

如何实现 LINUX 操作系统的定制

王勇 潘清 付长冬 (北京怀柔 3380 信箱 76 号 101416)

摘要:本文主要介绍如何根据目标系统要求,对 LINUX 操作系统进行定制,包括核心裁减和配置、核心生成、核心装载等具体内容。

关键词: LINUX 核心 定制 组件 装载

1. 概述

LINUX 操作系统是可裁减的、模块化的操作系统,通过系统本身提供的工具和方法可以将 LINUX 操作系统定制为一个符合目标系统的操作平台,同时 LINUX 还将它的组件分为直接的核心组件和运行时可装载的模块。

LINUX 操作系统的基本组件包括:

- (1)根文件系统;
- (2)EIDE/RLL/MEM 驱动程序;
- (3)内存管理;
- (4)进程和调度管理;
- (5)一些必要的 I/O 子系统。

可裁减的组件包括:

- (1)文件系统,如 smbfs、minix、xiafs、msdos、umdos、sysv、isofs、hpfs、nfs;
- (2)网络协议,如 TCP/IP、IPX;
- (3)字符设备驱动程序;
- (4)块设备驱动程序;
- (5)各种网络设备部件,如 NE2000、3COM3C509 等;
- (6)SCSI 设备部件;
- (7)ISDN 设备部件等。

本文的目的是说明如何对 LINUX 操作系统定制,以满足不同目标平台的需要。各种不同的目标平台对操作系统的功能要求不一样,并且各种目标平台的系统配置也不完全相同,因而为适应不同目标平台的需要,将 LINUX 操作系统进行定制是非常必要的。本文第 2 节介绍具体的定制过程,而第 3 节讲述如何对系统进行配置,第 4 节重点介绍如何在系统配置后生成新的核心,第 5 节介绍了如何装载新的核心。

2. 定制过程

LINUX 操作系统的定制过程基本分为三个步骤:

- (1)核心配置;

- (2)核心生成;

- (3)系统装载。

具体操作步骤如下:

- (1)cd /usr/src/linux
- (2)make menuconfig
- (3)make dep
- (4)make clean
- (5)make zImage
- (6)cp arch/i386/boot/zImage /linux2
- (7)cd /etc
- (8)vi lilo.conf
- (9)lilo
- (10)reboot

步骤(2)根据具体需要,对核心组件或可装载模块进行取舍。而步骤(3)、(4)主要负责清理核心生成前的环境。步骤(5)生成一个新的 LINUX 核心。步骤(6)将核心映象文件 zImage 拷贝到根目录下。步骤(8)对/etc/lilo.conf 文件进行修改,增加一个启动参数组,或者对原来的启动参数组进行修改。而步骤(9)根据/etc/lilo.conf 文件内容对主引导扇区记录进行修改。

3. 核心系统配置

(1)核心配置过程。LINUX 核心配置可以利用 make config 进行命令行交互方式配置,也可以利用 make menuconfig 进行菜单方式配置(本文以 make menuconfig 为例说明配置过程)。

在执行 make menuconfig 命令后,系统配置界面上的选项如下:

- code maturity level options
- loadable module support // 可装载模块选项
- general setup // 系统创建的有关参数
- floppy ,ide, and other block devices //块设备
- networking options //网络协议

- scsi support //SCSI 设备
- network device support //网络设备
- isdn subsystem //ISDN 设备
- cdrom drivers //CDROM 设备
- filesystems //文件系统
- character drivers //字符设备
- sound //声音设备
- kernel hacking

系统组件共有四种类型:

[*] : 直接的核心组件, 创建新的核心时直接包括到核心中。

[] : 核心不包括的组件, 创建新的核心时核心不包括此组件。

< M > : 可装载的运行时模块, 不直接包括到核心但在运行时可动态装载。

< > : 可裁减但不包括的模块不是核心组件, 而是可裁减模块。

用户可以根据目标平台的要求, 利用系统提供的编辑功能对配置选项进行配置, 对各个子菜单选项, 可以进入子菜单进行具体的选项配置。配置退出时, 系统将有关参数保存在系统配置文件中。在创建新的核心时, 系统根据系统配置文件进行编译和创建。

(2) 核心数据文件。核心系统配置的数据文件为 kernel.config, 它保存了系统配置的结果。而 arch/i386/Config.in 文件是系统配置的主界面文件, 其他目录下的 Config.in 是相应的子菜单或选项文件。menuconfig 是个 SHELL 文件, 它解释并执行 Config.in 的内容。而 Makefile 文件, 它说明了核心如何编译, 不同部件是否可以编译, 根据系统配置文件的参数决定。

kernel.config 文件中对系统创建的有关参数配置结果如下:

```
# CONFIG_MATH_EMULATION is not set
CONFIG_NET=y
# CONFIG_MAX_16M is not set
CONFIG_PCI=y
CONFIG_SYSVIPC=y
CONFIG_BINFMT_AOUT=y
CONFIG_BINFMT_ELF=y
CONFIG_KERNEL_ELF=y
CONFIG_M586=y
```

对于系统创建的有关参数配置界面中的具体显示由 Config.in 文件中如下部分完成:

```
mainmenu_option next_comment
comment 'General setup'
bool 'Kernel math emulation' CONFIG_MATH_
EMULATION
bool 'Networking support' CONFIG_NET
bool 'Limit memory to low 16MB' CONFIG_MAX_
16M
bool 'PCI bios support' CONFIG_PCI
if [ "$CONFIG_PCI" = "y" ]; then
if [ "$CONFIG_EXPERIMENTAL" = "y" ];
then
bool 'PCI bridge optimization (experimental)' CON-
FIG_PCI_OPTIMIZE
fi
fi
bool 'System V IPC' CONFIG_SYSVIPC
tristate 'Kernel support for a.out binaries' CONFIG_
BINFMT_AOUT
tristate 'Kernel support for ELF binaries' CONFIG_
BINFMT_ELF
...
endmenu
```

(3) 增加新的组件。通过核心系统配置过程增加新的核心组件主要考虑二个问题:

- ① 如何在配置界面中增加新的配置选项
- ② 如何编译增加的组件, 并连接到核心

下面以增加一个新的文件系统 LOGFS 为例, 说明如何通过核心系统配置过程增加新的核心组件。

首先在 /usr/src/linux/fs 目录下, 修改 Config.in 文件, 在菜单项中增加如下行:

```
mainmenu_option next_comment
comment 'Filesystems'
...
tristate 'log file system support' CONFIG_LOG_FS
...
endmenu
```

然后在 Makefile 文件中增加如下行:

```
ifeq ($(CONFIG_LOG_FS), Y)
sub_dirs += logfs
```

在 logfs 目录下, 是 LOGFS 文件系统的源码, 包括一个 Makefile 文件。

最后利用 `make menuconfig` 和 `make zImage` 操作可以在核心中增加新的 LOGFS 文件系统组件。

4. 核心生成

(1) LINUX 核心构成。通过编译程序,可以清楚地了解 LINUX 操作系统核心的具体构成,其构成如下:

```
arch/i386/kernel/head.o ;init/main.o ;init/version.o
;
arch/i386/kernel/kernel.o ;arch/i386/mm/mm.o ;
fs/fs.o ;ipc/ipc.o ;
net/network.a ; fs/filesystems.a ; drivers/block/
block.a ;
drivers/char/char.a ;drivers/net/net.a ; drivers/pci/
pci.a /usr/src/linux-2.0.30/arch/i386/lib/lib.a ;
```

(2) 具体编译过程。LINUX 操作系统新的核心由如下命令创建:

```
make dep
make clean
make zImage 或 make zlilo
```

通过 `make zImage` 命令最后在 `/usr/src/linux/arch/i386/boot` 目录下生成 `zImage` 文件。此文件就是新的 LINUX 操作系统核心映象文件。具体执行过程步骤如下:

①根据根 Makefile 和层次性的 Makefile 进行 LINUX 操作系统各核心组件的编译,生成相应的 .a 或 .o 文件;

②利用 `ld` 命令生成 `vmlinuz` 文件;

③利用 `as86` 和 `ld86` 命令生成 `bootsect` 和 `setup` 文件;

④利用 `build` 命令生成 `zImage` 文件;

```
objdump -k -q -o 0x1000 compressed/vmlinuz >
compressed/vmlinuz.out
tools/build bootsect setup compressed/vmlinuz.out
CURRENT > zImage
```

5. 核心装载

在 LINUX 操作系统新的核心生成后,如何加载新的 LINUX 操作系统核心,需要考虑以下二个问题:

(1) 如何在主引导记录中登记新的核心文件;

(2) 不同的核心文件如何统一表示。

LINUX 操作系统提供了 `boot` 加载器安装命令 `lilo` 在 MBR 登记新的核心文件,配置文件 `lilo.conf` 说明核心

文件如何在 `/dev/hda` 的 MBR 中表示。

`lilo.conf` 文件的内容如下:

```
boot = /dev/hda # compact # faster, but won't
work on all systems.
```

```
delay = 5
vga = normal # force sane state
# ramdisk = 0 # paranoia setting
# Linux bootable partition config begins
```

```
image = /vmlinuz
```

```
root = /dev/hda1
```

```
label = linux
```

```
read-only
```

如果在配置文件 `lilo.conf` 中增加新的核心文件表示,只要在 `lilo.conf` 文件尾增加如下行:

```
# Linux bootable partition config begins
image = 新的 LINUX 操作系统核心名
root = /dev/hda1
label = LINUX-1 # 核心表示符
read-only
```

在修改 `lilo.conf` 文件后,执行 `/etc/lilo` 命令,就将 `lilo.conf` 文件的内容写到主引导记录 MBR 中。而 `reboot` 重新启动系统后,在 `boot:` 后输入表示符 `LINUX-1`,就加载新的 LINUX 操作系统核心。

6. 结束语

LINUX 操作系统是一个可裁减的模块化的操作系统,利用它的定制特性可以生成符合不同需要的目标平台。根据系统提供的配置选项可以对系统进行裁减,也可以在系统中增加新的配置选项,从而实现对目标平台的定制。对于 LINUX 操作系统的核心生成,需要对层次性的 Makefile 进行维护和修改。对于新的核心加载需要修改 `lilo.conf` 文件,并利用 `lilo` 命令修改主引导记录。

参考文献

- [1] [美] Jack Tackett Jr., LINUX 大全,电子工业出版社,1998
- [2] [英] Phil Cornes, Linux 从入门到精通,电子工业出版社,1998

(来稿时间:1999年8月)