# 基于LFS的linux发行版制作

LFS是一部非常好的制作一个完整的操作系统的手册，但LFS是属于指导性的手册，因此它默认的前提条件是具备一定的Linux使用经验的用户群，所以它在每个软件包的安装部分只给出了在目录中的全部操作指令，而对于解压缩之类的则交给用户自己去解决，但对于很多第一次使用LFS的用户往往会出现一些不清楚某条命令应该是在哪里执行的问题，本文力图从实例上来解释这些问题。

本文虽然是采用VMWare Workstation 9的环境下制作的，但仍然符合使用真实机器上的过程，只是会在某些地方要根据具体机器进行更改，文中会在这些地方做出说明。

使用WMWare来写这篇文章是为了说明方便，因为VMWare在各种不同机器环境下模拟的虚拟设备几乎相同，所以用它来说明一些需要实际例子才说的清楚的地方非常合适，而且用VMWare来做即使出错也不会对真实的系统造成破坏，很适合新手使用。

本文力争完成一个完整的制作命令，可以根据本文以及LFS手册中提供的命令顺序输入就可以完成LFS了，通常命令表示为代码:

|  |
| --- |
| command |

恢复工作的方法更加适合在真实机器上制作LFS的同学，如果使用VMWare也可以直接使用VMWare的暂停功能或者快照功能来保存现场，继续的时候恢复现场就可以了。

# 一、准备工作：

下载LiveCD的ISO文件：http://ftp.osuosl.org/pub/lfs-livecd/lfslivecd-x86-6.3-r2145.

iso刻录ISO文件到光盘上，如果你是用真实机器当然少不了这步，不过如果你用VMWare的话，就可以直接使用ISO文件了。

以VMWare Workstation 9为例（真实机器可跳过此部分）

选择File->New->Virtual Machine...启动向导

选择Custom，并选择New-Workstation 9然后在选择Guest operating system里选择Linux，在Version里选择Other Linux 2.6.x kernel

存放目录、处理器数量和内存大小根据实际情况，建议内存不得小于128M，最好256M以上。

*相关知识点：*

*LFS-6.3采用了GCC4.1.2，如果使用128M编译GCC4.x.x话就需要使用swap了，但如果配置了256M就可以在没有swap的情况下完成编译，所以条件允许的情况下使用256M。*

*Network connection里选择Use network address translation(NAT)*

*SCSI Adpters选择默认的LSI Logic就可以了*

注意点：

这里选择的磁盘类型对于最后编译内核使用选项是有影响的，这里选择使用LSI Logic。

选择Create a new virtual disk

在Virtual Disk Type这步比较重要，你可以选择IDE也可以选择SCSI，但这里的选择直接影响到最后编译内核时的选项。这里以选择IDE为例子。

*相关知识点：*

*如果选择了IDE，则内核的默认设置就可以支持，但如果选择了SCSI，就必须在内核中加入对SCSI Adpters的支持，因为前面选择了LSI Logic，所以内核中就必须加入对LSI Logic的支持，否则将无法启动，相关部分在最后的内核编译部分有说明。*

*不过这里我建议选择使用IDE，一方面方便驱动，另外似乎LFS-6.3-r2145尚不能支持LSI Logic的SCSI磁盘，用该LiveCD启动后可能无法识别出硬盘来。*

对于磁盘大小，使用**4G**足够编译**LFS**了，但如果你打算编译更多的**BLFS**，这里可以考虑适当的增加一些大小，如果磁盘空间比较富裕就用16**G**好了。

完成向导后在虚拟机的界面里选择Edit virtual machine settings，将CD-ROM改为Use ISO image，然后选择LiveCD的ISO文件，如果你已经刻录好了光盘，将光盘放入光驱就行了。

点Start this virtual machine启动虚拟机

由于虚拟盘上没有任何信息，因此将自动从LiveCD中启动，在启动过程中会出现选择时区等信息，你可以按照实际情况选择，也可以按照默认选择，简单点就是等待一会系统会自动进行选择。这里选择时区为Asia/Shanghai，选择本地语言为English (USA, UTF-8)，其它的都按默认选择了。

在制作过程中，按照手册敲命令是一个很繁琐的事情，你可以在使用LiveCD启动系统后，对系统进行如下设置：

a. 设置客户端操作系统中root的密码

b. 修改/etc/ssh/sshd\_config文件， 去掉Port 22前的注释 ＃

c. /etc/rc.d/init.d/sshd start  重新启动ssh服务

然后就可以通过putty登录系统。登录系统后，你可以直接从手册中复制命令，粘贴在putty中。

*相关知识点*

*使用ifconfig命令可以查看你的机器的ip地址。*

启动完成LiveCD后就开始建造自己的LFS的历程了。

***LiveCD下的重要目录***

*/usr/share/LFS-BOOK-6.3-HTML目录存放的就是LFS手册了*

*/lfs-sources里面存放的就是建造LFS所需要的源码包，不需要到处下载软件了。*

# 二、准备新的分区

**1。创建磁盘分区：**

用cfdisk命令在磁盘上创建两个分区，/dev/hda1和/dev/hda2。/dev/hda1大小为1G，用作交换分区。磁盘的剩余空间留给/dev/hda2作为文件系统。

2。在新的分区上创建文件系统，代码:

|  |
| --- |
| mkswap /dev/hda1  **mke2fs -jv /dev/*hda2*** |

*相关知识点：*

*磁盘格式化一定要在磁盘分区未进行加载前进行。*

***mkswap****是用于将磁盘分区格式化为交换分区的命令。*

*这里我将****/dev/hda2****格式化了为****ext3****格式，如果你喜欢其它格式的文件系统，你可以使用相应的命令来格式化。*

注意：这里要根据实际情况建立和设置分区，如果你不太清楚这个问题，请先不要开始，否则可能造成难以恢复的损失！（这里只是根据**VMWare**里面的情况做的例子，在**VMWare**中相对安全些，建议初学者在虚拟机中开始。）

如果你的内存不太大，想在编译期间就使用上交换分区的话，可使用下面的命令激活交换分区。

|  |
| --- |
| swapon /dev/hda1 |

*相关知识点：*

*swapon用于激活交换分区*

*swapoff用于将激活的交换分区停用*

*可以通过free命令来查看当前的内存使用情况*

3。安装新的分区

创建文件系统挂接点目录，代码:

|  |
| --- |
| export LFS=/mnt/lfs  mkdir -pv $LFS |

*相关知识点：*

***export LFS=/mnt/lfs****这条命令的作用是为了后面引用****“****创作基地****”****的绝对路径方便而设置****LFS****这样的环境变量。*

加载/dev/hda2到“创作基地”，代码:

|  |
| --- |
| mount /dev/hda2 $LFS |

创建必要的目录并设置属性，创建源代码编译用目录，代码:

|  |
| --- |
| mkdir -v $LFS/sources  chmod -v a+wt $LFS/sources  cp /lfs-sources/\* $LFS/sources/ |

*相关知识点：*

*chmod a+wt是将目录或文件的属性设置为1777，这样任何人都可以对其进行读写。*

创建工具链目录，代码为:

|  |
| --- |
| mkdir -v $LFS/tools  ln -sv $LFS/tools / |

注意：

**ln -sv $LFS/tools**执行后应该会输出

**`/tools' -> `/mnt/lfs/tools'**

表示正确。

*相关知识点：*

*上面这两句就建立了神奇的工具链目录（是工具链目录不是工具链），这样的创建方式是为了在创建工具链和使用工具链创建目标系统的时候对于工具链的位置都是****/tools****，这样可保证工具链的正常使用*

创建**lfs**用户，代码:

|  |
| --- |
| groupadd lfs  useradd -s /bin/bash -g lfs -m -k /dev/null lfs |

设置lfs密码，代码:

|  |
| --- |
| passwd lfs |

将tools和sources目录的用户改为lfs，以便后面使用lfs来操作这两个目录，代码:

|  |
| --- |
| chown -v lfs $LFS/tools  chown -v lfs $LFS/sources |

登陆到lfs用户，代码:

|  |
| --- |
| su - lfs |

这时候你会发现命令行提示符已经由#改为了$

*相关知识点：*

*其实如果不使用lfs用root也是能完成工具链的，不过需要对root的环境变量进行修改，还要防止因为输入错误而导致覆盖主系统下的文件，所以LFS手册中制作工具链部分就是为了解决这种意外的发生而用lfs用户来建立工具链建立****lfs****用户的环境*

改变lfs用户的工作环境，代码:

|  |
| --- |
| cat > ~/.bash\_profile << "EOF"  exec env -i HOME=$HOME TERM=$TERM PS1='\u:\w\$ ' /bin/bash  EOF  cat > ~/.bashrc << "EOF"  set +h  umask 022  LFS=/mnt/lfs  LC\_ALL=POSIX  PATH=/tools/bin:/bin:/usr/bin  export LFS LC\_ALL PATH  EOF  source ~/.bash\_profile |

这里利用了bash的环境变量的设置文件将lfs的环境设置为符合编译工具链要求的最少的环境参数这里面最重要的就是**PATH**这个参数，目的是为了能够利用工具链里面的工具制作工具链：首先查找**/tools/bin**下是否有需要的命令，如果没有再到**/bin**和**/usr/bin**下找，然后用**/bin**或**/usr/bin**下面的命令来帮助生成需要的命令并放在**/tools/bin**下，这样此消彼涨，最终可完成一个自给自足的工具链。

到此为止就可以开始工具链的制作了，不过制作**LFS**是一个漫长而浩大的工程，所以要一直开机直到完成有时候比较困难，特别是在机器速度比较慢的情况下，能够重新启动到最后工作的状态是很重要的。在不同的阶段重新启动并恢复状态的步骤不完全相同，所以本文会在不同的阶段讨论重新启动恢复到工作状态的方法和步骤。

从现在开始一直到临时系统制作结束，也就是完成**Stripping**中间的步骤中如果重新启动的恢复步骤：

1.重新启动计算机，并从LiveCD启动

相关知识点：在VMWare中因为磁盘已经有了信息了，所以会从磁盘启动，需要在启动虚拟机中的机器时按F2进入虚拟机的虚拟BIOS，然后在BOOT中设置第一启动为CD-ROM，保存退出即可。

2.LiveCD启动过程同第一次启动选择一样。

3.加载分区

export LFS=/mnt/lfs

mkdir -pv $LFS

mount /dev/hda2 $LFS

4.加载交换分区（如果不想用交换分区或者没有交换分区可跳过此步骤）

swapon /dev/hda1

5.建立工具链的链接

ln -sv $LFS/tools /

6.创建lfs用户

groupadd lfs

useradd -s /bin/bash -g lfs -m -k /dev/null lfs

passwd lfs

chown -v lfs $LFS/tools

chown -v lfs $LFS/sources

su - lfs

7.建立lfs用户的环境

cat > ~/.bash\_profile << "EOF"

exec env -i HOME=$HOME TERM=$TERM PS1='\u:\w\$ ' /bin/bash

EOF

cat > ~/.bashrc << "EOF"

set +h

umask 022

LFS=/mnt/lfs

LC\_ALL=POSIX

PATH=/tools/bin:/bin:/usr/bin

export LFS LC\_ALL PATH

EOF

source ~/.bash\_profile

8.检查一下

export命令查看输出，应该是

declare -x HOME="/home/lfs"

declare -x LC\_ALL="POSIX"

declare -x LFS="/mnt/lfs"

declare -x OLDPWD

declare -x PATH="/tools/bin:/bin:/usr/bin"

declare -x PS1="file:///u://w///$ "

declare -x PWD="/home/lfs"

declare -x SHLVL="1"

declare -x TERM="linux"

9.进入编译目录

cd $LFS/sources

基本上就恢复工作状态了。

# 三、开始工具链的制作

工具链需要安装约26个软件包，由于存在依赖关系，这些软件包应当按照手册中的顺序依次装入。大多数软件包的安装过程基本类似，过程如下：

进入**LFS**包编译目录，代码:

|  |
| --- |
| cd $LFS/sources |

对于每一个软件包:

tar xvf foo.ver.tar.xx

cd foo.ver

如果要补丁,补丁的打法:

patch -Np1 -i /yourPatchName.patch

编译

每次完成编译安装后执行以下动作：

cd $LFS/sources

rm -rf foo.ver

*相关知识点：*

*大家可以注意到后面所有的解包命令均使用tar xvf来完成，而不管文件的压缩方式是bz2还是gz，这是因为较新的tar程序都具有自动识别后缀名并自动调用相应的解压缩工具的能力，所以可以不需要指定压缩方式，但对于早期的tar命令则可能不具备这个功能因此需要你根据包的压缩方式来指定，如bz2使用j，gz使用z，对应上面的binutils则是tar xvjf /lfs-sources/binutils-2.17.tar.bz2*

注意，在手册5.6节glibc安装完成后，要进行工具链的调整，调整方法按照手册5.7节所描述的步骤进行即可。

Stripping，这步是可有可无的，如果你打算今后还要用/tools里面的东西，那么可以strip一下来减少占用的磁盘空间，但如果做完目标系统后就删除了，不Strip也可以，反正最后也是要删掉的。

代码:

|  |
| --- |
| strip --strip-debug /tools/lib/\*  strip --strip-unneeded /tools/{,s}bin/\* |

info和man里面的内容在制作过程中没什么用处，所以删掉也没啥关系。代码:

|  |
| --- |
| rm -rf /tools/{info,man} |

退出**lfs**用户（这步不要少了），代码:

|  |
| --- |
| exit |

到目前为止，临时系统已经制作完成了，接着就要开始制作真正的目标系统了，如果你到目前为止没出什么问题，那么恭喜你成功的通过了一关，不过接着还有相当长的路。

现在你应该是处于**root**用户状态的，看看你的命令行提示符是不是回到了**#**。

从现在开始不在需要lfs用户来制作系统了，因此我们用代码:

|  |
| --- |
| chown -R root:root $LFS/tools |

重新设置目录权限，便于后面的工作。

# 四、安装正式系统软件包

创建三个重要目录。代码:

|  |
| --- |
| mkdir -pv $LFS/{dev,proc,sys} |

创建两个目标系统所必须的设备文件，代码:

|  |
| --- |
| mknod -m 600 $LFS/dev/console c 5 1  mknod -m 666 $LFS/dev/null c 1 3 |

利用主系统加载几个重要的文件系统，请注意这个步骤对于后面的工作极其重要。代码:

|  |
| --- |
| mount -v --bind /dev $LFS/dev  mount -vt devpts devpts $LFS/dev/pts  mount -vt tmpfs shm $LFS/dev/shm  mount -vt proc proc $LFS/proc  mount -vt sysfs sysfs $LFS/sys |

*相关知识点：*

*mount命令加载的分区在重新启动后就失效了，所以在这其中重新启动则需要重新加载。*

Chroot到目标系统的目录下，以便不受主系统的影响来制作目标系统。代码:

|  |
| --- |
| chroot "$LFS" /tools/bin/env -i \  HOME=/root TERM="$TERM" PS1='\u:\w\$ ' \  PATH=/bin:/usr/bin:/sbin:/usr/sbin:/tools/bin \  /tools/bin/bash --login +h |

注意：这个时候你的提示符会是**“I have no name!”**，没有关系继续我们的工作很快就可以正常了。

这个时候如果你关机或重新启动，那么重新开机后回到工作状态的步骤是：

1.重新启动计算机，并从LiveCD启动

2.加载分区

export LFS=/mnt/lfs

mkdir -pv $LFS

mount /dev/hda2 $LFS

3.加载交换分区（如果不想用交换分区或者没有交换分区可跳过此步骤）

swapon /dev/hda1

4.加载必要的文件系统

mount -v --bind /dev $LFS/dev

mount -vt devpts devpts $LFS/dev/pts

mount -vt tmpfs shm $LFS/dev/shm

mount -vt proc proc $LFS/proc

mount -vt sysfs sysfs $LFS/sys

5.Chroot到目标系统下

chroot "$LFS" /tools/bin/env -i \

HOME=/root TERM="$TERM" PS1='\u:\w\$ ' \

PATH=/bin:/usr/bin:/sbin:/usr/sbin:/tools/bin \

/tools/bin/bash --login +h

建立目标系统的目录结构。代码:

|  |
| --- |
| mkdir -pv /{bin,boot,etc/opt,home,lib,mnt,opt}  mkdir -pv /{media/{floppy,cdrom},sbin,srv,var}  install -dv -m 0750 /root  install -dv -m 1777 /tmp /var/tmp  mkdir -pv /usr/{,local/}{bin,include,lib,sbin,src}  mkdir -pv /usr/{,local/}share/{doc,info,locale,man}  mkdir -pv /usr/{,local/}share/{misc,terminfo,zoneinfo}  mkdir -pv /usr/{,local/}share/man/man{1..8}  for dir in /usr /usr/local; do  ln -sv share/{man,doc,info} $dir  done  mkdir -pv /var/{lock,log,mail,run,spool}  mkdir -pv /var/{opt,cache,lib/{misc,locate},local} |

创建几个必要的链接，因为在目标系统的编译过程中，部分编译程序会用绝对路径来寻找命令或文件。代码:

|  |
| --- |
| ln -sv /tools/bin/{bash,cat,echo,grep,pwd,stty} /bin  ln -sv /tools/bin/perl /usr/bin  ln -sv /tools/lib/libgcc\_s.so{,.1} /usr/lib  ln -sv /tools/lib/libstdc++.so{,.6} /usr/lib  ln -sv bash /bin/sh  touch /etc/mtab |

创建**root**及**nobody**用户和必要的组。代码:

|  |
| --- |
| cat > /etc/passwd << "EOF"  root:x:0:0:root:/root:/bin/bash  nobody:x:99:99:Unprivileged User:/dev/null:/bin/false  EOF  cat > /etc/group << "EOF"  root:x:0:  bin:x:1:  sys:x:2:  kmem:x:3:  tty:x:4:  tape:x:5:  daemon:x:6:  floppy:x:7:  disk:x:8:  lp:x:9:  dialout:x:10:  audio:x:11:  video:x:12:  utmp:x:13:  usb:x:14:  cdrom:x:15:  mail:x:34:  nogroup:x:99:  EOF |

重新加载bash，以使root用户起效，这样前面的提示符就不会是“I have no name!”。代码:

|  |
| --- |
| exec /tools/bin/bash --login +h |

创建和设置几个临时文件和日志文件。代码:

|  |
| --- |
| touch /var/run/utmp /var/log/{btmp,lastlog,wtmp}  chgrp -v utmp /var/run/utmp /var/log/lastlog  chmod -v 664 /var/run/utmp /var/log/lastlog |

到目前为止，创建目标系统的准备工作以基本完成，下面就要开始目标系统的软件包安装了。目标系统需要安装约40余个软件包，部分软件包需要安装2次，有些软件包安装后还需要进行配置，如果软件包提供了测试，则最好测试一下安装是否正确。每一个软件包的安装过程大致如下：

首先进入到源码目录下。代码:

|  |
| --- |
| cd /sources |

之后的步骤基本上就是解压软件包、打补丁、编译、测试、安装、配置。按照手册第6章的步骤进行即可。

注意：在Glibc（6.9）安装之后，要重新调整工具链，并测试工具链是否正确。按照6.10节做即可。

从现在开始一直到制作结束，重新开机后回到工作状态的步骤是：

1.重新启动计算机，并从LiveCD启动

2.加载分区

export LFS=/mnt/lfs

mkdir -pv $LFS

mount /dev/hda2 $LFS

3.加载交换分区（如果不想用交换分区或者没有交换分区可跳过此步骤）

swapon /dev/hda1

4.加载必要的文件系统

mount -v --bind /dev $LFS/dev

mount -vt devpts devpts $LFS/dev/pts

mount -vt tmpfs shm $LFS/dev/shm

mount -vt proc proc $LFS/proc

mount -vt sysfs sysfs $LFS/sys

5.Chroot到目标系统下

chroot "$LFS" /usr/bin/env -i \

HOME=/root TERM="$TERM" PS1='\u:\w\$ ' \

PATH=/bin:/usr/bin:/sbin:/usr/sbin \

/bin/bash --login

6.进入编译目录

cd /sources

export LFS=/sources

# 五、设置系统的启动脚本

**1。安装LFS-Bootscripts-6.3**

**与前面安装软件包的方法一样，按照手册7.2节，安装lfs-bootscripts-6.3**

代码:

|  |
| --- |
| tar xvf $LFS/lfs-bootscripts-6.3.tar.bz2  cd lfs-bootscripts-6.3  make install  cd ..  rm -rf lfs-bootscripts-6.3 |

2。时间设置（**Configuring the setclock Script**）

代码:

|  |
| --- |
| cat > /etc/sysconfig/clock << "EOF"  # Begin /etc/sysconfig/clock  UTC=1  # End /etc/sysconfig/clock  EOF |

3。设置**bash**下的键盘功能键设置

代码:

|  |
| --- |
| cat > /etc/inputrc << "EOF"  set horizontal-scroll-mode Off  set meta-flag On  set input-meta On  set convert-meta Off  set output-meta On  set bell-style none  "\eOd": backward-word  "\eOc": forward-word  "\e[1~": beginning-of-line  "\e[4~": end-of-line  "\e[5~": beginning-of-history  "\e[6~": end-of-history  "\e[3~": delete-char  "\e[2~": quoted-insert  "\eOH": beginning-of-line  "\eOF": end-of-line  "\e[H": beginning-of-line  "\e[F": end-of-line  EOF |

4。设置**Bash Shell**启动文件（**The Bash Shell Startup Files**）

代码:

|  |
| --- |
| cat > /etc/profile << "EOF"  # Begin /etc/profile  export LANG=zh\_CN.UTF-8  export INPUTRC=/etc/inputrc  alias ls="ls --color"  export PS1='\u:\w\$ '  # End /etc/profile  EOF |

5。设置本地网络名

代码:

|  |
| --- |
| echo "HOSTNAME=mylinux" > /etc/sysconfig/network |

6。设置**hosts**文件

代码:

|  |
| --- |
| cat > /etc/hosts << "EOF"  # Begin /etc/hosts (no network card version)  127.0.0.1 mylinux localhost  # End /etc/hosts (no network card version)  EOF |

7。设置网络的静态地址

代码:

|  |
| --- |
| cd /etc/sysconfig/network-devices  mkdir -v ifconfig.eth0  cat > ifconfig.eth0/ipv4 << "EOF"  ONBOOT=yes  SERVICE=ipv4-static  IP=192.168.1.1  GATEWAY=192.168.1.2  PREFIX=24  BROADCAST=192.168.1.255  EOF |

注意：**IP**、**GATEWAY**、**BROADCAST**的地址根据自己的实际情况设置。

8。设置**DNS,**代码:

|  |
| --- |
| cat > /etc/resolv.conf << "EOF"  # Begin /etc/resolv.conf  nameserver 你的首个DNS的地址  nameserver 你的第二DNS的地址  # End /etc/resolv.conf  EOF |

# 六、让你的LFS启动起来

建立**fstab**文件,代码:

|  |
| --- |
| cat > /etc/fstab << "EOF"  # Begin /etc/fstab  # file system mount-point type options dump fsck  # order  /dev/hda2 / xfs defaults 1 1  /dev/hda1 swap swap pri=1 0 0  proc /proc proc defaults 0 0  sysfs /sys sysfs defaults 0 0  devpts /dev/pts devpts gid=4,mode=620 0 0  shm /dev/shm tmpfs defaults 0 0  # End /etc/fstab  EOF |

注意：这里的磁盘名以及文件系统名需要根据实际情况修改。

编译并安装内核Linux-2.6.22.5,代码

代码:

|  |
| --- |
| cd /sources  tar xvf $LFS/linux-2.6.22.5.tar.bz2  cd linux-2.6.22.5  make mrproper  make menuconfig |

根据你的机器实际情况配置内核选项，这里为了说明方便，以VMWare5.5为基础虚拟的硬件来配置内核如果在建立虚拟机的时候是选择的BusLogic的SCSI磁盘，那么应该在Device Drivers->SCSI device support->SCSI low-level drivers下加入BusLogic SCSI support的支持，可以采用编译到内核来避免未用initrd脚本来加载模块而导致启动失败

如果想支持网络则应该在Device Drivers->Networking support->Ethernet (10 or 100Mbit)加入AMD PCnet32 PCI support的支持，可以采用编译到内核也可以编译成模块的方式

如果想支持声卡则应该在Device Drivers->Sound->Advanced Linux Sound Architecture->PCI devices加入(Creative) Ensoniq AudioPCI 1371/1373的支持，编译成模块即可。

文件系统的支持，需要根据之前将目标系统分区格式化的情况而定，使用了什么文件系统就需要加入该文件系统的支持。

接下来就是编译和安装内核，代码:

|  |
| --- |
| make  make modules\_install  cp -v arch/i386/boot/bzImage /boot/lfskernel-2.6.22.5  cp -v System.map /boot/System.map-2.6.22.5  cp -v .config /boot/config-2.6.22.5  install -d /usr/share/doc/linux-2.6.22.5  cp -r Documentation/\* /usr/share/doc/linux-2.6.22.5 |

配置Grub，使系统能启动，这里设置需要根据情况而修改，这里以之前介绍的分区设置为例：代码为:

|  |
| --- |
| grub |

在grub中输入

|  |
| --- |
| root (hd0,1)  setup (hd0)  quit |

设置**grub**启动菜单，代码:

|  |
| --- |
| cat > /boot/grub/menu.lst << "EOF"  # Begin /boot/grub/menu.lst  # By default boot the first menu entry.  default 0  # Allow 30 seconds before booting the default.  timeout 30  # Use prettier colors.  color green/black light-green/black  # The first entry is for LFS.  title LFS 6.3  root (hd0,1)  kernel /boot/lfskernel-2.6.22.5 root=/dev/hda2 vga=788  EOF |

注意：这里**root**后面的磁盘分区需要根据实际情况调整。

将**menu.lst**连接到**/etc**目录下，代码为:

|  |
| --- |
| mkdir -v /etc/grub  ln -sv /boot/grub/menu.lst /etc/grub |

重新启动你的机器，你将会看到你已成功地构建了一个你自己的linux系统。

这样的系统仅仅是一个最基础的linux系统，接下来你可以在此基础上参考BLFS进一步安装其他软件包。