

Содержание

1. Общие сведения о программе.....	4
2. Структура программы.....	4
2. Настройка программы.....	5
2.1 Создание LiveCD.....	5
2.2 Создание Initrd.....	6
2.3 Генерация дистрибутива LiveCD.....	7
3. Проверка системного ПО.....	9
Приложение А.....	11
Описание каталогов дистрибутива.....	11
Приложение Б.....	13
Текст и описание параметров файла isolinux.cfg.....	13
Приложение В.....	14
Содержимое каталогов образа initrd.....	14
Приложение Г.....	15
Текст скрипта linuxrc.....	15
Приложение Д.....	16
Текст скрипта docsetup.....	16
Приложение Е.....	17
Текст скрипта rc.s.....	17

Руководство системного программиста

1. Общие сведения о программе

K24_LiveCD представляет собой специализированный загрузочный компакт диск, основанный на открытой операционной системе Slackware10.2 (2.6.6) со специальным набором утилит и библиотек для работы с одноплатной ЭВМ поддерживающей DiskOnChip (далее DOC) носитель. Диск в частности предназначен для записи на заранее отформатированный DOC специально собранного дистрибутива Linux Slackware.

2. Структура программы

Загрузочный диск(далее LiveCD) включает три директории: isolinux, kernels/bare.i/, live.

Isolinux- каталог содержит файлы необходимые для начальной загрузки системы. Isolinux это загрузчик, позволяющий загружаться с CD, как с загрузочной дискеты.

Kernels/bare.i – каталог, содержит ядро операционной системы.

live – каталог содержит в себе дистрибутив ОС Linux, который предстоит записать на диск. В данном каталоге также находится дистрибутив для DOC носителя.



рис. 1 Этапы работы загрузочного диска K24_Livecd

2. Настройка программы

2.1 Создание LiveCD

Для создания LiveCD необходимо войти в систему с правами суперпользователя. Необходимо создать набор каталогов для LiveCD:

```
mkdir /root/livecd
mkdir /root/livecd/live
mkdir /root/livecd/kernels
mkdir /root/livecd/kernels/bare.i/
mkdir /root/livecd/isolinux
```

Каталог **live** содержит следующее дерево каталогов:

```
bin/  boot/  dev/
etc/  home/  lib/
mnt/  proc/  root/
sbin/ tmp/   usr/
```

/Isolinux – каталог, содержащий `initrd` - образ, `isolinux.cfg` – конфигурационный файл загрузчика `initrd`, `isolinux.b`, `isolinux.c`. Текст и описание параметров файла `isolinux.cfg` находится в приложении А.

Формат файла `isolinux.cfg` очень похож на формат файла `Lilo.conf`(man `lilo.conf`).

/Kernels/bare.i – каталог, содержащий ядро `bzImage` и `.config` – конфигурационный файл ядра. Подробнее о параметрах сборки ядра см. «Руководство системного программиста. Драпико Е.А.»[4]

На данном этапе дистрибутив компакт диска сформирован. Необходимо сделать диск загрузочным.

Для этого сформируем загрузчик `initrd`, в который скопируем минимальное дерево каталогов, библиотек, конфигурационных файлов и скриптов, для того чтобы система вначале загрузила этот образ, а далее сменила этот образ на каталог дистрибутива компакт диска(`/live`), который будет сформирован после создания `initrd`-образа.

2.2 Создание Initrd

Файл `initrd` (init Ram Disc) содержит корневую файловую систему дистрибутива. `initrd` будет содержать только системные файлы, поэтому 5-ти мегабайт будет достаточно.

Зарезервируем 5 мегабайт памяти для `Initrd`:
`dd if=/dev/zero of=/dev/ram0 bs=1k count=5000`

Создадим файловую систему Linux – ext2:
`mke2fs -m0 /dev/ram0 5000`

- `m0` определяет сколько процентов файловой системы выделить для пользователя `root`. Значение по умолчанию равно 5%.

монтируем часть памяти в директорию `/mnt/distr` для дальнейшего формирования образа `Initrd`

```
mount /dev/ram0 /mnt/distr
```

Сформируем маленький образ, в котором помимо самой системы будет еще 2 скрипта (`linuxrc` и `userface`).

Дерево каталогов образа `initrd` выглядит так:

bin/	dev/	etc/
lib/	linuxrc*	mnt/
proc/	root/	sbin/
sys/	userface *	var/

Содержимое каждого каталога дано в приложении В Этого достаточно для начальной загрузки ОС. Система после загрузки образа запускает скрипт `linuxrc`. Текст скрипта содержится в приложении Г.

Главная задача `linuxrc` монтирование `cd-rom` и `DiskOnChip`, вывод информации о корректном монтировании, запуск процесса `init` и смена корневой директории на директорию `/mnt/cdrom`, которая дает возможность работать уже не с 5 мб. образом, а с 700-750 мб. В `linuxrc` также осуществляется вызов другого скрипта – `userface`, в котором описаны функции для наглядного представления этапа загрузки.

Наконец, создаем файл `initrd`:

```
Umount /dev/ram0  
dd if=/dev/ram0 of=initrd bs=1k count=5000
```

Теперь имеем образ `initrd`, который скопируем в `isolinux`, и который будет необходим для начальной загрузки системы.

2.3 Генерация дистрибутива LiveCD

Теперь сформируем образ дистрибутива, который будет на CD и позволит работать уже с большим объемом информации.

Необходим следующий набор утилит:

- Стандартный набор утилит (работа с файлами и папками, проверка файловой системы, fdisk, и т.д.)
- Работа с модулями (insmod, rmmod, modprobe)
- Работа с ядром, которое далее скопируем на DOC.
- Midnight Commander
 - дистрибутив (l325mod)
 - скрипт работы с DOC

Дистрибутив (l325mod) будет скопирован на DOC-носитель. DOC имеет 32 мегабайта памяти и этого достаточно для записи на него полноценной рабочей ос. Структура каталогов дистрибутива l325mod :

bin\ \	dev\ \	etc\ \	home\ \
lib\ \	mnt \	proc\ \	root\ \
sbin \	tmp \	usr\ \	var\ \

Это обычный минимально собранный дистрибутив Linux для записи на DOC.

Для записи диска необходимо создать загрузочный iso-образ. Для создания образа воспользуемся стандартной утилитой mkisofs.

```
mkisofs -o PATH_TO_ISO.iso -R -J -hide-rr-moved -v -d -N -no-emul-boot -boot-info-table -boot-load-size 4 -b isolinux/isolinux.bin -c isolinux/isolinux.boot /path_to_distr
```

Проверить образ перед записью можно так:

```
Mount -o loop /path_to_iso /mnt/distr
```

Теперь запишем этот образ на компакт диск с помощью утилиты cdrecord.

```
Cdrecord -v speed=12 dev=/dev/cdrom /path_to_iso
```

Полная/быстрая очистка диска:

```
Cdrecord blank=all speed=12 dev=/dev/cdrom
```

```
Cdrecord blank=fast speed=12 dev=/dev/cdrom
```

После успешной записи CD перезагружаем компьютер.

После перезагрузки необходимо зайти в настройки BIOS и установить загрузку со SCSI устройства. В зависимости от версии BIOS для успешной загрузки с DOC может понадобиться отключить (аппаратно) жесткий диск установленный в системе. После выхода из настроек BIOS система загрузится с DOC[4] После загрузки с CD можно приступить к работе с DiskOnChip.

Для этого был написан скрипт docsetup. Текст скрипта находится в приложении Д.

При загрузке с DOC вместо linuxrc будет запускаться скрипт gc.s, специально написанный для DiskOnChip. Текст скрипта gc.s находится в приложении Е.

После загрузки пользователь получит приглашение о регистрации. На данном этапе *работе* с DiskOnChip завершена.

3. Проверка системного ПО

1. Вставьте компакт-диск в CD привод.
2. В BIOS назначьте загрузку с CD
3. Загружаемся с LiveCD
4. Устанавливаем дистрибутив I325mod на DOC командой `docsetup`
5. Перезагружаемся
6. В BIOS выставляем загрузку со SCSI устройства.
7. После загрузки должно появиться окно регистрации пользователя (`login:root`, `Password: toor` или `ghbdtm`)

Возможные проблемы с docsetup:

Если по каким-либо причинам команда `docsetup` не работает, то после загрузки LiveCD в консоли выполняем следующие действия:

1. Монтируем DOC в каталог `/mnt/doc`: `mount /dev/nftla1 /mnt/doc`
2. Копируем дерево каталогов дистрибутива I325mod на DiskOnChip в `/mnt/doc`
3. Монтируем cdrom: `mount /dev/sr0 /mnt/cdrom`
4. Переходим в каталог с ядром для DOC: `cd /mnt/cdrom/kernels/bare.i`
5. Копируем `bzImage` в `/boot` на DiskOnChip
6. Переходим в каталог `/boot` на DOC: `cd /mnt/doc/boot`
7. Изменяем устройство загрузки по умолчанию: `rdev bzImage /dev/nftla1`
8. Устанавливаем загрузчик в MBR носителя командой `lilo-mtd -r /mnt/doc -C /etc/lilo.conf`
9. Перезагружаемся
10. В BIOS выставляем загрузку со SCSI устройства
11. После загрузки должно появиться окно регистрации пользователя (`login:root`, `Password: toor` или `ghbdtm`)

Список литературы:

1. [http:// www.opennet.ru](http://www.opennet.ru)
2. Основы Slackware Linux - Алан Хикс, Павел Марьянов, Крис, Люменс, Дэвид Кантрелл, Logan Джонсон
3. Building a Slackware Live CD Daniel de Kok
4. Руководство системного программиста. Драпико Е.А.
5. Ставим ядро 2.6, или Ядерная физика для домохозяйки. Версия 2.0 Зарипов Михаил
6. <http://www.linux-live.org>
7. <http://slax.linux-live.org>
8. <http://slackware.com>
9. Операционная система UNIX – Андрей Робачевский
10. slackware.tomsk.ru
11. Linux: Ответы и Советы И. Грень – Мн.: Новое знание, 2004. – 350с.
12. <http://kernel.org>
13. <http://www.mgul.ac.ru/~t-alex/Linux/Bootdisk-HOWTO.htm#ss3.1>
14. <http://www.tolkuchka.dsip.net/index.php?showtopic=22625>
15. <http://citkit.ru/articles/53/>
16. Slackware Essentials Alan Hicks
17. Операционная система Linux INTUIT.ru

Приложение А

Описание каталогов дистрибутива

/bin

Название этого каталога происходит от слова "binaries" ("двоичные", "исполняемые"). В этом каталоге находятся исполняемые файлы самых необходимых утилит. Здесь содержатся такие программы, которые могут понадобиться системному администратору или другим пользователям для устранения неполадок в системе или при восстановлении после сбоя.

/boot

"Boot" - загрузка системы. В этом каталоге находятся файлы, необходимые для самого первого этапа - загрузки ядра - и, обычно, само ядро. Пользователю практически никогда не требуется непосредственно работать с этими файлами.

/dev

В этом каталоге находятся все имеющиеся в системе файлы особого типа, предназначенные для обращения к различным системным ресурсам и устройствам (англ. "devices" - "устройства", отсюда и сокращенное название каталога). Например, файлы /dev/ttyN соответствуют виртуальным консолям, где N - номер виртуальной консоли. Данные, введенные пользователем на первой виртуальной консоли, система считывает из файла /dev/tty1; в этот же файл записываются данные, которые нужно вывести пользователю на эту консоль. В файлах-дырках в действительности не хранятся никакие данные, при их помощи данные передаются.

Каталог для системных конфигурационных файлов. Здесь хранится информация о специфических настройках данной системы: информация о зарегистрированных пользователях, доступных ресурсах, настройках различных программ.

/etc – здесь расположены различные конфигурационные файлы.

/home

Здесь расположены каталоги, принадлежащие пользователям системы - домашние каталоги, отсюда и название "home". Отделение всех файлов, создаваемых пользователями, от прочих системных файлов дает очевидное преимущество: серьезное повреждение системы или необходимость обновления не затронет наиболее ценной информации - пользовательских файлов.

/lib

Название этого каталога - сокращение от "libraries" (англ. "библиотеки"). Библиотеки – это собрания стандартных функций, необходимых многим программам: операций ввода/вывода, рисования элементов графического интерфейса и т. д. Чтобы не включать эти функции в текст каждой программы, используются стандартные функции библиотек - это значительно экономит место на диске и упрощает написание программ. В этом каталоге содержатся библиотеки, необходимые для работы наиболее важных системных утилит (размещенных в /bin и /sbin). Каталог для монтирования (от англ. "mount") -

временного подключения файловых систем, например, на съемных носителях (CD-ROM и др.).

/proc

В этом каталоге все файлы "виртуальные" - они располагаются не на диске, а в оперативной памяти. В этих файлах содержится информация о программах (процессах), выполняемых в данный момент в системе.

/root

Домашний каталог администратора системы - пользователя root. Смысл размещать его отдельно от домашних каталогов остальных пользователей состоит в том, что /home может располагаться на отдельном устройстве, которое не всегда доступно (например, на сетевом диске), а домашний каталог root должен присутствовать в любой ситуации.

/sbin

Каталог для важнейших системных утилит (название каталога - сокращение от "system binaries"): в дополнение к утилитам /bin здесь находятся программы, необходимые для загрузки, резервного копирования, восстановления системы. Полномочия на исполнение этих программ есть только у системного администратора.

/tmp

Этот каталог предназначен для временных файлов: в таких файлах программы хранят необходимые для работы промежуточные данные. После завершения работы программы временные файлы теряют смысл и должны быть удалены. Обычно каталог /tmp очищается при каждой загрузке системы. Каталог

/usr

Здесь можно найти такие же подкаталоги bin, etc, lib, sbin, как и в корневом каталоге. Однако в корневой каталог попадают только утилиты, необходимые для загрузки и восстановления системы в аварийной ситуации - все остальные программы и данные располагаются в подкаталогах /usr. Прикладных программ в современных системах обычно установлено очень много, поэтому этот раздел файловой системы может быть очень большим.

/var

Название этого каталога - сокращение от "variable" ("переменные" данные). Здесь размещаются те данные, которые создаются в процессе работы разными программами и предназначены для передачи другим программам и системам (очереди печати, электронной почты и др.) или для сведения системного администратора (системные журналы, содержащие протоколы работы системы). В отличие от каталога /tmp сюда попадают те данные, которые могут понадобиться после того, как создавшая их программа завершила работу.

Приложение Б

Текст и описание параметров файла isolinux.cfg

```
default /kernels/bare.i/bzImage initrd=initrd init=linuxrc load_ramdisk=1
prompt_ramdisk=0 ramdisk_size=5000 root=/dev/ram0 rw
prompt 10
timeout 400
display message.txt
```

Default – устанавливает способ загрузки ядра по умолчанию

/kernels/bare.i/bzImage – относительный путь до ядра ОС.

initrd=initrd – имя initrd образа

init=linuxrc – имя главного инициализационного скрипта

load_ramdisk=1 – *разрешить* загружать образ в память

ramdisk_size=5000 – размер образа

root=/dev/ram0 – путь до корня файловой системы

rw - разрешение на чтение и запись.

Приложение В

Содержимое каталогов образа initrd

Каталог /bin			
Bash	cat	chroot	cp
hostname	lilo-mtd	login	ls
Mkdir	mount	pivot_root	rdev
Rm	sh	sleep	stty
Sync	umount		

Каталог /etc			
Group	hosts	inittab	mtab
passwd	shadow	shadow-	
rc.d/	rc.s	rc.sysvinit	

Каталог /lib					
ld-2.3.5.so	ld-linux.so.2	libblkid.so.1	libblkid.so.1.0	libc-2.3.5.so	libc.so.6
libcrypt-2.3.4.so	libcrypt-2.3.5.so	libcrypt.so.1	libdl-2.3.5.so	libdl.so.2	libm-2.3.5.so
libm.so.6	@libnsl-2.3.4.so	libnsl.so.1	libnss_files-2.3.4.so	libnss_files.so.2	libpthread-0.10.so
libpthread.so.0	@librt-2.3.5.so	librt.so.1	@libtermcap.a	libtermcap.so	libtermcap.so.2
libtermcap.so.2.0.8	libthread_db-1.0.so	libthread_db.so.1	libuuid.so.1	libuuid.so.1.2	

/sbin	
busybox	halt
init	poweroff
reboot	runlevel
shutdown	

Приложение Г

Текст скрипта linuxrc

```
#!/bin/sh
#linuxrc - main initialization script

PATH=".::/usr/sbin:/usr/bin:/sbin:/bin"
livedir=/mnt/cdrom/live
docdir=$livedir/mnt/doc
TERM=linux
export PATH TERM VERSION BOOT BOOTFS FLCFG FXCFG FIXED
. userface
header "Startin' K24 initialization Script"

echolog "Mountin' CD to /mnt/cdrom"
if mount /dev/sr0 /mnt/cdrom
then echolog "CD was mounted"
else echolog "Error: CD Wasn't Mounted"
fi

echolog "mountin' /proc filesystems"
mount -t proc proc $livedir/procVERS="0.1.1"
header "Version: $VERS"

echolog "Changin' hostname to DOC"
/bin/hostname DOC

echolog "Mountin' DOC to /mnt/doc"
if mount $livedir/dev/nftla1 $livedir/mnt/doc
then echolog "DOC was mounted"
else
echolog "Error: DOC Wasn't Mounted"
fi

header "Now u have eNtered to the K24_LiveCd"
echo "Call init"
exec /sbin/init
```

Приложение Д

Текст скрипта docsetup

```
#!/bin/sh
echo "Mountin' CD to /mnt/cdrom"
  if mount /dev/sr0 /mnt/cdrom
  then
    echo "CD was mounted"
  else
    echo "Error: CD Wasn't Mounted"
  fi

echo "Mountin' DOC to /mnt/doc"
  if mount /dev/nftla1 /mnt/doc
  then
    echo "DOC was mounted"
  else
    echo "Error: DOC Wasn't Mounted. Maybe it was mounted before?"
  fi
echo "Startin' DOC distribution"

echo "Delete oLD TRee FRom DOC"
rm -r /mnt/doc/*

echo "copy /mnt/doc /home/l325mod"
cp -R /home/l325mod/* /mnt/doc

echo "copy /mnt/home /home/l325mod"
cp -R /home/mup mnt/doc/home/

echo "rdev kernel"
rdev mnt/doc/boot/bzImage /dev/nftla1

echo "Patchin' kernel lilo-mtd"
lilo-mtd -r /mnt/doc -C /etc/lilo.conf
echo "DOC distribution are finished. Reboot and try to load from DiskOnChip"
```

Приложение E

Текст скрипта rc.s

```
#!/bin/sh
```

```
PATH=/sbin:/usr/sbin:/bin:/usr/bin
```

```
# Mount /proc right away:
```

```
/bin/mount -v proc /proc -n -t proc
```

```
echo "fsckin dOc fS"
```

```
/sbin/e2fsck -a /dev/nftla1
```

```
# Remount the root filesystem in read-write mode
```

```
echo "rEMounTin DoC Fs iN Rw"
```

```
/bin/mount -w -v -n -o remount /
```

```
# Any /etc/mtab that exists here is old, so we delete it to start over:
```

```
/bin/rm -f /etc/mtab*
```

```
# Remounting the / partition will initialize the new /etc/mtab:
```

```
/bin/mount -w -o remount /
```

```
mount -f /procmount -f /
```

```
echo "seTiN DaTe"
```

```
/sbin/hwclock -s -u
```

```
# This loads any kernel modules that are needed. These might be required to
```

```
# use your ethernet card, sound card, or other optional hardware.
```

```
if [ -x /etc/rc.d/rc.modules -a -r /proc/modules ]; then
```

```
    . /etc/rc.d/rc.modules
```

```
fi
```

```
# Configure runtime kernel parameters:
```

```
if [ -x /sbin/sysctl -a -r /etc/sysctl.conf ]; then
```

```
    /sbin/sysctl -e -p /etc/sysctl.conf
```

```
fi
```

```
echo "SeT uP NeT"  
/bin/ifconfig eth0 195.208.165.108 broadcast 195.208.165.255 netmask 255.255.255.0  
/sbin/route add -net default gw 195.208.165.250 metric 1
```

```
echo "sEt uP HosTNaMe"  
/bin/hostname doc1
```

```
# Carry an entropy pool between reboots to improve randomness.
```

```
if [ -f /etc/random-seed ]; then
```

```
    echo "Using /etc/random-seed to initialize /dev/urandom."
```

```
    cat /etc/random-seed > /dev/urandom
```

```
fi
```

```
# Use the pool size from /proc, or 512 bytes:
```

```
if [ -r /proc/sys/kernel/random/poolsize ]; then
```

```
    dd if=/dev/urandom of=/etc/random-seed count=1 bs=$(cat  
/proc/sys/kernel/random/poolsize) 2> /dev/null
```

```
else
```

```
    dd if=/dev/urandom of=/etc/random-seed count=1 bs=512 2> /dev/null
```

```
fi
```

```
chmod 600 /etc/random-seed
```

```
echo "sTaRtiN oPenSSh deamon"
```

```
/etc/rc.d/rc.sshd start
```

```
echo "StarTiN vsFTp DaemON"
```

```
/usr/sbin/vsftpd &
```