/usr/include/目录下的asm、linux和scsi等链接是指向要升级的内核源代码的。它们分别链向源代码目录下的真正的、该计算机体系结构（对于PC机来说，使用的体系结构是i386）所需要的真正的include子目录。如：asm指向/usr/src/linux/include/asm-i386等。若没有这些链接，就需要手工创建，按照下面的步骤进行：

# cd /usr/include/   
# rm -r asm linux scsi   
# ln -s /usr/src/linux/include/asm-i386 asm   
# ln -s /usr/src/linux/include/linux linux   
# ln -s /usr/src/linux/include/scsi scsi

　　这是配置非常重要的一部分。删除掉/usr/include下的asm、linux和scsi链接后，再创建新的链接指向新内核源代码目录下的同名的目录。这些头文件目录包含着保证内核在系统上正确编译所需要的重要的头文件。现在你应该明白为什么我们上面又在/usr/src下"多余"地创建了个名为linux的链接了吧？

　　接下来的内核配置过程比较烦琐，但是配置的适当与否与日后Linux的运行直接相关，有必要了解一下一些主要的且经常用到的选项的设置。   
　　   
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，不推荐使用）   
#make menuconfig（基于文本选单的配置界面，字符终端下推荐使用）   
#make xconfig（基于图形窗口模式的配置界面，Xwindow下推荐使用）   
#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）

　　这三个命令中，make xconfig的界面最为友好，如果你可以使用Xwindow，那么就推荐你使用这个命令，界面如下：

　　如果你不能使用Xwindow，那么就使用make menuconfig好了。界面虽然比上面一个差点，总比make config的要好多了，下图为make menuconfig的界面：

　　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   
　　如果使用的是make xconfig，使用鼠标就可以选择对应的选项。如果使用的是make menuconfig，则需要使用空格键进行选取。你会发现在每一个选项前都有个括号, 但有的是中括号有的是尖括号，还有一种圆括号。 用空格键选择时可以发现，中括号里要么是空，要么是"\*"，而尖括号里可以是空，"\*"和"M"这表示前者对应的项要么不要，要么编译到内核里；后者则多一样选择，可以编译成模块。而圆括号的内容是要你在所提供的几个选项中选择一项。   
　　在编译内核的过程中，最烦杂的事情就是这步配置工作了，很多新手都不清楚到底该如何选取这些选项。实际上在配置时，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。选择的原则是将与内核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；不需要的功能就不要选；与内核关心紧密而且经常使用的部分功能代码直接编译到内核中。

至于选项，因为比较复杂，只是简单做一介绍，编译时应视具体情况，参考帮助的内容再加以选择。

1. Code maturity level options   
　　代码成熟等级。此处只有一项：prompt for development and/or incomplete code/drivers，如果你要试验现在仍处于实验阶段的功能，比如khttpd、IPv6等，就必须把该项选择为Y了；否则可以把它选择为N。

2. Loadable module support   
　　对模块的支持。这里面有三项：

　　Enable loadable module support：除非你准备把所有需要的内容都编译到内核里面，否则该项应该是必选的。   
　　Set version inFORMation on all module symbols：可以不选它。   
　　Kernel module loader：让内核在启动时有自己装入必需模块的能力，建议选上。

3. Processor type and features   
　　CPU类型。内容蛮多的，不一一介绍了，有关的几个如下：

　　Processor family：根据你自己的情况选择CPU类型。   
　　High Memory Support：大容量内存的支持。可以支持到4G、64G，一般可以不选。   
　　Math emulation：协处理器仿真。协处理器是在386时代的宠儿，现在早已不用了。   
　　MTTR support：MTTR支持。可不选。   
　　Symmetric multi-processing support：对称多处理支持。除非你富到有多个CPU，否则就不用选了。

4. General setup   
　　这里是对最普通的一些属性进行设置。这部分内容非常多，一般使用缺省设置就可以了。下面介绍一下经常使用的一些选项：

　　Networking support：网络支持。必须，没有网卡也建议你选上。   
　　PCI support：PCI支持。如果使用了PCI的卡，当然必选。   
　　PCI access mode：PCI存取模式。可供选择的有BIOS、Direct和Any，选Any吧。   
　　Support for hot-pluggabel devices：热插拔设备支持。支持的不是太好，可不选。   
　　PCMCIA/CardBus support：PCMCIA/CardBus支持。有PCMCIA就必选了。   
　　System V IPC   
　　BSD Process Accounting   
　　Sysctl support：以上三项是有关进程处理/IPC调用的，主要就是System V和BSD两种风格。如果你不是使用BSD，就按照缺省吧。   
　　Power Management support：电源管理支持。   
　　Advanced Power Management BIOS support：高级电源管理BIOS支持。

5. Memory Technology Device（MTD）   
　　MTD设备支持。可不选。

6. Parallel port support   
　　并口支持。如果不打算使用串口，就别选了。

7. Plug and Play configuration   
　　即插即用支持。虽然Linux对即插即用目前支持的不如Windows好，但是还是选上吧，这样你可以拔下鼠标之类的体验一下Linux下即插即用的感觉。

8. Block devices   
　　块设备支持。这个就得针对自己的情况来选了，简单说明一下吧：

　　Normal PC floppy disk support：普通PC软盘支持。这个应该必选。   
　　XT hard disk support：   
　　Compaq SMART2 support：   
　　Mulex DAC960/DAC1100 PCI RAID Controller support：RAID镜像用的。   
　　Loopback device support：   
　　Network block device support：网络块设备支持。如果想访问网上邻居的东西，就选上。   
　　Logical volume manager（LVM）support：逻辑卷管理支持。   
　　Multiple devices driver support：多设备驱动支持。   
　　RAM disk support：RAM盘支持。

9. Networking options   
　　网络选项。这里配置的是网络协议。内容太多了，不一一介绍了，自己看吧，如果你对网络协议有所了解的话，应该可以看懂的。如果懒得看，使用缺省选项（肯定要选中TCP/IP networking哦）就可以了。让我们看看，TCP/IP、ATM、IPX、DECnet、Appletalk……支持的协议好多哦，IPv6也支持了，Qos and/or fair queueing（服务质量公平调度）也支持了，还有kHTTPd，不过这些都还在实验阶段。

10. Telephony Support   
　　电话支持。Linux下可以支持电话卡，这样你就可以在IP上使用普通的电话提供语音服务了。记住，电话卡可和modem没有任何关系哦。

11. ATA/IDE/MFM/RLL support   
　　这个是有关各种接口的硬盘/光驱/磁带/软盘支持的，内容太多了，使用缺省的选项吧，如果你使用了比较特殊的设备，比如PCMCIA等，就到里面自己找相应的选项吧。

12. SCSI support   
　　SCSI设备的支持。我没有SCSI的设备，所以根本就不用选，如果你用了SCSI的硬盘/光驱/磁带等设备，自己找好了。

13. Fusion MPT device support   
　　需要Fusion MPT兼容PCI适配器，不用选。

14. I2O device support   
　　需要I2O接口适配器支持，在智能Input/Output（I2O）体系接口中使用。

15. Network device support   
　　网络设备支持。上面选好协议了，现在该选设备了，可想而知，内容肯定多得很。还好还好，里面大概分类了，有ARCnet设备、Ethernet（10 or 100 Mbit）、Ethernet（1000Mbit）、Wireless LAN（non-hamradio）、Token Ring device、Wan interfaces、PCMCIA network device support几大类。我用的是10/100M的以太网，看来只需要选则这个了。还是10/100M的以太网设备熟悉，内容虽然多，一眼就可以看到我所用的RealTeck RTL-8139 PCI Fast Ethernet Adapter support，为了免得麻烦，编译到内核里面好了，不选M了，选Y。耐心点，一般说来你都能找到自己用的网卡。如果没有，你只好自己到厂商那里去要驱动了。

16. Amateur Radio support   
　　配置业余无线广播。

17. IrDA（infrared）support   
　　红外线支持。

18. ISDN subsystem   
　　如果你使用ISDN上网，这个就必不可少了。

19. Old CD-ROM drivers（not SCSI、not IDE）   
　　做的可真周到，原来那些非SCSI/IDE口的光驱谁还在用啊，自己选吧，用IDE的CD-ROM不用选。

20. Character devices   
　　字符设备。这个内容又太多了，先使用缺省设置，需要的话自己就修改。把大类介绍一下吧：

　　I2C support：I2C是Philips极力推动的微控制应用中使用的低速串行总线协议。如果你要选择下面的Video For Linux，该项必选。   
　　Mice：鼠标。现在可以支持总线、串口、PS/2、C&T 82C710 mouse port、PC110 digitizer pad，自己根据需要选择。   
　　Joysticks：手柄。即使在Linux下把手柄驱动起来意义也不是太大，游戏太少了。   
　　Watchdog Cards：虽然称为Cards，这个可以用纯软件来实现，当然也有硬件的。如果你把这个选中，那么就会在你的/dev下创建一个名为watchdog的文件，它可以记录你的系统的运行情况，一直到系统重新启动的1分钟左右。有了这个文件，你就可以恢复系统到重启前的状态了。   
　　Video For Linux：支持有关的音频/视频卡。   
　　Ftape, the floppy tape device driver：   
　　PCMCIA character device support：

21. File systems   
　　文件系统。内容又太多了，老法子，在缺省选项的基础上进行修改。介绍以下几项：

　　Quota support：Quota可以限制每个用户可以使用的硬盘空间的上限，在多用户共同使用一台主机的情况中十分有效。   
　　DOS FAT fs support：DOS FAT文件格式的支持，可以支持FAT16、FAT32。   
　　ISO 9660 CD-ROM file system support：光盘使用的就是ISO 9660的文件格式。   
　　NTFS file system support：ntfs是NT使用的文件格式。   
　　/proc file system support：/proc文件系统是Linux提供给用户和系统进行交互的通道，建议选上，否则有些功能没法正确执行。

　　还有另外三个大类都规到这儿了：Network File Systems（网络文件系统）、Partition Types（分区类型）、Native Language Support（本地语言支持）。值得一提的是Network File Systems里面的两种：NFS和SMB分别是Linux和Windows相互以网络邻居的形式访问对方所使用的文件系统，根据需要加以选择。

22. Console drivers   
　　控制台驱动。一般使用VGA text console就可以了，标准的80\*25的文本控制台。

23. Sound   
　　声卡驱动。如果你能在列表中找到声卡驱动那自然最好，否则就试试OSS了。

24. USB supprot   
　　USB支持。很多USB设备，比如鼠标、调制解调器、打印机、扫描仪等，在Linux都可以得到支持，根据需要自行选择。

25. Kernel hacking   
　　配置了这个，即使在系统崩溃时，你也可以进行一定的工作了。普通用户是用不着这个功能的。

　　配置完后，存盘退出，当然你也可以把现在的配置文件保存起来，这样下次再配置的时候就省力气了。

内核编译的步骤：

先解压下载好的内核文件

   xz -d linux-2.6.32.71.tar.xz

     tar xvf  linux-2.6.32.71.tar

make menuconfig会生成 .config的配置文件

1、将/boot/config-2.6.18-53.e15 拷贝到Linux2.6.29内核目录下

2、make menuconfig 退出

3、make bzImage(20分钟）

4、make modules编译模块（编译选择为M的选项）

5、安装模块

make modules\_install

1. 制作文件系统

mkinitrd initrd-2.6.29 2.6.29

(/lib/modules/下必须有内核版本）会产生initrd-2.6.29

1. 内核安装过程

cp /arch/x86/boot/bzImage /boot/vmlinuz-2.6.29

cp /linux2.6.29/initrd-2.6.29 /boot/

cp是拷贝命令，你也可以手动操作拷贝

8、修改kconfig文件（控制grub的选项）

vim /etc/grub.conf

title my-linux

kernel /vmlinuz-2.6.29 ro \*\*\*

initrd /initrd-2.6.29

思考编译过程中每个步骤具体的含义！

重启 执行uname -a 查看系统版本