



马哥教育  
最专业的Linux培训机构

# 系统启动和内核管理

- ❖ CentOS 5和6的启动流程
- ❖ 服务管理
- ❖ Grub管理
- ❖ 自制Linux
- ❖ 启动排错
- ❖ 编译安装内核

马哥教育

www.magedu.com

## ❖ Linux: kernel+rootfs

**kernel:** 进程管理、内存管理、网络管理、驱动程序、文件系统、安全功能

**rootfs:** 程序和glibc

库: 函数集合, **function**, 调用接口 (头文件负责描述)

过程调用: **procedure**, 无返回值

函数调用: **function**

程序: 二进制执行文件

## ❖ 内核设计流派:

单内核(**monolithic kernel**): **Linux**

把所有功能集成于同一个程序

微内核(**micro kernel**): **Windows, Solaris**

每种功能使用一个单独子系统实现

## ❖ Linux内核特点:

支持模块化: **.ko** (内核对象)

如: 文件系统, 硬件驱动, 网络协议等

支持内核模块的动态装载和卸载

## ❖ 组成部分:

核心文件: **/boot/vmlinuz-VERSION-release**

**ramdisk**: 辅助的伪根系统

**CentOS 5**: **/boot/initrd-VERSION-release.img**

**CentOS 6, 7**: **/boot/initramfs-VERSION-release.img**

模块文件: **/lib/modules/VERSION-release**



- 1.加载**BIOS**的硬件信息，获取第一个启动设备。
- 2.读取第一个启动设备**MBR**的引导加载程序(**grub**)的启动信息
- 3.加载核心操作系统的核心信息，核心开始解压缩，并尝试驱动所有的硬件设备。
- 4.核型执行**init**程序并获取运行信息。
- 5.**Init**执行/**etc/rc.d/rc.sysinit**文件。
- 6.启动核心的外挂模块(/**etc/modprobe.conf**)。
- 7.**Init**执行运行的各个批处理文件(**scripts**)。
- 8.**Init**执行/**etc/rc.d/rc.local**。
- 9.执行/**bin/login**程序，等待用户登录。
- 10.登录之后开始以**Shell**控制主机。



# 启动流程

- ❖ **POST: Power-On-Self-Test**, 加电自检, 是**BIOS**功能的一个主要部分。负责完成对**CPU**、主板、内存、硬盘子系统、显示子系统、串并行接口、键盘、**CD-ROM**光驱等硬件情况的检测。

**ROM: BIOS, Basic Input and Output System**, 保存着有关计算机系统最重要的基本输入输出程序, 系统信息设置、开机加电自检程序和系统启动自举程序等。

**RAM: CMOS**互补金属氧化物半导体, 保存各项参数的设定

按次序查找引导设备, 第一个有引导程序的设备为本次启动设备

- ❖ **bootloader**: 引导加载器, 引导程序

**windows: ntloader**, 仅是启动**OS**

**Linux**: 功能丰富, 提供菜单, 允许用户选择要启动系统或不同的内核版本; 把用户选定的内核装载到内存中的特定空间中, 解压、展开, 并把系统控制权移交给内核

**LILO: LInux LOader**

**GRUB: GRand Unified Bootloader**

**GRUB 0.X: GRUB Legacy, GRUB2**

## ❖ MBR:

446: bootloader, 64: 分区表, 2: 55AA

## ❖ GRUB:

primary boot loader : 1st stage, 1.5 stage

secondary boot loader : 2nd stage, 分区文件

## ❖ kernel:

自身初始化:

探测可识别到的所有硬件设备

加载硬件驱动程序（可能借助于ramdisk加载驱动）

以只读方式挂载根文件系统

运行用户空间的第一个应用程序: /sbin/init



# 系统启动流程

❖ **init**程序的类型:

❖ **SysV: init, CentOS 5之前**

配置文件: `/etc/inittab`

❖ **Upstart: init, CentOS 6**

配置文件: `/etc/inittab, /etc/init/*.conf`

❖ **Systemd: systemd, CentOS 7**

配置文件: `/usr/lib/systemd/system`  
`/etc/systemd/system`

www.magedu.com

# 启动流程

## ❖ ramdisk:

- ❖ 内核中的特性之一：使用缓冲和缓存来加速对磁盘上的文件访问

ramdisk --> ramfs 提高速度

CentOS 5: initrd, 工具程序: mkinitrd

CentOS 6: initramfs, 工具程序: mkinitrd, dracut

## ❖ 系统初始化:

POST --> BootSequence (BIOS) -->

Bootloader(MBR) --> kernel(ramdisk) --> rootfs(只读) --> init (systemd)

# 启动流程

❖ `/sbin/init` CentOS6之前

❖ 运行级别：为系统运行或维护等目的而设定；0-6：7个级别

0：关机

1：单用户模式(**root**自动登录), **single**, 维护模式

2：多用户模式，启动网络功能，但不会启动**NFS**；维护模式

3：多用户模式，正常模式；文本界面

4：预留级别；可同3级别

5：多用户模式，正常模式；图形界面

6：重启

❖ 默认级别：3, 5

❖ 切换级别：`init #`

❖ 查看级别：`runlevel ; who -r`

❖ **init**读取其初始化文件: **/etc/inittab**

初始运行级别(**RUN LEVEL**)

系统初始化脚本

对应运行级别的脚本目录

捕获某个关键字顺序

定义**UPS**电源终端/恢复脚本

在虚拟控制台生成**getty**

在运行级别**5**初始化

马哥教育  
www.magedu.com

- ❖ 配置文件: `/etc/inittab`
- ❖ 每一行定义一种action以及与之对应的process

`id:runlevel:action:process`

`action:`

`wait`: 切换至此级别运行一次

`respawn`: 此process终止, 就重新启动之

`initdefault`: 设定默认运行级别; process省略

`sysinit`: 设定系统初始化方式, 此处一般为指定

`/etc/rc.d/rc.sysinit`

`ca::ctrlaltdel:/sbin/shutdown -t3 -r now`

`id:3:initdefault:`

`si::sysinit:/etc/rc.d/rc.sysinit`

`l0:0:wait:/etc/rc.d/rc 0`

`l1:1:wait:/etc/rc.d/rc 1...`

`l6:6:wait:/etc/rc.d/rc 6`

## ❖ /etc/inittab

设置系统默认的运行级别

id:3:initdefault:

## ❖ /etc/init/control-alt-delete.conf

## ❖ /etc/init/tty.conf

## ❖ /etc/init/start-ttys.conf

## ❖ /etc/init/rc.conf

## ❖ /etc/init/prefdm.conf

马哥教育

www.magedu.com



## ❖ `/etc/rc.d/rc.sysinit`: 系统初始化脚本

- (1) 设置主机名
- (2) 设置欢迎信息
- (3) 激活udev和selinux
- (4) 挂载`/etc/fstab`文件中定义的文件系统
- (5) 检测根文件系统，并以读写方式重新挂载根文件系统
- (6) 设置系统时钟
- (7) 激活swap设备
- (8) 根据`/etc/sysctl.conf`文件设置内核参数
- (9) 激活lvm及software raid设备
- (10) 加载额外设备的驱动程序
- (11) 清理操作

❖ 说明: `rc N -->` 意味着读取 `/etc/rc.d/rcN.d/`

**K\*: K##\*:** ##运行次序; 数字越小, 越先运行; 数字越小的服务, 通常为依赖到别的服务

**S\*: S##\*:** ##运行次序; 数字越小, 越先运行; 数字越小的服务, 通常为被依赖到的服务

```
for srv in /etc/rc.d/rcN.d/K*; do  
    $srv stop
```

```
done
```

```
for srv in /etc/rc.d/rcN.d/S*; do  
    $srv start
```

```
done
```

## ❖ chkconfig命令

- ❖ 查看服务在所有级别的启动或关闭设定情形：

```
chkconfig [--list] [name]
```

- ❖ 添加：

SysV的服务脚本放置于/etc/rc.d/init.d (/etc/init.d)

```
chkconfig --add name
```

```
#!/bin/bash
```

```
#LLLL 表示初始在哪个级别下启动，-表示都不启动
```

```
# chkconfig: LLLL nn nn
```

- ❖ 删除：

```
chkconfig --del name
```

- ❖ 修改指定的链接类型

```
chkconfig [--level levels] name <on|off|reset>
```

--level LLLL: 指定要设置的级别；省略时表示2345

## ❖ ntsysv命令

❖ **service** 命令：手动管理服务

**service** 服务 **start|stop|restart**

**service --status-all**

❖ 瞬态（**Transient**）服务被**xinetd**进程所管理

进入的请求首先被**xinetd**代理

配置文件：**/etc/xinetd.conf**、**/etc/xinetd.d/<service>**

与**libwrap.so**文件链接

用**chkconfig**控制的服务：

**chkconfig tftp on**

马哥教育  
www.magedu.com

- ❖ 注意：正常级别下，最后启动一个服务S99local没有链接至 `/etc/rc.d/init.d` 一个服务脚本，而是指向了 `/etc/rc.d/rc.local` 脚本
- ❖ 不便或不需写为服务脚本放置于 `/etc/rc.d/init.d/` 目录，且又想开机时自动运行的命令，可直接放置于 `/etc/rc.d/rc.local` 文件中
- `/etc/rc.d/rc.local` 在指定运行级别脚本后运行
- 可以根据情况，进行自定义修改

马哥教育

www.magedu.com

- ❖ 1:2345:respawn:/usr/sbin/mingetty tty1
- ❖ 2:2345:respawn:/usr/sbin/mingetty tty2
- ❖ ...
- ❖ 6:2345:respawn:/usr/sbin/mingetty tty6  
mingetty会自动调用login程序
- ❖ x:5:respawn:/etc/X11/prefdm -nodaemon

马哥教育

www.magedu.com



- ❖ 总结: `/sbin/init --> (/etc/inittab) --> 设置默认运行级别 --> 运行系统初始脚本、完成系统初始化 --> (关闭对应下需要关闭的服务)启动需要启动服务 --> 设置登录终端`
- ❖ CentOS 6 init程序为: `upstart`, 其配置文件:  
`/etc/inittab, /etc/init/*.conf`, 配置文件的语法 遵循 `upstart` 配置文件语法格式, 和CentOS5不同

马哥教育

www.magedu.com

## ❖ CentOS 6启动流程:

POST --> Boot Sequence(BIOS) --> Boot Loader --> Kernel(ramdisk) --> rootfs --> switchroot --> /sbin/init --> (/etc/inittab, /etc/init/\*.conf) --> 设定默认运行级别 --> 系统初始化脚本rc.sysinit --> 关闭或启动对应级别的服务 --> 启动终端

## ❖ grub: GRand Unified Bootloader

grub 0.x: grub legacy

grub 1.x: grub2

grub legacy:

stage1: mbr

stage1\_5: mbr之后的扇区, 让stage1中的bootloader能识别stage2所在的分区上的文件系统

stage2: 磁盘分区(/boot/grub/)

- ❖ 配置文件: `/boot/grub/grub.conf` <-- `/etc/grub.conf`
- ❖ **stage2**及内核等通常放置于一个基本磁盘分区
- ❖ 功用:
- ❖ (1) 提供启动菜单、并提供交互式接口
  - a: 内核参数
  - e: 编辑模式, 用于编辑菜单
  - c: 命令模式, 交互式接口
- ❖ (2) 加载用户选择的内核或操作系统
  - 允许传递参数给内核
  - 可隐藏启动菜单
- ❖ (3) 为菜单提供了保护机制
  - 为编辑启动菜单进行认证
  - 为启用内核或操作系统进行认证

## ❖ grub的命令行接口

**help:** 获取帮助列表

**help KEYWORD:** 详细帮助信息

**find (hd#,#)/PATH/TO/SOMEFILE:**

**root (hd#,#)**

**kernel /PATH/TO/KERNEL\_FILE:** 设定本次启动时用到的内核文件；额外还可添加许多内核支持使用的**cmdline**参数

例如: **max\_loop=100 selinux=0 init=/path/to/init**

**initrd /PATH/TO/INITRAMFS\_FILE:** 设定为选定的内核提供额外文件的**ramdisk**;

**boot:** 引导启动选定的内核

## ❖ 识别硬盘设备:

(hd#, #)

hd#: 磁盘编号, 用数字表示; 从0开始编号

#: 分区编号, 用数字表示; 从0开始编号

(hd0,0) 第一块硬盘, 第一个分区

## ❖ 手动在grub命令行接口启动系统:

```
grub> root (hd#, #)
```

```
grub> kernel /vmlinuz-VERSION-RELEASE ro  
root=/dev/DEVICE
```

```
grub> initrd /initramfs-VERSION-RELEASE.img
```

```
grub> boot
```

❖ 配置文件: `/boot/grub/grub.conf`

**default=#**: 设定默认启动的菜单项; 落单项(**title**)编号从0开始

**timeout=#**: 指定菜单项等待选项选择的时长

**splashimage=(hd#,#)/PATH/TO/XPM\_FILE**: 菜单背景图片文件路径

**hiddenmenu**: 隐藏菜单

**password [--md5] STRING**: 启动菜单编辑认证

**title TITLE**: 定义菜单项“标题”, 可出现多次

**root (hd#,#)**: grub查找stage2及kernel文件所在设备分区; 为grub“根”

**kernel /PATH/TO/VMLINUZ\_FILE [PARAMETERS]**: 启动的内核

**initrd /PATH/TO/INITRAMFS\_FILE**: 内核匹配的ramfs文件

**password [--md5] STRING**: 启动选定的内核或操作系统时进行认证



❖ **grub-md5-crypt**命令

❖ 破解root口令:

启动系统时, 设置其运行级别1

❖ 进入单用户模式:

(1) 编辑grub菜单(选定要编辑的title, 而后使用e命令);

(2) 在选定的kernel后附加

1, s, S或single都可以;

(3) 在kernel所在行, 键入“b”命令

www.magedu.com

## ❖ 安装grub:

### (1) grub-install

安装grub stage1和stage1\_5到/dev/DISK磁盘上，并复制GRUB相关文件到 DIR/boot目录下

```
grub-install --root-directory=DIR /dev/DISK
```

### (2) grub

```
grub> root (hd#,#)
```

```
grub> setup (hd#)
```

马哥教育  
www.magedu.com

## ❖ 分区并创建文件系统

`fdisk /dev/sdb`

分两个必要的分区

`/dev/sdb1`对应`/boot`    `/dev/sdb2`对应根 `/`

`mkfs.ext4 /dev/sdb1`

## ❖ 挂载boot

`mkdir /mnt/boot`

`mount /dev/sdb1 /mnt/boot`

## ❖ 安装grub

`grub-install --root-directory=/mnt /dev/sdb`

## ❖ 恢复内核和initramfs文件

```
cp /boot/vmlinuz-2.6.32-642.el6.x86_64  
/mnt/boot/
```

```
cp /boot/initramfs-2.6.32-642.el6.x86_64.img  
/mnt/boot
```

## ❖ 建立grub.conf文件

```
Vim /mnt/boot/grub.conf
```

```
kernel /vmlinuz-2.6.32-642.el6.x86_64  
root=/dev/sda2 selinux=0 init=/bin/bash  
chroot /mnt/sysroot
```

## ❖ 创建一级目录

```
mkdir /mnt/sysroot
```

```
mount /dev/sdb2 /mnt/sysroot
```

```
mkdir -pv
```

```
/mnt/sysroot/{etc,lib,lib64,bin,sbin,tmp,var,usr,sys,proc,  
,opt,home,root,boot,dev,mnt,media}
```

## ❖ 复制bash和相关库文件

马哥教育

www.magedu.com

- ❖ 在根文件系统无法使用时需要，如/bin/mount删除
- ❖ 对系统没有特殊要求
- ❖ 从光盘引导（boot.iso或者安装光盘#1）
- ❖ 从USB盘（由boot.iso制作）引导

马哥教育

www.magedu.com



## ❖ 文件系统重组

Anaconda将会询问是否应该挂载文件系统

`/mnt/sysimage/*`

`/mnt/stage2`

`$PATH`包括硬盘的目录

## ❖ 文件系统节点

提供系统特定的设备文件

`mknod`了解major/minor #'s

马哥教育

www.magedu.com

- ❖ 系统在引导期间，很重要的一个过程就是init进程读取其配置文件/etc/inittab，启动系统基本服务程序及默认运行级别的服务程序完成系统引导，如果/etc/inittab误删除或修改错误，Linux将无法启动。此时，只有通过救援模式才可以解决此类问题。
  - 有备份文件的回复方法
  - 没有备份文件的恢复办法

马哥教育

www.magedu.com

## ❖ 有备份文件的恢复办法:

进入救援模式，执行**chroot**命令后，如果有此文件的备份（强烈建议系统中的重要数据目录，如**/etc**、**/boot**等要进行备份），直接将备份文件拷贝回去，退出重启即可。如果是配置文件修改错误，如比较典型的**/boot/grub/grub.conf**及**/etc/passwd**的文件修改错误，也可以直接修正恢复。假设有备份文件**/etc/inittab.bak**，则在救援模式下执行：

```
sh-3.1# chroot /mnt/sysimage
```

```
sh-3.1# cp /etc/inittab.bak /etc/inittab
```

# 系统配置文件丢失修复

## ❖ 没有备份文件的恢复办法

如果一些配置文件丢失或软件误删除，且无备份，可以通过重新安装软件包来恢复，首先查找到`/etc/inittab`属于哪一个RPM包

```
# chroot /mnt/sysimage
```

```
# rpm -qf /etc/inittab
```

```
initscripts-9.03.49-1.el6.centos.x86_64
```

退出chroot模式：

```
# exit
```

挂载存放RPM包的安装光盘（在救援模式下，光盘通常挂载在  
`/mnt/source`目录下）：  
www.magedu.com

# 系统配置文件丢失修复

```
# mount /dev/sr0 /mnt/source
```

CentOS6系统的RPM包存放在光盘Package目录下，另外，因为要修复的硬盘系统的根目录在/mnt/sysimage下，需要使用-**root**选项指定其位置。覆盖安装/etc/inittab文件所在的RPM包：

```
# rpm -ivh --replacepkgs | force /mnt/source/Packages/  
initscripts-9.03.49-1.el6.centos.x86_64.rpm
```

其中的rpm命令选项“**--replacepkgs**”表示覆盖安装，执行完成后，即已经恢复了此文件。

马哥教育  
www.magedu.com

# 系统配置文件丢失修复

如果想只提取RPM包中的/etc/inittab文件进行恢复，可以在进入救援模式后，执行命令：

```
# rpm2cpio /mnt/source/Packages/initscripts-9.03.49-1.el6.centos.x86_64.rpm |
```

```
cpio -idv ./etc/inittab
```

```
# cp etc/inittab /mnt/sysimage/etc
```

注意此命令执行时不能将文件直接恢复至/etc目录，只能提取到当前目录下，且恢复的文件名称所在路径要写完整的路径。提取文件成功后，将其复制到根分区所在的/mnt/sysimage目录下相应位置即可

www.magedu.com



# 内核编译

❖ 单内核体系设计、但充分借鉴了微内核设计体系的优点，为内核引入模块化机制。

❖ 内核组成部分：

**kernel**：内核核心，一般为**bzImage**，通常在**/boot**目录下，名称为**vmlinuz-VERSION-RELEASE**；

❖ **kernel object**：内核对象，一般放置于**/lib/modules/VERSION-RELEASE/**

[ ]: N

[M]: M

[\*]: Y

马哥教育  
www.magedu.com

辅助文件：**ramdisk**

**initrd**

**initramfs**



❖ 运行中的内核:

uname命令:

uname - print system information

uname [OPTION]...

-n: 显示节点名称;

-r: 显示VERSION-RELEASE;

-a:显示所有信息

马哥教育

www.magedu.com

# 内核模块命令

## ❖ lsmod命令:

显示由核心已经装载的内核模块

显示的内容来自于: `/proc/modules` 文件

## ❖ modinfo命令:

显示模块的详细描述信息

`modinfo [ -k kernel ] [ modulename|filename... ]`

`-n`: 只显示模块文件路径

`-p`: 显示模块参数

`-a`: author

`-d`: description

`-l`: license

`lsmod |grep xfs;modinfo xfs`

# 内核模块管理

❖ **modprobe**命令:

装载或卸载内核模块

❖ **modprobe** [ -C config-file ] [ modulename ] [ module  
parame-ters... ]

❖ 配置文件: /etc/modprobe.conf,  
/etc/modprobe.d/\*.conf

❖ **modprobe** [ -r ] modulename...

马哥教育

www.magedu.com

# 内核模块管理

## ❖ depmod命令:

内核模块依赖关系文件及系统信息映射文件的生成工具

## ❖ 装载或卸载内核模块:

## ❖ insmod命令: 指定模块文件, 不自动解决依赖模块

```
insmod [ filename ] [ module options... ]
```

```
insmod `modinfo -n exportfs`
```

```
insmod `modinfo -n xfs`
```

## ❖ rmmod

```
rmmod [ modulename ]
```

```
rmmod xfs
```

```
rmmod exportfs
```

## ❖ /proc 目录:

- ❖ 内核把自己内部状态信息及统计信息，以及可配置参数通过proc伪文件系统加以输出

参数：只读：输出信息

可写：可接受用户指定“新值”来实现对内核某功能或特性的配置

## ❖ /proc/sys

(1) sysctl命令用于查看或设定此目录中诸多参数

`sysctl -w path.to.parameter=VALUE`

`sysctl -w kernel.hostname=mail.magedu.com`

(2) echo命令通过重定向方式也可以修改大多数参数的值

`echo "VALUE" > /proc/sys/path/to/parameter`

`echo "websrv" > /proc/sys/kernel/hostname`

## ❖ sysctl命令:

默认配置文件: `/etc/sysctl.conf`

### (1) 设置某参数

`sysctl -w parameter=VALUE`

### (2) 通过读取配置文件设置参数

`sysctl -p [/path/to/conf_file]`

## ❖ 内核中的路由转发:

`/proc/sys/net/ipv4/ip_forward`

## ❖ 常用的几个参数:

`net.ipv4.ip_forward`

`net.ipv4.icmp_echo_ignore_all`

`vm.drop_caches`

## ❖ /sys目录:

**sysfs:** 为用户使用的伪文件系统，输出内核识别出的各硬件设备的相关属性信息，也有内核对硬件特性的设定信息；有些参数是可以修改的，用于调整硬件工作特性。

**udev**通过此路径下输出的信息动态为各设备创建所需要设备文件，**udev**是运行用户空间程序

专用工具：**udevadmin, hotplug**

**udev**为设备创建设备文件时，会读取其事先定义好的规则文件，一般在**/etc/udev/rules.d**及**/usr/lib/udev/rules.d**目录下

www.magedu.com



## ❖ ramdisk文件的制作:

### (1) mkinitrd命令

为当前正在使用的内核重新制作ramdisk文件

```
mkinitrd /boot/initramfs-$(uname -r).img $(uname -r)
```

### (2) dracut命令

为当前正在使用的内核重新制作ramdisk文件

```
dracut /boot/initramfs-$(uname -r).img $(uname -r)
```

马哥教育

www.magedu.com

## ❖ 前提:

- (1) 准备好开发环境
- (2) 获取目标主机上硬件设备的相关信息
- (3) 获取目标主机系统功能的相关信息  
例如:需要启用相应的文件系统
- (4) 获取内核源代码包  
[www.kernel.org](http://www.kernel.org)

马哥教育

[www.magedu.com](http://www.magedu.com)

## ❖ 包组(CentOS 6):

Server Platform Development  
Development Tools

## ❖ 目标主机硬件设备相关信息:

CPU:

```
#cat /proc/cpuinfo
```

```
#x86info -a
```

```
#lscpu
```

马哥教育

www.magedu.com

## ❖ PCI设备:

lspci

-v

-vv

lsusb

-v

-vv

lsblk 块设备

## ❖ 了解全部硬件设备信息

hal-device

# 内核编译安装系统

- ❖ 安装开发包组
- ❖ 下载源码文件
- ❖ `.config`: 准备文本配置文件
- ❖ `make menuconfig`: 配置内核选项
- ❖ `make [-j #]`
- ❖ `make modules_install`: 安装模块
- ❖ `make install` : 安装内核相关文件

安装bzImage为/`boot/vmlinuz-VERSION-RELEASE`

生成initramfs文件

编辑grub的配置文件

# 编译安装内核实例

- ❖ `tar xf linux-3.10.67.tar.xz -C /usr/src`
- ❖ `cd /usr/src`
- ❖ `ln -sv linux-3.10.67 linux`
- ❖ `cd /usr/src/linux`
- ❖ `cp /boot/config-$(uname -r) ./config`
- ❖ `make menuconfig`
- ❖ `make -j 2`
- ❖ `make modules_install`
- ❖ `make install`
- ❖ `reboot`

## ❖ (1) 配置内核选项

支持“更新”模式进行配置：**make help**

(a) **make config**: 基于命令行以遍历的方式去配置内核中可配置的每个选项

(b) **make menuconfig**: 基于curses的文本窗口界面

(c) **make gconfig**: 基于GTK (GNOME) 环境窗口界面

(d) **make xconfig**: 基于QT(KDE)环境的窗口界面

支持“全新配置”模式进行配置

(a) **make defconfig**: 基于内核为目标平台提供的“默认”配置进行配置

(b) **make allyesconfig**: 所有选项均回答为“yes”

(c) **make allnoconfig**: 所有选项均回答为“no”



## ❖ (2) 编译

❖ 全编译:`make [-j #]`

❖ 编译内核的一部分功能:

(a) 只编译某子目录中的相关代码:

```
# cd /usr/src/linux
```

```
# make dir/
```

(b) 只编译一个特定的模块:

```
# cd /usr/src/linux
```

```
# make dir/file.ko
```

例如: 只为e1000编译驱动:

```
#make drivers/net/ethernet/intel/e1000/e1000.ko
```

## ❖ 如何交叉编译内核：

编译的目标平台与当前平台不相同；

**# make ARCH=arch\_name**

## ❖ 要获取特定目标平台的使用帮助

**# make ARCH=arch\_name help**

**# make ARCH=arm help**

马哥教育

www.magedu.com

- ❖ 在已经执行过编译操作的内核源码树做重新编译：
- ❖ 需要事先清理操作：

**# make clean:** 清理大多数编译生成的文件，但会保留 **config** 文件等

**# make mrproper:** 清理所有编译生成的文件、**config** 及某些备份文件

**# make distclean:** **mrproper**、**patches** 以及编辑器备份文件

马哥教育

www.magedu.com

- ❖ 1、破解root口令，并为grub设置保护功能
- ❖ 2、破坏本机grub stage1，而后在救援模式下修复之
- ❖ 3、删除vmlinuz和initramfs文件，无法启动，恢复之
- ❖ 4、删除/etc/fstab和/boot目录的所有文件，并恢复之
- ❖ 5、增加新硬盘，在其上制作能单独运行kernel和bash的系统
- ❖ 6、编译安装kernel，启用支持ntfs文件系统功能

马哥教育

www.magedu.com

- ❖ 博客: <http://magedu.blog.51cto.com>
- ❖ 主页: <http://www.magedu.com>
- ❖ QQ: 1661815153, 113228115
- ❖ QQ群: 203585050, 279599283

马哥教育  
[www.magedu.com](http://www.magedu.com)



马哥教育  
最专业的Linux培训机构

# Thank You!