

# The Debian Cookbook



Михаил Конник aka *virens*  
По следам *Записок Дебианщика*

Данная работа распространяется под лицензией Creative Commons Attribution-Non-Commercial-Share Alike 3.0 Unported License. Вы можете копировать, распространять, показывать эту работу, и создавать производные работы в некоммерческих целях на условиях: 1) обязательной ссылки на автора (*virens*, <http://mydebianblog.blogspot.com/>) и 2) распространении любых производных работ на условиях этой же лицензии (ссылка на эту лицензию обязательна!).

**Пожалуйста соблюдайте условия лицензии.**

# Оглавление

<b>1</b>	<b>Анализ данных</b>	<b>1</b>
1.1	Свободные утилиты восстановления данных . . .	1
1.2	Глубокий анализ данных, Эпизод 1: foremost . .	3
1.3	Глубокий анализ данных, Эпизод 2: The Sleuth Kit	7
1.4	Глубокий анализ данных, Эпизод 3: повреждённые разделы . . . . .	11
1.5	Отчего умирают винчестеры: небольшой обзор причин выхода из строя жёстких дисков . . . . .	14
<b>2</b>	<b>Эмуляция</b>	<b>1</b>
2.1	Создание диска IMG для QEMU . . . . .	1
2.2	USB в эмуляторе QEMU . . . . .	1
2.3	Монтирование диска QEMU . . . . .	3
2.4	Послать в QEMU сочетание клавиш . . . . .	3
2.5	Перегон образов QEMU в VirtualBox . . . . .	3
<b>3</b>	<b>Системное</b>	<b>1</b>
3.1	Монтирование разделов в Linux . . . . .	1
3.2	Файл подкачки: swap-файл и swap-раздел в Linux	3
3.3	Как создать ramdisk в Linux? . . . . .	6
3.4	Два слова о /proc . . . . .	7
3.5	cfdisk и mkfs: разметка и форматирование разделов в Linux . . . . .	12

3.6	Дата и время: date в Linux . . . . .	15
<b>4</b>	<b>Железо</b>	<b>1</b>
4.1	Как создать ramdisk в Linux . . . . .	1
4.2	Запись CD и DVD дисков в Linux . . . . .	2
4.3	Проверка правильности записи на CD и DVD диск	6
4.4	Ошибки при записи DVD и как с ними бороться	7
4.5	Как узнать подробности оборудования в Linux?	10
4.6	Настройка SANE: scanner in linux . . . . .	23
<b>5</b>	<b>Сети</b>	<b>1</b>
5.1	Швейцарский нож для сетей - netcat . . . . .	1
5.2	Настройка сетевого интерфейса в Linux . . . . .	2
5.3	Как узнать MAC адрес сетевой карты . . . . .	7
5.4	Настройка и использование SSH в Linux . . . . .	10
<b>6</b>	<b>Принтеры</b>	<b>1</b>
6.1	Сетевая печать из Linux на Windows принтер ис- пользуя SAMBA . . . . .	1
6.2	Сетевая печать: как печатать из Linux в Linux .	2
6.3	Как настроить принтеры с помощью CUPS . . .	3
<b>7</b>	<b>Ядро</b>	<b>1</b>
7.1	Управление ядром Linux реального времени . .	1
7.2	Как собрать ядро в Линукс . . . . .	3
<b>8</b>	<b>Продуктивность</b>	<b>1</b>
8.1	GraphViz: создание графов и взаимосвязей . . .	1
8.2	Презентации beamer . . . . .	3
<b>9</b>	<b>Данные</b>	<b>1</b>
9.1	EXIF теги в Linux . . . . .	1
9.2	Небольшая шпаргалка по командам SVN . . . .	3
9.3	Вертикальная конкатенация двух файлов . . . .	11
9.4	Как сделать бекап при помощи dd . . . . .	12
9.5	Subversion - быстрый старт . . . . .	15

<b>10 Цифровые камеры</b>	<b>1</b>
10.1 Экспозиция более 30 секунд на цифровой камере: Long-time remote shooting . . . . .	1
10.2 Настройка веб-камеры Logitech QuickCam Messenger в Linux . . . . .	7
<b>11 Навигация</b>	<b>1</b>
11.1 Закачка карт в GPS навигатор Garmin GPSMap 60CSx . . . . .	1
11.2 Garmin 60CSx: технические особенности устройства . . . . .	4
11.3 Маршруты и треки с помощью GPS-навигаторов Garmin . . . . .	6
<b>12 Мультимедиа</b>	<b>1</b>
12.1 МОСР . . . . .	1
12.2 Немного о lossless кодировании видео в Linux . . . . .	3
12.3 Видеозахват и сжатие видео в Linux . . . . .	6
12.4 Установка Mplayer в Linux . . . . .	9
12.5 Мультимедийные клавиши и кнопки в Linux . . . . .	14
<b>13 Изображения</b>	<b>1</b>
13.1 Чтение MAT-файлов MATLAB в nir2 . . . . .	1
13.2 Установка и впечатления новой версии графического анализатора NIR2 . . . . .	6
13.3 Продвинутое средства анализа изображений в nir2 . . . . .	8
<b>14 Интернет</b>	<b>1</b>
14.1 настроить сетевую карту в Linux . . . . .	1
14.2 Wget - повелитель закачек, или Как использовать wget . . . . .	7
<b>15 Научное</b>	<b>1</b>
15.1 Научная поисковая система на вашем Linux-десктопе . . . . .	1
15.2 Совместная работа над документами L <sup>A</sup> T <sub>E</sub> X . . . . .	3

<b>16 Debian-специфичное</b>	<b>1</b>
16.1 Поиск пакета по меткам в Debian GNU/Linux:	
debtags . . . . .	1
<b>17 ЛaTeX</b>	<b>1</b>
17.1 Мелкие тонкости презентаций в Beamer . . . . .	1

## Глава 1

# Анализ данных

### 1.1 Свободные утилиты восстановления данных

Задача: восстановить информацию с повреждённого CD-диска, копирование с которого прерывается из-за ошибок чтения.

#### 1.1.1 Утилита dd

Старая, как сам UNIX, утилита, входящая решительно в любой нормальный дистрибутив UNIX-основанных систем. Более того, на блоге уже упоминался вариант её использования для проверки дисков. Теперь она послужит нам для нужд восстановления. Собственно, её использование приведёт к копированию данных, несмотря на ошибки:

```
$ dd if=/dev/cdrom of=~ /bad.iso bs=2048 conv=noerror
```

Игнорирование ошибок чтения обеспечивает именно последний параметр, при этом вы будете наблюдать многочисленные сообщения об ошибках.

### 1.1.2 Утилита `recoverdm`

В комплект поставки дистрибутивов она входит редко, но распространяется под лицензией GPL, и скачать её отсюда. Требует рутовых прав, так как использует некоторое "трюкачество" например сброс состояния контроллера и чтение "сырых" данных (RAW-data). Обеспечивает большую точность восстановления, чем `dd`, и больший набор параметров. Однако работает гораздо медленнее `dd` в силу того, что гораздо большее количество попыток чтения совершается. Способна восстанавливать данные с дискет, CD/DVD и жёстких дисков. Кроме того, поставляется с утилитой `mergebad`, которая полезна в случае, например, когда есть несколько компакт-дисков, и все они битые, а информация на них распределена. Компиляция из исходников посредством команды `make` не представляет никакой трудности. Нужно поставить на ней соответствующие права и выполнить команду:

```
$ recoverdm -t 20 -i /dev/hda -o bad.iso -l badsectors.bad
-n 10 -r 12
```

После чего начнётся медленное и печальное восстановление. Лучше всего ставить эту процедуру на ночь, так как при сколько-нибудь обширных сбоях чтения компакт-дисков такое считывание может занять несколько часов (мой злосчастный миморекс читался 7 часов). В процессе считывания вы будете видеть в консоли такие вот сообщения:

```
Trying RAW read...
Tried reading 10 times, failed doing that. Continuing...
error at 732188672 bytes, retrying: 1
error at 732188672 bytes, retrying: 2
error at 732188672 bytes, retrying: 3
seek to end...
error at 732188672 bytes, retrying: 4
resetting device ...
error at 732188672 bytes, retrying: 5
```

```
resetting controller ...
error at 732188672 bytes, retrying: 6
error at 732188672 bytes, retrying: 7
error at 732188672 bytes, retrying: 8
error at 732188672 bytes, retrying: 9
error at 732188672 bytes, retrying: 10
Trying RAW read...
Tried reading 10 times, failed doing that. Continuing...
```

которые будут тем многочисленнее, чем больше заперчен диск и чем больше вы поставили попыток чтения. Естественно, утилита не волшебная, и те данные, которые восстановить не удалось, просто заменяет пустыми блоками. Если на диске было много фотографий, потерю десятка вы переживёте менее болезненно, чем всего диска целиком. Восстановление этой программой занимает существенно больше времени, чем dd. Не ждите, что диск с обширными повреждениями (более 10% секторов) проутюжится меньше, чем за 7-8 часов (при многократном считывании с параметрами, указанными выше). Но есть проверенный способ: ставим на ночь и идём спать...

## 1.2 Глубокий анализ данных, Эпизод 1: foremost

Есть ситуации: ваша флешка начинает помирать, диск плохо читается и на нём важные данные, или вы пришли к какому-нибудь недругу и подозреваете, что у него на винчестере есть данные, которые вам нужны, а он их показывать не хочет.

Имеется класс программ "судебного анализа данных" (forensic analysis), позволяющих без шума и пыли (и ректальной имплантации горячих паяльников) выудить данные, даже если они хитро записаны.



### 1.2.1 Что есть для этого в Дебиан?

Чего только не найдёшь в Дебиановском репозитории! Например, очень и очень интересная программа `foremost`. Она позволяет искать файлы на сменных носителях / внутри образов дисков по hex-данным, характерным заголовкам и окончаниям. В Sarge версия довольно старая, но с сайта можно скачать тарболл и скомпилировать его.

### 1.2.2 Foremost в действии

Отлично, программа собрана и установлена, с требуемого носителя содран образ при помощи `dd`, теперь осталось поискать там файлы.

Попробуем поискать файлы, замаскированные под другой формат. Берём флешку, втыкаем и не монтируем - пробуем выдрать оттуда файлы типа `doc`, один из которых переименован в `jpg` (наивный юноша...):

```
$ foremost -t doc -o /opt/foremost-1.3/output/ -i /dev/sdf
```

После чего идём в подкаталог `./output` и наблюдаем радостную картину - файллик обнаружился. А вот и отчёт программы:

```
Foremost version 1.3 by Jesse Kornblum,
Kris Kendall, and Nick Mikus
Audit File

Foremost started at Sat Dec 16 21:48:07 2006
Invocation: ./foremost -t doc
-o /opt/foremost-1.3/output/ -i /dev/sdf
Output directory: /opt/foremost-1.3/output
Configuration file : /opt/foremost-1.3/foremost.conf
-----
File: /dev/sdf
Start: Sat Dec 16 21:48:07 2006
```

```
Length: 15 MB (16121856 bytes)
```

```
Num (bs=512) Size Offset
```

```
0: 129.jpg 155 KB 66048
```

```
Finish: Sat Dec 16 21:48:12 2006
```

```
1 FILES EXTRACTED
```

```
doc:= 1
```

Имя не сохранено, но содержимое в порядке. Нагретый палец и утюг можно отложить в сторону. :-)

Другой пример. Пусть хакер Нео хочет скрытно передать товарищу Морфею диск с изображением кодов к Матрице (фотографией голой Тринити). Для этого можно схитрить: призывать писать программе cddrecord не iso-образ, а просто файл:

```
$ cddrecord -v speed=0 dev=ATAPI:0,0,0 matrixcodes
```

На другом конце Морфей делает

```
$ dd if=/dev/cdrom bs=2048 of=~ /temp/matrix.jpg
```

Но вот всех застукал агент Смит, приволок в отделение и ласково спрашивает, что на болванке. Хакер Нео с ясными глазами говорит почти правду - ничего, болванка пустая (ясное дело, что "в лоб" такая болванка не читается). Агент Смит знает Линукс и поэтому он набирает в консоли:

```
$ foremost -t all -o ~/output/ -i /dev/hda
```

И выуживает из диска крамольные данные: в подкаталоге ..output/ появляется файл audit.txt следующего содержания:

```
Foremost version 1.3 by Jesse Kornblum,  
Kris Kendall, and Nick Mikus
```

```
Audit File

Foremost started at Sat Dec 16 22:15:26 2006
Invocation: ./foremost -t all
-o /opt/foremost-1.3/output/ -i /dev/hda
Output directory: /opt/foremost-1.3/output
Configuration file : /opt/foremost-1.3/foremost.conf
-----
File: /dev/hda
Start: Sat Dec 16 22:15:26 2006
Length: 604 KB (618496 bytes)

Num (bs=512) Size Offset

0: 0.jpg 88 KB 0
Finish: Sat Dec 16 22:15:28 2006

1 FILES EXTRACTED

jpg:= 1
-----

Foremost finished at Sat Dec 16 22:15:28 2006
```

и каталог `..output/jpg/` с этим файлом...

Ещё немного поигравшись с программой, можно сказать следующее. Сразу, без дополнительных танцев, находит `foremost` графические файлы `tif`, `jpg`, `png`, `bmp`, звуковые файлы `wav`, виндовые `exe`-шники, все офисные форматы (мелкоОфиса и ОпенОфиса), архивы `rar` и `zip` и многое другое. Линуксовые архивы типа `bzip2` и `r7zip` "в лоб" программа не берёт, но это дело не сильно облегчает, так как, задавшись целью, можно и их выдрать с диска.

### 1.2.3 Аналогичные программы

Такие программы особенно не афишируются, и крайне неохотно раздают за просто так. Или надо оставлять свои паспортные данные и доказывать, что вы работаете в полиции, в суде или в КГБ :-). Но всё-таки кое-что имеется. Это проприетарные Safeback, Encase, safecopy, dvdisaster и некоторые другие. Особые параноики полагают, что старый-добрый dd тоже является программой из этой же серии.

## 1.3 Глубокий анализ данных, Эпизод 2: The Sleuth Kit

Продолжая тему о программах анализа данных, в этом посте пойдёт речь о семействе утилит судебного анализа (forensic analys) The Sleuth Kit. В дистрибутив Debian оно пока не входит (в Sarge v3.1r1 во всяком случае), но это не мешает скачать сырцы с сайта проекта и собрать самостоятельно.

### 1.3.1 Установка

Для того, чтобы начать использовать The Sleuth Kit, требуется распаковать тарбол в любую директорию и просто набрать в ней make. Для сборки программы нужны библиотеки SSL, которые нужно предварительно поставить.

```
# apt-get install libssl0.9.7 libssl-dev
```

После того, как всё настроится и установится, можно приступать к сборке:

```
# make
```

В результате должно всё собраться, а утилиты появятся в подкаталоге ../bin, который до сборки был пуст. После сборки там появится много утилит, часть которых будет описываться далее.

### 1.3.2 Ищем и находим данные

После установки в вашем распоряжении окажется почти три десятка утилит, способных дать исчерпывающую информацию и том, что и как записано на носителе. Разумеется, утилиты прекрасно работают с raw-данными, полученными `dd` или `recoverdm`, о которой уже было написано. Следует отметить, что если программа `foremost` предназначена скорее для экспресс-анализа и представляет собой утилиту вида "всё в одном флаконе" то `The Sleuth Kit` это набор утилит для более глубокого исследования данных. Но это лучше показать на примере, в котором используется версия 2.07.

### 1.3.3 Пример

Пусть имеется образ флешки в файле `1.img`, и на ней есть данные, которые нужно извлечь без монтирования. Для этого сначала смотрим, какие структуры данных вообще присутствуют на диске - это делает утилита `mmls` - media management lister. Она показывает разметку диска, в том числе пустые области (unallocated spaces), а так же адреса начала и окончания партий. Поддерживаются следующие типы партий:

```
dos (DOS-based partitions [Windows, Linux, etc.])
mac (MAC partitions)
bsd (BSD Disklabels [FreeBSD, OpenBSD, NetBSD])
sun (Sun Volume Table of Contents (Solaris))
gpt (GUID Partition Table (EFI))
```

Так, применяем `mmls` для того, чтобы узнать, какое расположение и тип партий:

```
$ mmls 1.img
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors
```

Slot	Start	End	Length	Description
00:	-----	0000000000	0000000000	0000000001 Primary Table (#0)
01:	-----	0000000001	0000000031	0000000031 Unallocated
02:	00:00	0000000032	0000031359	0000031328 DOS FAT12 (0x01)
03:	-----	0000031360	0000031487	0000000128 Unallocated

Всё верно, досовская файловая система на флешке (выделение полужирным - моё). Теперь известно, откуда она начинается и где заканчивается - эта информация нужна для работы других утилит.

Отлично, теперь мы знаем тип файловой системы и где она располагается. Посмотрим, как много данных на ней есть и что мы можем оттуда выдрать - в этом деле нам поможет другая утилита, `fsstat`. Вызываем её, сообщая сведения, полученные от `mmls`:

```
penta4@penta4rce:~/temp $ fsstat -f fat -o 0000000032 1.img
FILE SYSTEM INFORMATION
-----
File System Type: FAT12

OEM Name: +/J8LIHC
Volume ID: 0x913
Volume Label (Boot Sector): SANVOL
Volume Label (Root Directory):
File System Type Label: FAT12

Sectors before file system: 32

File System Layout (in sectors)
Total Range: 0 - 31327
* Reserved: 0 - 0
** Boot Sector: 0
* FAT 0: 1 - 12
* FAT 1: 13 - 24
```

```
* Data Area: 25 — 31327
** Root Directory: 25 — 56
** Cluster Area: 57 — 31320
** Non-clustered: 31321 — 31327
```

#### METADATA INFORMATION

```
-----
Range: 2 — 500226
Root Directory: 2
```

#### CONTENT INFORMATION

```
-----
Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 2 — 3909
```

#### FAT CONTENTS (in sectors)

```
-----
57-64 (8) -> EOF
65-80 (16) -> EOF
81-88 (8) -> EOF
89-96 (8) -> EOF
97-408 (312) -> EOF
409-688 (280) -> EOF
689-696 (8) -> EOF
697-1000 (304) -> EOF
```

Отлично, теперь мы знаем, сколько файлов записано и где они расположены. Самое время посмотреть на структуру каталогов и файлов, начиная с корневого каталога. Для этого воспользуемся утилитой `fls`, которая показывает не только записанные, но и удалённые файлы. Посмотрим, что есть в корневом каталоге:

```
$ fls -f fat -o 0000000032 1.img
d/d 3: DCIM
d/d 4: SCENE
```

```
r/r * 6: raw1.bz2
r/r 8: cdpocket.pdf
r/r 10: raw1
```

Чудесно, знаем не только имена файлов, но и их смещения, которые нам потребуются, чтобы прочесть файлы. Звёздочка означает, что файл удалён: но его можно попробовать восстановить, если после удаления не проводилось интенсивного перезаписывания файлов. Если файлов много, или они в каталогах, и требуется найти смещение файла, имя которого известно, следует воспользоваться утилитой `ifind`.

```
$ ifind -a -n cdpocket.pdf -f fat -i raw -o 0000000032 1.img
8
```

Результатом является смещение файла, которое требуется для его извлечения. Всё, в наших руках вся информация о файлах - осталось их извлечь. Посмотрим, например, на файл `cdpocket.pdf`, для извлечения которого используем утилиту `icat`:

```
$ icat -f fat -i raw -o 0000000032 1.img 8 > cdpocket.pdf
```

В текущем каталоге после выполнения этой команды появится файл `cdpocket.pdf` - читается и просматривается соответствующей программой.

## 1.4 Глубокий анализ данных, Эпизод 3: повреждённые разделы

Проблема: флеш-накопитель или винчестер не монтируются, в логах сообщения о повреждении таблицы разделов - что делать? Решение: программа `Testdisk` может серьёзно помочь в деле восстановления убитых разделов в Линукс.



### 1.4.1 Ситуация

Мне нужно было перепрошить DVD-привод, о чём я уже писал ранее. После того, как всё удачно завершилось, мне нужно было перезагрузиться. Конечно, всё было выполнено в штатном режиме: `shutdown -r now` и всё шло нормально. Однако после перезагрузки мой 400Гб винчестер, который обычно висит на `/dev/sdc1`, монтироваться отказался наотрез. Я пошёл искать правды в логах `dmesg`:

```
kernel: scsi1 : ata_piix
kernel: Vendor: ATA Model: WDC WD360GD-00FL Rev: 31.0
kernel: Type: Direct-Access ANSI SCSI revision: 05
kernel: Vendor: ATA Model: WDC WD2500JD-00H Rev: 08.0
kernel: Type: Direct-Access ANSI SCSI revision: 05
kernel: Vendor: ATA Model: WDC WD4000KS-00M Rev: 07.0
kernel: Type: Direct-Access ANSI SCSI revision: 05
```

винчестер ядро видит, и даже верно определяет его объём - значит, это не шлейф отошёл, и всё гораздо хуже. А вот и проблема:

```
kernel: SCSI device sdc: 781422768 512-byte hdwr sectors
(400088 MB)
kernel: SCSI device sdc: drive cache: write back
kernel: SCSI device sdc: 781422768 512-byte hdwr sectors
(400088 MB)
kernel: SCSI device sdc: drive cache: write back
kernel: sdc: unknown partition table
```

Всё, приехали - на винчестере нет разделов! Попытка при-монтировать `/dev/sdc` приводила к:

```
EXT3-fs error (device sdc): ext3_check_descriptors: Block
bitmap
for group 880 not in group (block 0)!
EXT3-fs: group descriptors corrupted !
```

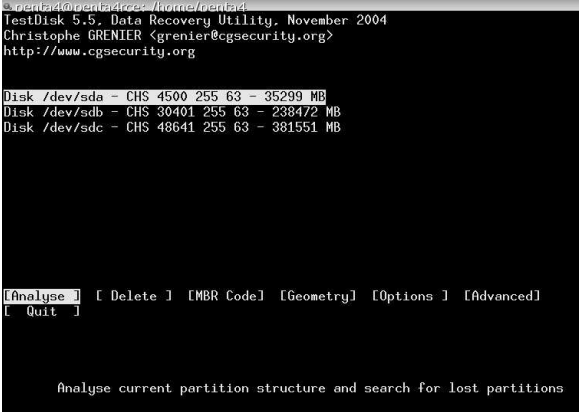
Да, не весело. Сокраментальный вопрос: что делать!?

### 1.4.2 Поиск ответа на извечный вопрос

Восстанавливаем разделы с помощью testdisk Собственно, apt-get install testdisk был проделан давно, и от рута запускаем:

```
# testdisk
```

Программа работает в интерактивном режиме, выводя все партии. Так что будьте предельно осторожны! Будут выведены все подключённые дисковые накопители:



```
gentoo@gentoo:~$ testdisk
TestDisk 5.5, Data Recovery Utility, November 2004
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

Disk /dev/sda - CHS 4500 255 63 - 35299 MB
Disk /dev/sdb - CHS 30401 255 63 - 238472 MB
Disk /dev/sdc - CHS 48641 255 63 - 381551 MB

[Analyse] [Delete] [MBR Code] [Geometry] [Options] [Advanced]
[Quit]

Analyse current partition structure and search for lost partitions
```

Так, заходим дальше, на устройство с повреждённой партицией. Программа напишет, что повреждение имеет место быть и предложит проанализировать таблицу разделов. Естественно, соглашаемся. Работать testdisk будет пропорционально объёму винчестера: будет произведён поиск резервных копий информации о структуре данных. Если вам повезёт, то копии будут найдены и будет предложено записать на диск изменения. Записываем. После этого предлагается перезагрузиться, чтобы изменения вступили в силу (забытое действие, которое реализуется shutdown -r now :-))

После этого система загрузилась и подмонтировала диск, как будто ничего и не было. Даже никаких сообщений типа "transactions replayed". Данные в полном порядке - в общем, как будто ничего и не было.

## 1.5 Отчего умирают винчестеры: небольшой обзор причин выхода из строя жёстких дисков

Он есть у каждого из нас. Он - наше всё в прямом смысле слова, маленький кусочек высоких технологий и точной механики, на котором хранятся наши бесценные данные: фотографии, тексты, фильмы, музыка, конфиги и собственно операционная система. Это - Винчестер. Но рано или поздно, он выходит из строя - почему?

Так или иначе, я старался внимательно следить за хитросплетениями мыслей авторов статей и смог-таки кое-что выцедить оттуда.

### 1.5.1 Причины, статистика, анализ

Итак, статья "Failure Trends in a Large Disk Drive Population" проливает немного света на причины выхода из строя винчестеров на серверах Гугл - авторы собирали данные в течении полутора лет (с декабря 2005 по август 2006) с почти 100.000 винчестеров, диски были SATA и ATA, 5400 и 7200 RPM, ёмкостью от 80 до 400Гб разных производителей.

Главное, на что упирают гугловцы, что SMART - вовсе не панацея от всех бед и что ориентирование только на данные SMART в большинстве случаев (особенно индивидуальных) ничего не даёт. Большая часть их дисков померли в то время, как по SMART они были совершенно здоровы и без каких либо сообщений об ошибках.

Сообщается также, что вероятность гибели винчестера слабо связана с его степенью загруженности. Но если SMART

сыплет ошибками типа scan errors, reallocation counts, offline reallocation counts, and probational counts - дело дрянь и пора делать бекапы :-)

### **Возраст**

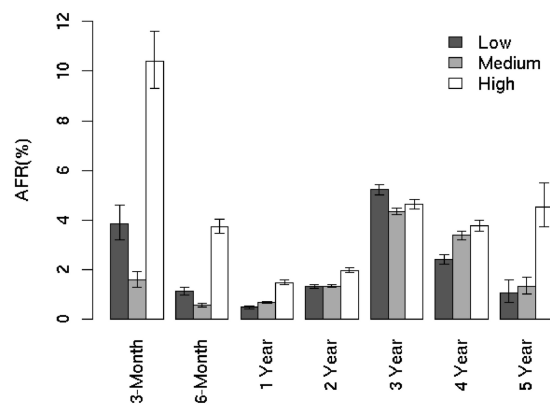
Вероятность отказа, как и следовало ожидать, растёт с возрастом диска, то есть с тем временем, которое он отработал. Диски, работающие до 1 года, чаще всего дохнут в первые три месяца. Резкий скачок вероятности выхода из строя - 2 года.

### **Производитель**

В исследовании говорится, что, вопреки расхожему мнению, выход из строя жёсткого диска слабо зависит от того, кто произвёл этот диск, а больше зависит от конкретного экземпляра и в меньшей степени от того, в каких условиях он эксплуатируется. В доказательство они приводят тот факт, что данные по сбоям, регистрируемым SMART, почти не зависят от того, диски чьих производителей анализируются.

### **Нагрузки**

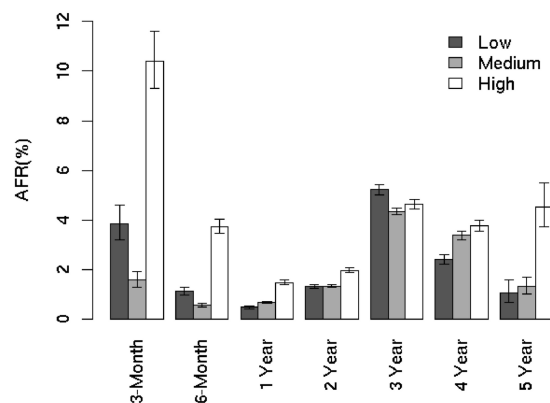
Дальше они приводят данные по зависимости смертности дисков от степени их загрузки (т. е. от дисковых операций).



Оказывается, что только очень новые (до 3 месяцев) и очень старые (старше 3 лет) чаще всего дохнут от высоких нагрузок, в остальных возрастных категориях вероятность выхода из строя от нагрузок зависит слабо.

### Температура

Считается, что температура - важнейший фактор для жёсткого диска и что лучше диски охлаждать. Здесь главное не дойти до маразма: температура винчестера ниже 15 градусов по Цельсию удваивает среднюю частоту выхода их из строя.



Гугловцы выяснили, что с повышением температуры винчестера риск отказа растёт медленно - хуже того, есть тенденция к тому, что дискам больше страшны низкие температуры. Интересно, что минимальный риск выхода из строя приходится на интервал температур от 36 до 45 градусов. Риск выхода из строя при температурах меньше 25 градусов почти вдвое больше, чем при 45, и возрастает быстро с уменьшением температуры.

Диски возраста до 2 лет чащедохнут от холода (при температуре от 15 до 30 градусов), а старики (от 3 лет) мрут от перегрева (более 45 градусов).

### 1.5.2 Анализ данных SMART

Самые важные ошибки, на которые следует обращать внимание: Scan Error, Reallocation Count Offline reallocation Probational Count

### Ошибка сканирования (Scan Error).

Электроника диска время от времени сканирует поверхность диска незаметно для пользователя и передаёт данные SMART - если будут найдены битые сектора, они, как правило, вскоре будут заменены на свободные. Однако гугловцы говорят: после первой же ошибки сканирования поверхности, вероятность выхода из строя винчестера в следующие 60 дней возрастает почти в 40 раз!

### Количество перемещений (Reallocation Count).

Если при чтении информации возникают ошибки ввода-вывода и операционная система о них сообщает, такие ошибки перехватываются SMART и сбойный сектор заменяется нормальным из набора доступных. Количество перемещений отражает износ поверхности, однако это ещё не повод бить тревогу: около 90% гугловских винчестеров имеют отличное от нуля количество перемещений, хотя при этом годовая вероятность сбоя (Annualized Fault Rate, AFR) повышается в 3-6 раз. После первого же перемещения сбойного участка, вероятность выхода из строя в следующие 60 дней увеличивается в 14 раз.

### Остальные ошибки

(в том числе Seek Error) не дают заметного вклада в общую статистическую картину дисковой смертности. Примечательно, что, например, выход диска из строя слабо соотносится с количеством циклов "старт-стоп". Однако если диску более 3 лет, следует его использовать непрерывно, так как частых включений и выключений вероятность выхода из строя повышается на 2%.

В общем, гугловцы призывают не уповать на SMART и его в общем не высокую предсказательную силу (более 56% всех умерших дисков не имели отметок об ошибках SMART), а больше налегать на бекапы и резервирование, чем почти никто не занимается, пока жареный петух не клюнет в известное

место.

На десерт - самое вкусное: распределение вероятностей ошибок по данным SMART. На кладбище гугловых винчестеров винчестеры встречаются со следующим распределением сбояв:

- Ошибки, которые SMART не отловила - 60%
- Reallocation Count - около 40%
- Seek Error - 30%
- Offline Reallocation - 28%
- Probe Count - 20%
- Scan Error - 15%
- CRC Error - менее 5%.

Ясно, что винчестеры дохнут не от одной ошибки, а чаще всего от нескольких, лидирует в которых сбойные сектора и ошибки позиционирования.

Следует отметить, что в винчестерах отдельных производителей **Raw\_Read\_Error\_Rate** и **Seek\_Error\_Rate** параметры достигают максимума и обнуляются несколько раз в день. Это связано с политикой некоторых производителей в отношении SMART: в эти параметры пишутся все ошибки, а остальные производители только те, что не смог отловить контроллер.

### Время наработки на отказ

Другая статья, "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?" подробно разбирает, что такое MTTF, или mean time to failure. Статистика также очень впечатляющая (около 100.000 устройств).

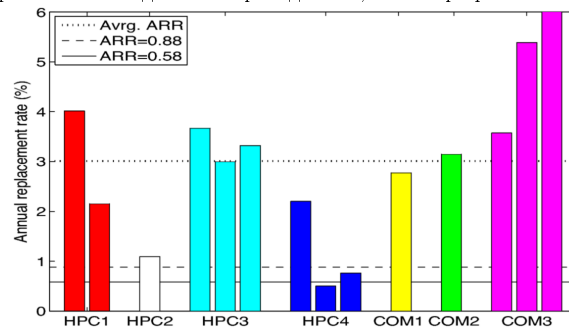
Многие производители оценивают отказоустойчивость двумя связанными друг с другом оценками: ежегодная частота ошибок (Annualized failure rate, AFR) и среднее время отказа (mean time to failure). AFR оценивается на основе предсказаний по результатам ускоренных тестов, а MTTF оценивается как время работы за год делённой на AFR. Ведущие производители дисков заявляют, что MTTF их устройств - от 1 млн.



часов до 1.5 млн. часов в соответствии с AFR равной 0.58% и 0.88% соответственно.

В статье говорится о том, что их данные о частоте замены винчестеров ввиду сбоев, мягко говоря, расходятся с тем, что заявляет производитель. Так, в трёх дата-центрах, в которых снимались данные для этой статьи в течение 5 лет, в общем случае замены жёстких дисков в связи со сбоями были несколько чаще, чем замена планок оперативной памяти, в 2.5 раза чаще, чем замена процессоров, в 2 раза чаще, чем замена материнских плат. Факт остаётся фактом: сбой винчестеров - один из самых распространённых причин остановки узлов дата-центров для замены оборудования.

Дальше в рамках исследования было вычислено значение ежегодной частоты ошибок (AFR) для всех датацентров, в которых это исследование проводилось, и вот график:

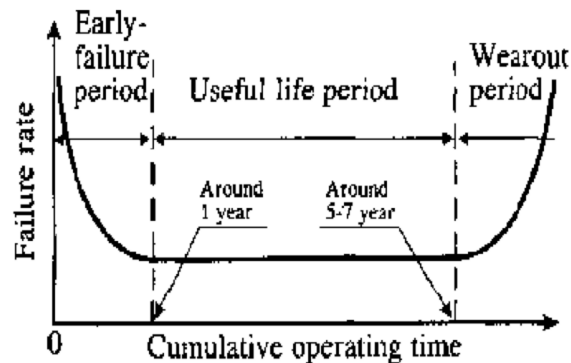


Он стоит тысячи слов: горизонтальная сплошная прямая соответствует заявляемым 1.5 млн. часам безотказной работы, горизонтальная пунктирная - 1 млн. часов, а точечная - реальному усреднённому времени работы. Согласно этому, AFR составляет 3%, а соответствующее MTTF - около 300 тыс. часов.

Разброс таких данных велик: AFR составляет от 0.5% до 13.6%, и это в дата-центрах. Последняя цифра соответству-

ет примерно 7 годам работы винчестера, но понятно, что в бытовых устройствах эта цифра намного скромнее: постоянно меняющаяся температура устройства, небольшое время непрерывной работы, скачки напряжения, большое количество циклов "старт-стоп" и прочее сильно сокращают время жизни жёстких дисков.

Ещё один замечательный график, показывающий жизненный цикл жёстких дисков в зависимости от времени работы:



Интересные выводы в работе такие:

1. MTTF, которое заявляет производители, более чем в 3 раза превосходит тот, который оценен в реальных условиях дата-центров.
2. Для старых винчестеров, отработавших 5-8 лет, переоценка MTTF производителями составляет более 30 раз.
3. Даже для сравнительно новых жёстких дисков (менее 3 лет работы) MTTF производителем завышена по крайней мере в 6 раз.
4. Частота замен для дорогостоящих SCSI-дисков и обычных SATA почти одинакова.

И далее по работе: Для дисков, чьё время непрерывной работы менее 5 лет, частота замены в связи со сбоями в 2-10 раз

больше той, которая следует из времени МТТФ, а для старше 8 лет эта частота замен в 30 раз больше.

### 1.5.3 Как просмотреть информацию SMART

Для этого уже должны быть установлен пакет `smartmontools`, который содержит в том числе утилиту `smartctl`. После этого:

- \* для IDE-дисков пишем `smartctl -all /dev/hda`
- \* для SCSI-дисков `smartctl -all /dev/sda`
- \* для SATA-дисков `smartctl -all -d ata /dev/sda`

Будет выведена длинная таблица, в которой будет много интересного.

## Глава 2

# Эмуляция

### 2.1 Создание диска IMG для QEMU

Создаём "жесткий диск" для установки Windows XP:

```
$ qemu-img create mustdie.img 4096M
```

Это создаст образ winxp.img диска 4Гб для QEMU. Вставляем диск с Windows в CD-привод и пишем:

```
$ qemu -m 512 -hda mustdie.img -cdrom /dev/cdrom -boot  
d -localtime
```

Это запустит эмулятор QEMU с 512Мб памяти, жёстким диском в виде файла mustdie.img и CD-приводом.

### 2.2 USB в эмуляторе QEMU

Запускаем QEMU с такими параметрами:

```
$ qemu -usb -m 200 -hda mustdiexp.img -boot c -localtime
```

Обязательный параметр `-usb`. Включаем флешку, и сначала запускаем `dmesg` (скажет адрес устройства), а потом `lsusb` (выдаст идентификатор устройства).

**Как узнать адрес устройства в Linux?** Спросим `dmesg`:

```
usb 1-6: new high speed USB device using ehci_hcd and
address 15
usb 1-6: configuration #1 chosen from 1 choice
scsi9 : SCSI emulation for USB Mass Storage devices
usb-storage: device found at 15
usb-storage: waiting for device to settle before scanning
scsi 9:0:0:0: Direct-Access JetFlash TS4GJFV35 8.07 PQ: 0 ANSI:
2
sd 9:0:0:0: [sde] 7987200 512-byte hardware sectors (4089 MB)
```

Полужирным я подсветил искомое - адрес нашего устройства **15**. Теперь интересуемся у `lsusb` насчёт идентификатора устройства, и он нам всё выдаст:

```
penta4@penta4rce: $ lsusb
Bus 005 Device 003: ID 046e:5500 Behavior Tech. Computer Corp.
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 015: ID 058f:6387 Alcor Micro Corp.
Bus 001 Device 014: ID 04fc:0c15 Sunplus Technology Co., Ltd
Bus 001 Device 001: ID 0000:0000
```

Идентификатор устройства **15** есть **058f:6387**. В QEMU заходим в гостевую систему, нажимаем `CTRL+ALT+2` и попадаем в консольный режим QEMU. Там мы пишем:

```
$ usb_add host:058f:6387
```

Если QEMU пишет, что не может добавить ваше устройство, а вы уверены, что всё сделано правильно - проверьте права на файл-устройство, оно должно быть с полными (777) правами. В консоли, откуда запущен QEMU, увидите:

```
/proc/bus/usb/001/015: Permission denied
```

Стало быть, делаем так:

```
# cd /proc/bus/usb
# chmod -R 777 001/
```

После этого устройство готово к работе.

## 2.3 Монтирование диска QEMU

Как получить доступ к файлам внутри "жёсткого диска" QEMU? Создаём точку монтирования, даём ей нужные права и прописываем в конфиге `/etc/fstab`:

```
/opt/qemu/images/win2k.img /mnt/windows vfat
rw,user,loop,noauto,offset=32256 0 0
```

Теперь можно монтировать и отмонтировать образ диска.

## 2.4 Послать в QEMU сочетание клавиш

Для переключения виртуальных консолей в гостевой Linux-системе следует переключиться в монитор и использовать команды вида:

```
sendkey ctrl-alt-f2
```

Регистр важен: например, "sendkey ctrl-alt-delete" работает, а вот "sendkey CTRL-ALT-DELETE" не будет работать. Хотя в документации говорят, что должно работать оба варианта.

## 2.5 Перегон образов QEMU в VirtualBox

Всё для преобразования уже в комплекте VirtualBox.

```
$ qemu-img convert -O raw test.img test.raw
$ VBoxManage convertdd test.raw test.vdi
$ VBoxManage modifyvdi /full/path/to/test.vdi compact
```

Обратная конвертация из VirtualBox в QEMU с помощью `vditool` для Linux, последовательность действий такова:

```
$ vditool COPYDD vbox-image.vdi raw-image.img
$ qemu-img convert -f raw raw-image.img -O qcow2
  qemu-image.qcow
```

В отличие от образов QEMU, примонтировать которые не составляет проблемы, образы VirtualBox примонтировать вовсе не так просто.

## Глава 3

# Системное

### 3.1 Монтирование разделов в Linux

В переводе на язык UNIX, это звучит так:

```
# mount -t vfat /dev/hda3 /mnt/storage
```

Если нужно часто монтировать одни и те же разделы, занесите их в `/etc/fstab` - это общесистемный конфиг

Файл `/etc/fstab` состоит из колонок, разделителями между которыми являются символы табуляции (клавиша TAB). Вот пример содержимого файла `/etc/fstab`:

```
# /etc/fstab: static file system information.
/dev/hda2 / reiserfs notail,noatime 0 1
/dev/hdb /mnt/cdrom iso9660 ro,user,noauto 0 0
/dev/sda1 /mnt/flash vfat
        iocharset=koi8-r,codepage=866,rw,user,auto 0 0
```

Формат колонок такой:



- Монтируемое устройство (файл-устройство, он же раздел, который вы хотите примонтировать).
- Точка монтирования (в какую директорию в файловой системе монтировать).
- Тип монтируемой файловой системы.
- Опции монтирования (кодировка языка, необходимость монтирования при старте).
- Dump (обычно 0).
- Pass (обычно 0).

Важно, чтобы вы вставляли Tab после каждой колонки при редактировании /etc/fstab

### 3.1.1 Кодировки кириллицы при монтировании

Временное, однократное монтирование, например, флешки, выглядит в командной строке так:

```
# mount -t vfat /dev/XX /mnt/YY -o
iocharset=utf8,codepage=866
```

Пример: монтирование типичной флешки выглядит так

```
# sudo mount -t vfat /dev/sda1 /mnt/flash -o
iocharset=utf8,codepage=866
```

То есть монтируем раздел /dev/sda1 типа vfat в точку монтирования /mnt/flash

### 3.1.2 Монтирование NTFS

Временное, однократное монтирование, например, раздела с данными Windows, выглядит так:

```
# mount -t ntfs-3g /dev/XX /mnt/YY -o umask=0,nls=utf8
```

То есть монтируем раздел /dev/sda1 типа ntfs-3g в точку монтирования /mnt/flash

## 3.2 Файл подкачки: swap-файл и swap-раздел в Linux

Прежде, чем что-то создавать, хорошо бы узнать, сколько swap-пространства у нас уже имеется в системе - для этого следует дать команду в консоли от рута:

```
# swapon -s
```

Результат будет в виде:

Filename	Type	Size	Used	Priority
/dev/hda1	partition	289128	0	-1

Итак, описание вывода команды:

- Filename описывает имеющиеся у вас своп-пространства и где они находятся.
- Type указывает тип пространства: partition (раздел) или file (файл).
- Size сообщает общий размер Swap-пространств.
- Used говорит о том, сколько сейчас свопа задействовано.
- Priority указан приоритет, т.е. какие пространства системе использовать вначале.

### 3.2.1 Создание swap-файла в Linux

1. Открываем консоль/терминал и получаем полномочия root или используем sudo:

```
$ su
```

2. Даём команду:

```
$ sudo dd if=/dev/zero of=/swapfile bs=1M count=500
```

В команде `dd` для задания размеров можно использовать суффиксы К, М, Г для килобайт, мегабайт и гигабайт соответственно. В данном примере это 500 Мегабайт файла подкачки.

3. Поясняем системе, что созданный пустой файл это всё-таки файл подкачки для Linux:

```
$ sudo mkswap /swapfile
```

4. Подключаем созданный `swap`-файл:

```
$ sudo swapon /swapfile
```

При этом в выводе команды `top` или команды `free` должно появиться упоминание, что свопинга в системе поприбавилось. Чтобы отключить файл подкачки, пишем

```
$ sudo swapoff /swapfile
```

Чтобы не подключать `swap`-файл или `swap`-раздел каждый раз, полезно занести запись в `/etc/fstab` следующего содержания:

```
/swapfile none swap sw 0 0
```

На всякий случай отмечу, что каждый раз создавать `swap`-файл не нужно: просто подключаете и отключаете его с помощью `swapon/swapoff`. Работа со `swap`-разделами в Linux происходит аналогичным образом.

### 3.2.2 Приоритет SWAP-файлов

Создавать и использовать `swap`-файлов в Linux можно любое количество. При этом можно указать приоритет подключаемого `swap`-файла или раздела (хотя ядро умеет самостоятельно распределять по разделам/файлам подкачки).

Например, высший приоритет для файла подкачки задаётся так:

```
# swapon -p 1 /opt/swapfile
```

Приоритет является целым числом от 0 до 32767.

### Очистка swap-пространства после ресурсоёмких приложений

Командой `swaponoff` -а, запущенной от имени `root`, можно отключить использование всех разделов и файлов подкачки. После ввода команды содержимое свопа за несколько минут загружается обратно в оперативную память, а сам раздел подкачки отключается. После загрузки содержимого свопа в оперативную память включем своп обратно командой `swapon` -а.

### Системные настройки использования своппинга - Linux

За интенсивность обращения системы к swap-файлам и swap-разделам отвечает параметр `swappiness`, равный по умолчанию 60. Значение параметра может быть в пределах от 0 - наименьшее использование подкачки, до 100 - подкачка используется часто.

Насчёт оптимального значения параметра `swappiness` есть много разных мнений. Так, например, один из ведущих разработчиков ядра Эндрю Мортон считает, что для десктопа лучше ставить большое значение, чтобы всякое блондваре скинуть в своп и использовать оперативную память для чего-то нужного.

Временно (до перезагрузки системы) изменить этот параметр можно с помощью команды:

```
# echo 50 > /proc/sys/vm/swappiness
```

Чтобы изменить значение по умолчанию, необходимо изменить параметр `vm.swappiness`:

```
vm.swappiness=50
```

в файле `/etc/sysctl.conf`

Следует, впрочем, отметить, что со `vm.swappiness` сильно перегибать палку не стоит. При больших значениях система теряет в отзывчивости (будет вытеснять память, с которой работают приложения, в своп, хотя оперативной памяти ещё много). При малых значениях система работает отзывчивей, но когда оперативная память заканчивается, система начинает активно свопиться и притормаживать. Также можно попробовать увеличить/уменьшить объём потребляемой системой памяти за счёт изменения размеров дискового кеша. Уровень выделяемой под кеш памяти хранится в

```
/proc/sys/vm/vfs_cache_pressure
```

Значение по умолчанию: 100. Чтобы использовать меньше памяти под дисковые кеши (что вообще говоря не есть хорошая идея), ставим значение 50. Если, наоборот, хочется больше отзывчивости системы, увеличиваем размер кеша не скупясь:

```
# echo 1000 > /proc/sys/vm/vfs_cache_pressure
```

и далее продолжаем играть с параметрами вплоть до полного удовлетворения. Для того, чтобы настройки стали постоянными, заносим параметр

```
vm.vfs_cache_pressure = 1000
```

в файл `/etc/sysctl.conf`

### 3.3 Как создать ramdisk в Linux?

Бывают задачи, когда нужно работать с большим количеством мелких файлов и делать это по возможности быстро, без завязок на меееедленную файловую подсистему.

Собственно, в ядре должна быть поддержка `tmpfs`, и если вы (или авторы вашего дистрибутива) не любители усложнять себе жизнь, она там должна быть. Быстренько создаём новую точку монтирования:

```
# mkdir /mnt/tmpfs/  
# chmod 777 /mnt/tmpfs/
```

и даём все права на использование. После этого даём команду (от рута или через sudo):

```
# mount -t tmpfs -o size=xxxM tmpfs /mnt/tmpfs/или  
$ sudo mount -t tmpfs -o size=xxxM tmpfs /mnt/tmpfs/
```

где xxxM есть размер RAMdisk в Мегабайтах. Будьте осторожны и не размахивайтесь слишком широко: если вы попросите создать ramdisk больше, чем имеете оперативной памяти, система начнёт сгружать всё в swap-файл и дела пойдут очень медленно и грустно.

Если же вы хотите изменить размер ramdisk - это просто:

```
# mount -o remount -o size=yyyM /mnt/tmpfs/или  
$ sudo mount -o remount -o size=yyyM /mnt/tmpfs/
```

При этом содержимое диска не пропадёт, но его размер тут же увеличится (при уменьшении также ничего не пропадёт, если только данных не больше, чем диск).

### 3.4 Два слова о /proc

Используя файловую систему /proc, можно изменять многие параметры системы без необходимости перезагрузки машины и многое узнать о работе системы. О некоторых полезных командах - далее в этом посте.

Файловая система /proc - это виртуальная файловая система[1], поэтому файлы, которые вы найдете в этом каталоге, на самом деле не занимают места на вашем жестком диске. Это карта, создаваемая ядром и присоединяемая к вашей обычной файловой системе, чтобы обеспечить доступ к настройкам ядра и информации о системе.

### 3.4.1 Общие слова

Не стоит открывать эти файлы обычным текстовым редактором - это может нарушить целостность данных или ядро может успеть изменить значения в этих файлах. Чтобы этого избежать, следует пользоваться командами `echo` и `cat` для установки и просмотра содержимого файлов соответственно.

Чтобы передать значение `fake` в файл `/proc/your/file`, следует дать команду:

```
# echo "fake" > /proc/your/file
```

Для того, чтобы увидеть значение в файле, используем `cat`:

```
# cat /proc/your/file
```

получим:  
`fake`

В основном в `/proc` вы найдете файлы `read-only` за исключением `/proc/sys`, которая содержит большинство параметров ядра и предназначена для изменения во время работы системы.

### 3.4.2 Примеры использования `/proc`

В каталоге `/proc` множество файлов, большинство из них - только для чтения, из них можно узнать многое об оборудовании. Но есть файлы и на запись - они позволяют менять на лету параметры работы ядра. Вот некоторые из них:

#### `/proc/scsi/scsi`

Позволяет добавить или отключить диск, если он поддерживает горячую замену (SCSI или SATA). Указать ядру на подключение нового диска можно командой:

```
# echo "scsi add-single-device w x y z" > /proc/scsi/scsi
```

Чтобы эта команда работала правильно, вы должны указать параметры значений *w*, *x*, *y*, и *z* следующим образом:

- *w* — это ID хост адаптера, где первый адаптер имеет ID ноль (0))
- *x* — это канал SCSI на хост адаптере, где первый канал ноль (0)
- *y* — это SCSI ID устройства
- *z* — это номер LUN, где первый LUN ноль (0)

Этот трюк можно проворачивать не только с дисками: в своё время это пришлось делать мне для подключения много-слотового кардридера.

Чтобы извлечь диск из системы без перезагрузки, размонтируем его и даём команду:

```
# echo "scsi remove-device w x y z" > /proc/scsi/scsi
```

Перед тем как ввести эту команду и удалить SCSI диск, убедитесь, что вы отмонтировали файловые системы на этом диске.

#### **/proc/sys/kernel/acct**

Здесь содержатся три конфигурируемых значения, которые управляют подсчётом процессов, основанном на свободном пространстве файловой системы:

1. Если свободное пространство ниже значения в процентах, то процесс подсчета останавливается.
2. Если свободное пространство выше, то процесс запускается.
3. Частота в секундах, с которой проверяются предыдущие два значения.

Чтобы изменить значения в этом файле, вам следует использовать разделенный список параметров. Значения по умолчанию: 2 4 30

Для их изменения следует дать команду



```
# echo "3 5 40" > /proc/sys/kernel/acct
```

Эти значения остановят подсчёт, если в файловой системе менее 3 процентов свободного пространства и начнет опять если появится 5 или более процентов. Проверка производится каждые 40 секунд.

#### **/proc/sys/kernel/ctrl-alt-del**

Этот файл содержит двоичное значение, которое управляет реакцией системы на комбинацию ctrl+alt+delete. Возможны два значения:

1. Ноль (0) значит, что ctrl+alt+delete принимается и отправляется программе init, что обеспечит правильный останов и перезагрузку как если бы вы ввели команду reboot.
2. Один (1) значит, что ctrl+alt+delete не принимается и никакого чистого отключения не происходит - перезагрузка как при нажатии на RESET.

Значение по умолчанию — 0.

#### **/proc/sys/kernel/panic**

Задаёт время в секундах, которое ядро будет ждать перед перезагрузкой если произойдет "kernel panic". Установка в ноль (0) секунд отключит возможность перезагрузки при kernel panic. Значение по умолчанию - 0.

#### **/proc/sys/kernel/sysrq**

Включает или выключает возможность использования Волшебной SysRQ-клавишей (Magick System Request Key). Если в ядре она активирована, чаще всего значение в этом файле 1. Об этой клавише и её возможностях было написано тут.

### `/proc/sys/vm/swappiness`

Этот параметр позволяет регулировать стремление системы сбрасывать данные из памяти в файл подкачки. Чрезмерное значение здесь приведёт к интенсивному использованию swap-файла, что нежелательно в ряде случаев (ноутбуки, lowlatency-системы). Слишком маленькое значение может привести к тому, что при заполнении памяти будет принудительно запущен OOMkiller (процесс, запускающийся при исчерпании памяти и убивающий наиболее ресурсоёмкие задачи).

Значение параметра может быть в пределах от 0 (наименьшее использование подкачки) до 100 (подкачка используется часто). Значение по умолчанию - 60. Есть сообщения, что при большом объёме памяти (от 1Гб) оптимальным значением является 10. Для этого:

```
# echo "10" > /proc/sys/vm/swappiness
```

### 3.4.3 Как сделать настройки в `/proc` постоянными

Для того, чтобы изменения в директории `/proc/sys` стали постоянными, нужно внести изменения в файл конфигурации для `sysctl` - `/etc/sysctl.conf`

Формат этого файла требует некоторых пояснений. Так как `sysctl` может только изменять переменные в директории `/proc/sys`, то часть имени переменной обозначающая директорию отбрасывается. Другое изменение касается слэшей, которые заменяются на точки. Вот два простых правила для преобразования файлов в `/proc/sys` и переменных в `sysctl`:

1. Уберите из начала команды `/proc/sys`;
2. Замените слэши на точки в имени файла.

Эти два правила позволят вам преобразовать любой файл в `/proc/sys` в любое имя переменной в `sysctl`. Обычное преобразование имени файла в переменную:

```
/proc/sys/dir/file --> dir.file  
dir1.dir2.file --> /proc/sys/dir1/dir2/file
```

Вы можете увидеть все переменные, доступные для изменения, используя команду `sysctl -a`.

Хотя ниже следующее замечание относится к другой файловой системе, `/sys`, я считаю полезным привести здесь один пример. Это смена планировщиков ввода-вывода на лету. Чтобы узнать, какой планировщик стоит сейчас, даём команду:

```
$ cat /sys/block/sdX/queue/scheduler
```

Заменяя `sdX` на интересующее нас устройство. Текущий планировщик отмечен в квадратных скобках:

```
noop anticipatory deadline [cfq]
```

То есть сейчас стоит `cfq` - `complete fair queue`, полностью честный планировщик. Планировщики для разных дисков могут быть разными. Значения, которые можно передать в этот файл, такие: `"as"` `"cfq"` `"deadline"` или `"noop"`. Об этом в других постах.

### 3.5 cfdisk и mkfs: разметка и форматирование разделов в Linux

После приобретения нового винчестера или флешки нужно разбить свежее пространство на разделы и отформатировать их под требуемую файловую систему Linux. Это очень просто только почему-то об этом не пишут.

#### 3.5.1 Создание раздела на жёстком диске в Linux

Для этого используем. Для создания разделов потребуются привилегии `root`, добываемые командой `su`.

После этого пишем:

```
# cfdisk /dev/hdb
```

Нажимаем на New, создать раздел. Выбираем Primary (основной) Создастся, нажимаем Bootable (если нужно, чтобы он был загрузаемый). Если уверены, что всё сделано правильно - записываем изменения, нажимаем Write и пишем yes

Всё, разделы созданы, выходим из программы, выбирая Quit.

#### Если на диске "битый"раздел

К примеру, таблица разделов на флешке повреждена настолько, что проще всё снести и создать заново - тогда делаем так:

```
# cfdisk -z /dev/hdb
```

Параметр -z создаёт нулевую таблицу разделов, и позволяет сразу начать формировать новую.

### 3.5.2 Форматирование дисков в Linux

Для этого сначала нужно выбрать файловую систему.

Далее, по умолчанию в Debian поставлены утилиты ext3. Если нужных нет - ставим:

```
# aptitude install reiserfsprogs xfsprogs jfsutils
```

После этого пишем:

```
$ sudo mkfs -t типФС /файл/устройства
```

Вместо /файл/устройства пишем имя формируемого раздела, например /dev/hdb1. В нашем примере это выглядит так:

```
$ sudo mkfs -t ext3 /dev/hdb1
```

После форматирования раздел сразу же можно смонтировать и использовать.

### 3.5.3 Настройка файловой системы

**ВНИМАНИЕ! Всё нижеописанное делается на ОТМОНТИРОВАННОЙ файловой системе!** Здесь нужно ещё отметить тот факт, что при форматировании в ext3, для пользователя root резервируется 5% места. Чтобы убрать 5 процентное резервирование места для root в ext3 нужно дать команду:

```
# tune2fs -m 0 /dev/hdb1
```

После этого простому пользователю будет доступно всё место.

Так же имеет место быть принудительная проверка диска после некоторого количества загрузок. Если вам это не нужно, процедуру можно отменить командой:

```
# tune2fs -c 0 /dev/sda1
```

Либо сделать так, чтобы проверка происходила реже, например, 1 раз в 100 загрузок:

```
# tune2fs -c 100 /dev/sda1
```

Просмотреть свойства файловой системы можно командой

```
# tune2fs -l /dev/sda1
```

и узнать о ней много интересного.

### 3.5.4 Информация о разделах винчестера

Иногда требуется узнать, какие разделы на винчестере есть и какая там файловая система. Для того, чтобы узнать информацию о разделах жесткого диска, набираем от root пользователя:

```
$ fdisk -l /dev/sda
```

в ответ получаем:

```
Disk /dev/sda: 160.0 GB, 160041885696 bytes
255 heads, 63 sectors/track, 19457 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier : 0xd0f4738c

Device Boot Start End Blocks Id System
/dev/sda1 1 61 489951 82 Linux swap / Solaris
/dev/sda2 * 15567 19457 31254457+ 83 Linux
/dev/sda3 62 15566 124543912+ 83 Linux
```

Откуда видим: есть три раздела, первый из них (/dev/sda1) это раздел подкачки (500 мегабайт). Второй раздел (/dev/sda2) на 30Гб, третий (/dev/sda3) на 120Гб. Активный раздел (который сейчас используется операционной системой) помечен звёздочкой.

Узнать, на каком разделе вы сейчас работаете, можно с помощью команды:

```
df .
```

точка на конце обязательна

### 3.6 Дата и время: date в Linux

Время от времени приходится поправлять часы или узнавать в консоли точное время - для этого, как известно, есть команда date. Но у неё есть множество других замечательных возможностей, о которых владельцы UNIX-систем могут и не догадываться. Итак...

Просто просмотреть время можно командой:

```
$ date
```

```
Чтв Авг 21 14:07:23 MSD 2008
```

Чтобы изменить время часов, нужно заделаться рутом и написать:

```
# date MMDDhhmm
```

То есть в виде МесяцДеньЧасМинута, остальное `[MMDDhhmm[[CC]YY][.ss]]` по вкусу. Об это хорошо (и главное в самом верху) написано в мануале `man date`, однако остальное писалось, по всей видимости, роботами для роботов.

Среди функций `date` есть также вычисление дат. Например, вы хотите узнать, какое число было вчера:

```
$ date --date="yesterday" +F
```

или

```
$ date --date="1 days ago" +F2008-08-13
```

Или, скажем, какое число будет в воскресенье через две недели:

```
$ date --date="sunday next 2 week"
```

```
Sun Aug 31 01:00:00 EEST 2008
```

Если специально не оговаривать срок, то названия дней недели командой `date` будут восприниматься как ближайшие. Например, узнать дату ближайшей пятницы:

```
$ date --date="friday" +F2008-08-15
```

Тот же результат вы получите, если добавите ко дню недели слово `this` то есть `date --date="this friday" +%F`

Прошедшие дни недели указываются при помощи ключевого слова `last` например, какое число было в прошлую пятницу:

```
$ date --date="last friday" +F2008-08-08
```

Думаю, что это хороший повод ещё разок за рюмкой чая почитать man date здесь (на русском).

А ещё можно смотреть не просто на даты, а на календарик в консоли - для этого есть замечательная команда cal - без параметров она выдаст календарь на текущий месяц:

```
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

Просто и со вкусом. Если же вам нужен календарь на год:

```
$ cal 2009
```

даст искомое. Календарь на месяц можно приготовить так: cal 02 2009 выдаст календарь на февраль 2009 года.

Помимо cal существует также ncal - Если же нужен “вертикальный” календарик, вроде

```
Февраль 2010
Пн 1 8 15 22
Вт 2 9 16 23
Ср 3 10 17 24
Чт 4 11 18 25
Пт 5 12 19 26
Сб 6 13 20 27
Вс 7 14 21 28
```



это можно сделать с помощью команды `psal`. Кстати, этот вывод очень хорошо вставляется в Inkscape и далее можно быстро набросать простой календарик для ежедневника.

## Глава 4

# Железо

### 4.1 Как создать ramdisk в Linux

В ядре должна быть поддержка tmpfs. Быстренько создаём новую точку монтирования:

```
# mkdir /mnt/tmpfs/  
# chmod 777 /mnt/tmpfs/
```

и даём все права на использование. После этого даём команду (от рута или через sudo):

```
# mount -t tmpfs -o size=xxxM tmpfs /mnt/tmpfs/
```

где **xxxM** есть размер RAMdisk в Мерабайтах.

Если же вы хотите изменить размер ramdisk - это просто:

```
# mount -o remount -o size=yууM /mnt/tmpfs/
```

При этом содержимое диска не пропадёт, но его размер тут же увеличится (при уменьшении также ничего не пропадёт, если только данных не больше, чем диск).

## 4.2 Запись CD и DVD дисков в Linux

Приводятся только основные команды и параметры утилит.

### 4.2.1 Как узнать адрес привода

Запись на привод проводится через ATAPI, а не эмуляцию SCSI, как когда-то делали. Для этого пишем в консолюке:

```
$ sudo cdbrecord -scanbus dev=ATAPI
```

Выдаст он нам что-то вроде:

```
Cdbrecord-Clone 2.01.01a01 (i686-pc-linux-gnu) Copyright
(C) 1995-2004 Joerg Schilling
'@(#)scsitransp.c 1.91 04/06/17 Copyright
1988,1995,2000-2004 J. Schilling').
scsibus0:
0,0,0 0) ' _NEC ' 'DVD _RW ND-3550A ' '1.05' Removable
CD-ROM
0,1,0 1) *
0,2,0 2) *
0,3,0 3) *
0,4,0 4) *
0,5,0 5) *
0,6,0 6) *
0,7,0 7) *
```

Выделенная полужирным строка - это и есть наш привод, а цифры 0,0,0 - его адрес, который и подставляем в качестве цели для записи диска через ATAPI. Но об этом ниже.

### 4.2.2 Запись CD-R/RW дисков

Для этого нужно установить либо сторонние пакеты cdrtools, либо пакеты из Etch, которые называются cdrkit (свободная

версия cdrtools). Тогда в вашем распоряжении окажутся утилиты `cdrecord` для записи/очистки дисков, `mkisofs` для создания ISO-образов из файлов, `cdda2wav` для сдирания с образа аудиотреков.

### Создание ISO-образа

Если нужно создать сперва ISO-образ для пропаливания, пользуемся утилитой `mkisofs`. Если нужны длинные имена файлов и русские их названия, то включаем расширение `Joliet`. Короче говоря, чтобы из каталога `/home/vasya/stuff` сделать iso-образ под названием `disk.iso`, пишем в консолю следующее:

```
$ mkisofs -joliet -long -r -o disk.iso /home/vasya/stuff
```

После этого в каталоге, из которого вызывалась программа, появится `disk.iso`, который теперь можно пропаливать на болванку.

### 4.2.3 Запись CD-R дисков

Если уже есть ISO-образ и его надо просто пропалить, то надо вызвать `cdrecord` и указать ему скорость пропаливания, адрес устройства и непосредственно сам образ. Делается это так:

```
$ cdrecord -v speed=40 dev=ATAPI:0,0,0 my.iso
```

То есть записать с подробным выводом сообщений на скорости 40 на устройство `ATAPI:0,0,0`, (Primary Master 1-ого IDE-канала) образ `my.iso` - всё логично и понятно. Пока пропаливается диск - сворачиваем окошко консоли и занимаемся другими делами.

### 4.2.4 Запись CD-RW дисков в Linux

Для того, чтобы очистить болванку, есть несколько методов, реализованных в `cdrecord`: быстрая очистка, только указан-

ная сессия, полная очистка и другие. Например, для быстрой очистки пишем:

```
$ cdrecord -v blank=fast dev=ATAPI:0,0,0
```

и через пару секунд на него уже можно писать новые данные. Запись CD-RW дисков ничем не отличается от записи обычных однократных болванок, и команда:

```
$ cdrecord -v speed=40 dev=ATAPI:0,0,0 my.iso
```

пропаливает диск точно так же. Только следите за указанием скорости пропаливания - для перезаписываемых болванок она ниже, чем для однократных.

#### 4.2.5 Запись DVD-R/RW дисков

Для этого нужно поставить утилиты dvd+rw-tools, после чего в нашем распоряжении будет growisofs для создания и записи образов на DVD, а так же dvd+rw-format для предварительного форматирования перезаписываемых DVD+RW.

##### Запись DVD-/+R дисков в Linux

Для этого нам потребуется утилита growisofs, которая входит в комплект dvd+rw-tools, и объединяющая в себе mkisofs и cdrecord в одном флаконе. Для того, чтобы пропалить на DVD-болванку (без разницы - это DVD+R или DVD-R) файлы из каталога /home/vasya/something, пишем в консоли:

```
$ growisofs -Z /dev/cdrom -R -J /home/vasya/something
```

где после -Z идёт устройство, которое является резаком. Даже если оно называется /dev/cdrom, не надо удивляться: утилите всё равно. Если уж так хочется прожечь болванку побыстрее, можно рекомендовать growisofs скорость ключом - speed=N так:

```
$ growisofs -speed=8 -Z /dev/cdrom -R -J  
/home/vasya/something
```

Помним, что для DVD скорость  $1\times = 1385$  Кб/сек. Если нужно пропалить уже готовый ISO-образ, команда изменится так:

```
$ growisofs -Z /dev/cdrom=image.iso
```

В процессе записи будет постоянно выводиться информация о текущей скорости записи, прогрессе и оставшемся до конца пропаливания времени.

#### 4.2.6 Запись DVD-/+RW дисков

Для начала болванку нужно отформатировать, это делается так:

```
# dvd+rw-format -f /dev/cdrom
```

Если у вас другое устройство для записи, отличное от `/dev/cdrom`, замените его на своё. После этого запись на диск идёт так же, как и для обычных DVD-болванок:

```
# growisofs -Z /dev/cdrom=image.iso
```

После этого начнётся запись, а мы можем заняться чем-то более полезным, чем созерцание процентов записываемого образа.

#### Возможность использования DVD-/+RW как дискеты

Есть возможность использования пакетной записи и файловой системы `udf` (должна быть скомпилирована поддержка ядра), чтобы работать с перезаписываемым DVD-диском как с большой дискетой. Как мне любезно написал в комментариях ув.тов. Apopuious, выглядит это так:

```
# dvd+rw-format -f /dev/cdrom
# mkudffs /dev/cdrom
```

После чего работаем как с простым RW сменным носителем типа дискетки или флешки:

```
# mount -t udf /dev/cdrom /mnt/cdrom
# cp /etc/* /mnt/cdrom/
# rm /mnt/cdrom/shadow
# eject
```

### 4.3 Проверка правильности записи на CD и DVD диск

Утилита `dd` входит в состав любого дистрибутива и является чрезвычайно полезной во многих случаях.

#### 4.3.1 Для CD-дисков

Вставляем пропаленный диск, и, не монтируя его, пишем в консольке:

```
# dd if=/dev/cdrom | md5sum
```

После этого видим:

```
# dd if=/dev/cdrom bs=2048 | md5sum
355562+0 входных записей
355562+0 выходных записей

728190976 bytes transferred in
299,413377 seconds (2432059 bytes/sec)

e941bb0ecac20a2043c9f8e69fbab63a —
```

### 4.3.2 Для DVD-дисков

При записи на DVD пишется так же и служебная информация, поэтому для проверки DVD-дисков утилитой `dd` нужно точно указывать размер и количество блоков. Для того, чтобы узнать, сколько значимых блоков (extents) имеется в iso-образе, выполняем:

```
$ echo $(( $(ls -l my.iso | awk '{ print $5 }') / 2048 ))
```

Результатом будет число блоков:

```
169383
```

Теперь, зная, что размер блока равен 2048 байт и количество блоков на DVD-диске, даём команду:

```
$ dd if=/dev/cdrom bs=2048 count=169383 | md5sum
```

Результатом её выполнения будет вывод:

```
$ dd if=/dev/cdrom bs=2048 count=169383 | md5sum
169383+0 records in
169383+0 records out
e941bb0ecac20a2043c9f8e69fbab63a —
```

Как видно, что записанный на DVD образ и isoшник на жёстком диске совпадают, так что запись прошла успешно.

## 4.4 Ошибки при записи DVD и как с ними бороться

Случается, что при записи DVD-дисков возникают ошибки. Обычно при этом болванка выбрасывается, однако есть способы некоторые ошибки исправить и записать-таки диск.



#### 4.4.1 Сбой в процессе записи в начале или середине диска

В начале или середине диска запись обрывается и growisofs выдаёт сообщение

```
Executing 'builtin_dd if=1.iso of=/dev/cdrom obs=32k
seek=0'
/dev/cdrom: "Current Write Speed" is 16.4x1352KBps.
0/4393730048 ( 0.0) @0x, remaining ??? RBU
100.0 UBU 0.00/4393730048 (
0.0) @0x, remaining ??? RBU 100.0 UBU 0.0
0/4393730048 ( 0.0) @0x, remaining ??? RBU
100.0 UBU 0.00/4393730048 (
0.0) @0x, remaining ??? RBU 100.0 UBU 0.0
0/4393730048 ( 0.0) @0x, remaining ??? RBU
100.0 UBU 0.01671168/4393730048 ( 0.0) @0.4x,
remaining 1007:27 RBU 100.0 UBU 1.9
:-[ WRITE@LBA=330h failed with
SK=3h/ASC=0Ch/ACQ=00h]:
Input/output error
:-( write failed : Input/output error
/dev/cdrom: flushing cache
/dev/cdrom: closing track
:-[ CLOSE TRACK failed with SK=3h/ASC=0Ch/ACQ=00h]:
Input/output error
/dev/cdrom: closing session
/dev/cdrom: reloading tray
```

Болванку выбрасывать не надо, а только запустить процесс ещё раз - диск будет продолжен с того места, где запись прервалась. После этого диск нормально читается и проходит проверку на MD5-суммы.

#### 4.4.2 Ошибка закрытия сессии

Диск полностью записался, и начинается закрытие сессии - иногда здесь происходит ошибка:

```
2144544*2KB out @ average 9.9x1352KBps
/dev/cdrom: flushing cache
/dev/cdrom: closing track
/dev/cdrom: closing session
:-[ CLOSE SESSION failed with
    SK=5h/ASC=72h/ACQ=03h]:
Input/output error
```

Обидно: диск-то записался, но читаться он не будет, так как не закрыта сессия.

Для этого есть фирменная "затычка" от авторов dvd+rw-tools: небольшая программа close, и нужен её исходник. Его нужно скачать отсюда, установить/скачать исходные коды dvd+rw-tools и скопировать close.cpp в каталог с исходниками dvd+rw-tools. После этого в каталоге с исходниками компилируем close.cpp командой:

```
$ g++ -o close close.cpp
```

Всё, даём ей права на исполнение, вставляем в привод диск с неудачно закрытой сессией и пишем:

```
# close /dev/hda
```

Если /dev/hda ваш пишущий DVD-привод, конечно. После этого диск будет закрыт, его можно прочесть и он проходит проверку MD5-сумм.

Так же для закрытия сессии можно использовать команду

```
# dvd+rw-format -lead-out /dev/scdN
```

Должно работать и для обычных DVD-R дисков.

## 4.5 Как узнать подробности оборудования в Linux?

Часто, особенно при настройке нового оборудования в Linux или решения проблем со старым, требуется знать подробности об устройстве: фирма-изготовитель, марка и другие опознавательные знаки устройства.

### 4.5.1 lspci

Весьма распространённая проблема на ноутбуках - кардридеры от дядюшки Ляо, к которым даже не знаешь, как подключиться. В консоли от простого пользователя пишем:

```
# lspci
```

В результате будет выведена информация об устройствах, подключённых к шине PCI, как и следует из названия утилиты.

Теперь есть довольно подробная информация об устройстве, которая структурирована так:

```
01:03.2 Generic system peripheral  
[0805]: Ricoh Co Ltd R5C822 SD/SDIO/MMC/MS/MSPro  
Host Adapter (rev 17)
```

Идентификатор (Device ID), Класс (Class), Производитель (Vendor), Наименование устройства (Device), Ревизия (Revision).

Можно получить дополнительную информацию по устройству, которая может быть иногда полезна при устранении проблем с работой оборудования. Для этого нужно воспользоваться подробным (verbose) режимом работы программы:

```
# lspci -v
```

Вывод будет очень длинным, но то, что относится к рассматриваемому в примере устройству, будет выглядеть так:

```
01:03.2 Generic system peripheral [0805]:
Ricoh Co Ltd R5C822 SD/SDIO/MMC/MS/MSPro Host
Adapter (rev 17)
Subsystem: ASUSTeK Computer Inc. Unknown device 1997
Flags: bus master, medium devsel, latency 64, IRQ 21
Memory at fe8fe400 (32-bit, non-prefetchable) [size=256]
Capabilities: [80] Power Management version 2
```

Отсюда можно узнать, например, номер IRQ, на котором висит устройство.

### 4.5.2 Устройства с подключением по USB

Аналогичная команда есть и для USB-устройств, что тоже весьма полезно, например, при подключении USB-IR или USB-Bluetooth устройств. Так вы узнаете точную марку устройства:

```
# lsusb
```

Результат:

```
Bus 005 Device 001: ID 0000:0000
Bus 004 Device 003: ID 0a12:0001 Cambridge
Silicon Radio, Ltd Bluetooth Dongle (HCI mode)
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 002: ID 046d:c03e Logitech,
Inc. Premium Optical Wheel Mouse
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
```

Можно так же заметить, что помимо синезубого адаптера, сейчас подключена ещё и оптическая мышка с колёсиком от Logitech.

### 4.5.3 Расширенная информация: lshw

Эта утилита по умолчанию, как правило, не устанавливается, но в репозиториях она должна быть (есть в Дебиане 4 ставится при помощи `aptitude install lshw`). Утилита `lshw` выводит структурированный список оборудования вместе с информацией об устройствах. Например, простой запуск от пользователя:

```
# lshw
```

даёт много пищи для размышлений:

```
notebeast
description : Computer
width: 32 bits
*-core
description : Motherboard
physical id : 0
*-memory
description : System memory
physical id : 0
size : 495MB
*-cpu
product: Intel(R) Pentium(R) M processor 1.73GHz
vendor: Intel Corp.
physical id : 1
bus info: cpu@0
version : 6.13.8
size : 1729MHz
capacity: 1729MHz
width: 32 bits
capabilities : fpu fpu_exception wp vme
de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat clflush dts acpi mmx
fxsr sse sse2 ss tm pbe nx est tm2 cpufreq
*-pci
```

```
description : Host bridge
product: Mobile 915GM/PM/GMS/910GML
Express Processor to DRAM Controller
vendor: Intel Corporation
physical id: 100
bus info: pci@00:00.0
version: 04
width: 32 bits
clock: 33MHz
*—display:0
description : VGA compatible controller
product: Mobile 915GM/GMS/910GML
Express Graphics Controller
vendor: Intel Corporation
physical id: 2
bus info: pci@00:02.0
version: 04
size: 256MB
width: 32 bits
clock: 33MHz
% capabilities : vga bus _master cap _list
configuration: latency=0
<...>
```

Список длинный, поэтому приведена только часть. Очень полезно для анализа оборудования.

#### 4.5.4 Ищем сведения об устройстве с помощью dmesg в Linux

Эта команда предоставляет доступ к логам ядра - того, что происходило при загрузке и того, что происходит в процессе работы. При загрузке устройства инициализируются и опознаются системой, и если что-то идёт не так, в логах обязательно это будет отражено.

Лог dmesg как правило очень длинный и изобилует подроб-

ностями работы с оборудованием. Читать его полностью довольно утомительно, поэтому стоит воспользоваться потоковой утилитой поиска `grep`. Например, чтобы узнать подробности о процессоре, нужно написать:

```
# dmesg | grep CPU

SLUB: Genslabs=22, HWalign=64, Order=0-1,
MinObjects=4, CPUs=1, Nodes=1
CPU: After generic identify, caps:
afe9fbff 00100000 00000000 00000000
00000180 00000000 00000000
CPU: L1 I cache: 32K, L1 D cache: 32K
CPU: L2 cache: 2048K
CPU: After all inits, caps: afe9fbff
00100000 00000000 00002040 00000180
00000000 00000000
% Intel machine check reporting enabled on CPU#0.
CPU: Intel(R) Pentium(R) M processor 1.73GHz stepping 08
ACPI: CPU0 (power states: C1[C1] C2[C2])
ACPI: Processor [CPU1] (supports 8 throttling states)
Switched to high resolution mode on CPU 0
Use X86_ACPI_CPUFREQ (acpi-cpufreq) instead.
```

Так можно многое узнать о работе своего железа. Эта информация может пригодиться при сборке своего ядра и/или устранении проблем (последние строки `dmesg` содержат информацию о том, что происходило недавно с системой). Именно поэтому на форумах требуют выкладывать логи системы - это и есть исчерпывающая информация о работе вашей системы и её оборудовании.

Лог `dmesg` и многие другие логи хранятся в `/var/log/` хотя часть из них доступна только привилегированному пользователю - пользуйтесь `sudo`.

#### 4.5.5 Ищем сведения об устройстве с помощью /proc в Linux

Файловая система /proc на самом деле является "слепок" состояния системы и её переменных, там хранится множество полезных сведений о системе. Например, уровень заряда батарей ноутбука и скорости вращения вентиляторов, информация о подключённых устройствах - и многое другое.

Чтобы просмотреть файлы в директории /proc делаем так:

```
$ ls /proc/
```

В ответ получите список файлов, в которых хранится информация о текущем состоянии системы. Например, чтобы узнать подробности о процессоре, можно написать:

```
$ cat /proc/cpuinfo
```

В результате чего имеем весьма подробную информацию о процессоре:

```
processor : 0
vendor_id : GenuineIntel
cpu family : 6
model : 13
model name : Intel(R) Pentium(R) M processor 1.73GHz
stepping : 8
cpu MHz : 1729.000
cache size : 2048 KB
fdiv_bug : no
hlt_bug : no
f00f_bug : no
coma_bug : no
fpu : yes
fpu_exception : yes
cpuid level : 2
wp : yes
flags : fpu vme de pse tsc msr
```



```
pae mce cx8 apic sep mtrr pge mca
cmov pat clflush dts acpi mmx fxsr
sse sse2 ss tm pbe nx est tm2
bogomips : 3459.64
clflush size : 64
```

Здесь же, в каталоге /proc, хранятся и обновляются сведения о скоростях вращения вентиляторов, заряде батарей и прочих ACPIшных радостях (если оные поддерживаются для вашего оборудования ядром). Вот, к примеру, как можно узнать состояние батарей ноутбука:

```
$ cat /proc/acpi/battery/BAT0/info
```

Отсюда берут данные все графические утилиты мониторинга, типа **klaptop\_check** (в KDE висит в трее и показывает состояние батарей). Вот что там имеется:

```
present: yes
design capacity: 50600 mWh
last full capacity: 51381 mWh
battery technology: rechargeable
design voltage: 11100 mV
design capacity warning: 5060 mWh
design capacity low: 2530 mWh
capacity granularity 1: 506 mWh
capacity granularity 2: 506 mWh
model number: S5-2P24
serial number:
battery type: LiIon
OEM info: ASUSTEK
```

Как видно, весьма полезная, и главное - постоянно обновляемая информация.

#### 4.5.6 Ищем сведения о жёстком диске в Linux

Сведения о ёмкости и геометрии винчестера можно получить при помощи `fdisk`, запущенном от рута или через `sudo`:

```
# fdisk -l /dev/hda
```

```
Disk /dev/hda: 60.0 GB, 60011642880 bytes
255 heads, 63 sectors/track, 7296 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Device Boot Start End Blocks Id System
/dev/hda1 1 36 289138+ 82 Linux swap / Solaris
/dev/hda2 * 37 7296 58315950 83 Linux
```

Если требуется получить более глубокие сведения о возможностях винчестера, например о поддержке им DMA или S.M.A.R.T, потребуется утилита `hdparm`, включённая в состав (почти?) всех современных дистрибутивов. Она так же требует рутовых полномочий, и обращаться с ней нужно весьма осторожно (использование параметров без чёткого понимания того, что вы делаете, может привести к потере данных). Так, например, можно запросить информацию о винчестере и его возможностях:

```
$ sudo hdparm -I /dev/hda
```

```
/dev/hda:
```

```
ATA device, with non-removable media
Model Number: HTS421260H9AT00
Serial Number: HKA210AJGKHV1B
Firmware Revision: HA2OA70G
Standards:
Used: ATA/ATAPI-7 T13 1532D revision 1
Supported: 7 6 5 4
Configuration:
```

```

Logical max current
cylinders 16383 65535
heads 16 1
sectors/track 63 63
--
CHS current addressable sectors: 4128705
LBA user addressable sectors: 117210240
LBA48 user addressable sectors: 117210240
device size with M = 1024*1024: 57231 MBytes
device size with M = 1000*1000: 60011 MBytes (60 GB)
Capabilities :
LBA, IORDY(can be disabled)
Standby timer values: spec'd by Vendor, no device specific
minimum
R/W multiple sector transfer: Max = 16 Current = 16
Advanced power management level: 128 (0x80)
Recommended acoustic management value: 128, current value:
254
DMA: mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3
udma4 *udma5
Cycle time: min=120ns recommended=120ns
PIO: pio0 pio1 pio2 pio3 pio4
Cycle time: no flow control=240ns IORDY flow control=120ns
Commands/features:
Enabled Supported:
* SMART feature set
Security Mode feature set
* Power Management feature set
* Write cache
* Look-ahead
* Host Protected Area feature set
* WRITE_BUFFER command
* READ_BUFFER command
* NOP cmd
* DOWNLOAD_MICROCODE
* Advanced Power Management feature set

```

```

Power-Up In Standby feature set
* SET_FEATURES required to spinup after power up
Address Offset Reserved Area Boot
SET_MAX security extension
* Automatic Acoustic Management feature set
* 48-bit Address feature set
* Device Configuration Overlay feature set
* Mandatory FLUSH_CACHE
* FLUSH_CACHE_EXT
* SMART error logging
* SMART self-test
* General Purpose Logging feature set
* WRITE_{DMA|MULTIPLE}_FUA_EXT
* 64-bit World wide name
Security:
Master password revision code = 65534
supported
not enabled
not locked
not frozen
not expired: security count
not supported: enhanced erase
46min for SECURITY ERASE UNIT.
HW reset results:
CBLID— above Vih
Device num = 0 determined by the jumper
Checksum: correct

```

Здесь можно видеть не только размер и геометрию винчестера, но и то, что DMA для диска активирован (стоит `udma5`) и возможности S.M.A.R.T (самотестирование) поддерживается.

Так же можно узнать о состоянии винчестера при помощи утилит `hddtemp` и `smartctl`. Например, текущую температуру винчестера можно узнать так:

```
# hddtemp /dev/hda
```

```
/dev/hda: HTS421260H9AT00: 33°C
```

Возможный выход винчестера из строя помогает засечь команда `smartctl`, с помощью которой можно протестировать жёсткий диск:

```
# smartctl -a /dev/hda
```

Диагностический вывод здесь тоже весьма подробный, вот только часть:

```
SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME FLAG VALUE WORST THRESH
TYPE
UPDATED WHEN_FAILED RAW_VALUE
1 Raw_Read_Error_Rate 0x000b 100 100 062 Pre-fail Always
- 0
2 Throughput_Performance 0x0005 100 100 040 Pre-fail
Offline - 0
3 Spin_Up_Time 0x0007 206 206 033 Pre-fail Always - 2
4 Start_Stop_Count 0x0012 099 099 000 Old_age Always -
1719
5 Reallocated_Sector_Ct 0x0033 100 100 005 Pre-fail Always
- 0
7 Seek_Error_Rate 0x000b 100 100 067 Pre-fail Always - 0
8 Seek_Time_Performance 0x0005 100 100 040 Pre-fail Offline
- 0
9 Power_On_Hours 0x0012 088 088 000 Old_age Always -
5549
10 Spin_Retry_Count 0x0013 100 100 060 Pre-fail Always - 0
12 Power_Cycle_Count 0x0032 100 100 000 Old_age Always -
875
191 G-Sense_Error_Rate 0x000a 100 100 000 Old_age Always
- 0
```

```

192 Power-Off_Retract_Count 0x0032 100 100 000 Old_age
    Always - 2
193 Load_Cycle_Count 0x0012 071 071 000 Old_age Always -
    299291
194 Temperature_Celsius 0x0002 166 166 000 Old_age Always
    - 33 (Lifetime Min/Max 11/43)
196 Reallocated_Event_Count 0x0032 100 100 000 Old_age
    Always - 1
197 Current_Pending_Sector 0x0022 100 100 000 Old_age
    Always - 0
198 Offline_Uncorrectable 0x0008 100 100 000 Old_age Offline
    - 0
199 UDMA_CRC_Error_Count 0x000a 200 200 000 Old_age
    Always - 0

```

#### 4.5.7 Считываем информацию о BIOS с помощью dmidecode в Linux

Причиной множества мелких пакостей может быть BIOS или его настройки, особенно на ноутбуках. В Линукс имеется несколько утилит для чтения информации о BIOS, например dmidecode. Здесь можно узнать о точном названии и версии BIOS, возможностях BIOS и другой интересной информации. Пишем от рута:

```
# dmidecode
```

и в результате получаем, например, такие сведения:

```

# dmidecode 2.8
SMBIOS 2.3 present.
34 structures occupying 1318 bytes.
Table at 0x000F8DD0.

Handle 0x0000, DMI type 0, 20 bytes
BIOS Information

```

Vendor: American Megatrends Inc.  
Version: 0205  
Release Date: 06/10/2005  
Address: 0xF0000  
Runtime Size: 64 kB  
ROM Size: 512 kB  
Characteristics:  
ISA is supported  
PCI is supported  
PC Card (PCMCIA) is supported  
PNP is supported  
BIOS is upgradeable  
BIOS shadowing is allowed  
ESCD support is available  
Boot from CD is supported  
Selectable boot is supported  
BIOS ROM is socketed  
EDD is supported  
5.25"/360 KB floppy services are supported (int 13h)  
5.25"/1.2 MB floppy services are supported (int 13h)  
3.5"/720 KB floppy services are supported (int 13h)  
3.5"/2.88 MB floppy services are supported (int 13h)  
Print screen service is supported (int 5h)  
8042 keyboard services are supported (int 9h)  
Serial services are supported (int 14h)  
Printer services are supported (int 17h)  
CGA/mono video services are supported (int 10h)  
ACPI is supported  
USB legacy is supported  
AGP is supported  
LS-120 boot is supported  
ATAPI Zip drive boot is supported  
BIOS boot specification is supported  
Function key-initiated network boot is supported

Весьма познавательно, и не нужно перезагружаться и входить в BIOS.

## 4.6 Настройка SANE: scanner in linux

Если производитель не закрывает спецификаций на оборудование, то сканер должен поддерживаться и будет работать с помощью SANE - Scanner Access Now Easy.

### 4.6.1 Установка сканера

Нужен пакет sane и xsane. Итак, ставим:

```
# aptitude install sane xsane
```

Осталось поискать сканер: от рута пишем:

```
# sane-find-scanner
```

Эта утилита должна выдать что-то вроде:

```
# sane-find-scanner will now attempt to detect
# your scanner. If the result is different from
# what you expected, first make sure your scanner
# is powered up and properly connected to your computer.

# No SCSI scanners found. If you expected something
# different, make sure that you have loaded a
# SCSI driver for your SCSI adapter.

found USB scanner (vendor=0x04b8 [EPSON],
product=0x011b [EPSON Scanner])
at libusb:001:089

# Your USB scanner was (probably) detected.
# It may or may not be supported by
# SANE. Try scanimage -L and read the
```



```
# backend's manpage.
```

Сканер опознан, нашёлся и готов радовать нас.

#### 4.6.2 Сканировать может только root?

По умолчанию только root имеет доступ к сканеру, но это легко исправить. Для этого нужно внести пользователя, под которым хочется сканировать, в группу scanner. Делается это так:

```
# gpasswd -a имя_пользователя scanner
```

Для того, чтобы изменения вступили в силу, нужно перелогиниться. После этого в консолике пишем:

```
$ groups
```

При этом будет выдан список всех групп, в которых пользователь состоит. Например, вот что пишет эта утилита для меня

```
penta4 dialout cdrom floppy audio video plugdev scanner
```

Последняя строчка говорит о том, что пользователь penta4 имеет право сканировать.

## Глава 5

# Сети

### 5.1 Швейцарский нож для сетей - netcat

Простая и очень нужная утилита, способная передавать данные по сети. Утилита netcat (или nc) должна быть в любом дистрибутиве, и даже в Knoppix она тоже есть.

Идея netcat очень проста: он просто перенаправляет поток данных в порт сетевого устройства, а на другом конце поток перенаправляется в файл. Отсюда и название: NETcat - тот же cat, но сетевой. При передаче файлов следует первым запускать nc на сервере (куда будут передаваться данные), а потом уже с клиента (откуда передаются данные) пересылать файлы.

#### 5.1.1 На стороне сервера (куда передаём данные)

Нужно запустить netcat так, чтобы он слушал определённый порт, и перенаправлял получаемый поток в файл. Вот так:

---

```
$ nc -l -p 3333 > file.txt
```

Это откроет на прослушивание порт 3333, и сохранит полученные данные в файл `file.txt` в текущем каталоге. При этом программа не вернёт управление консоли, и будет ждать, пока вы вручную завершите её.

### 5.1.2 На стороне клиента (откуда передаются данные)

: Для этого передаём по конвейеру файл `nc` так:

```
$ cat file.txt | nc 192.168.1.1 3333
```

Это передаст файл `file.txt` на машину с IP-адресом 192.168.1.1, где уже запущен `nc`, слушающий порт 3333.

Можно с помощью `nc` устроить чат с пользователем на другой машине. Со стороны клиента просто пишете

```
$ nc 192.168.1.1 3333
```

и на том конце будет видно всё, что вы печатаете, в том числе кириллицу.

## 5.2 Настройка сетевого интерфейса в Linux

### 5.2.1 Текущие настройки сетевой карты в Linux

В консоли с привилегиями рута команду:

```
# ifconfig
```

или

```
$ sudo ifconfig
```

В результате в консоль будет выдано что-то вроде:

```
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:104 errors:0 dropped:0 overruns:0 frame:0
TX packets:104 errors:0 dropped:0 overruns:0 carrier:0
collisions :0 txqueuelen:0
RX bytes:9387 (9.1 KiB) TX bytes:9387 (9.1 KiB)

eth0 Link encap:Ethernet HWaddr 00:0A:E4:53:AA:2D
inet addr:192.168.1.5 Bcast:192.168.1.255
Mask:255.255.255.0
UP BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions :0 txqueuelen:1000
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
Interrupt:21 Base address:0x2c00
```

Отсюда видно, что настроены два сетевых интерфейса: петлевой (loopback) под названием lo, и обычная сетевая карта (которая Ethernet) называемая eth0. Причём если в интерфейс eth0 вставлен сетевой провод, идущий в свитч, будет написано RUNNING

Простая настройка сетевого интерфейса возможна командой:

```
# ifconfig eth0 192.168.1.5
```

Чтобы заглушить сетевой интерфейс, набираем:

```
# ifdown eth0
```

### 5.2.2 Настройка сетевого соединения в Линукс

Для этого потребуются права администратора и текстовый редактор. Так же нужно представлять, какой IP-адрес следует присвоить сетевой карте, адрес шлюза и прочие параметры.

Сначала нужно выяснить, как называется сетевой интерфейс, соответствующий вашей сетевой карте. Делается это так: в консоли пишем

```
$ dmesg | grep -i Eth
```

В результате должны получить что-то вроде (выделение моё):

```
8139too Fast Ethernet driver 0.9.28
eth0: RealTek RTL8139 at 0xdf822c00,
00:15:f2:51:ad:da, IRQ 21
eth0: Identified 8139 chip type 'RTL-8101'
```

Так, похоже, что называется он eth0 (потому, что ethernet, 0 - нулевое устройство - довольно логично). В этом примере дальше будет предполагаться, что интерфейс - eth0.

### 5.2.3 Настройка сетевого соединения со статическим IP

Для этого следует отредактировать файл:

```
# nano /etc/network/interfaces
```

или

```
$ sudo nano /etc/network/interfaces
```

Перед тем, как его править, следует решить, какой IP-адрес назначается машине. Если вы подключаетесь к уже настроенной сети - хорошо бы посмотреть, какие настройки есть там.

Например, если подключение происходит к CorvettTelecom, настройки вам дадут. Если же сеть самопальная, например из двух компьютеров, нижеследующий пример подойдет.

В общем, при назначении статических адресов нужно указать: IP-адрес (address), маску подсети (netmask) и адрес шлюза (gateway). Для локальной сети файл `/etc/network/interfaces` нужно изменить так:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.5
netmask 255.255.255.0
gateway 192.168.1.1
```

В этом примере назначен IP-адрес 192.168.1.5.

#### 5.2.4 Настройка сетевого соединения с динамическим IP

Редактируем тот же файл, отвечающий в Линукс за сетевые настройки:

```
# nano /etc/network/interfaces
```

или

```
$ sudo nano /etc/network/interfaces
```

Здесь всё проще: нужно лишь привести файл к виду:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

Так же можно настроить и из консоли:

```
#dhcpcd eth0
```

Для этого в системе должен быть установлен пакет `dhcpc3-client`

### 5.2.5 Два слова о беспроводных сетях

Чтобы изменить тип интерфейса

```
# wlanconfig ath0 destroy  
# wlanconfig ath0 create wlandev wifi0  
# wlanmode adhoc
```

Для того, чтобы посмотреть список сетей, набираем:

```
# iwlist ath0 scan
```

или

```
# wlanconfig ath0 list ap
```

Если есть открытая сеть и нужно к ней подключиться, набираем от рута:

```
# iwconfig ath0 essid SomeESSID
```

Для того, чтобы соединение шифровалось с использованием WEP, пишем:

```
# iwconfig ath0 key ...
```

### 5.2.6 Настройка адресов серверов имён (DNS, nameserver)

Если вы не настроите это, то сможете подключаться к ресурсам сети только по IP-адресам, что несколько неудобно. Для того, чтобы в Линукс прописать адреса серверов имён (DNS address), нужно отредактировать файл `/etc/resolv.conf` или создать его, если его нет.

```
# nano /etc/resolv.conf
```

или

```
$ sudo nano /etc/resolv.conf
```

И прописать адреса следующим образом:

```
nameserver 192.168.1.1
nameserver 192.168.2.1
```

столько, сколько их нужно. Слово `nameserver` добавлять обязательно.

Чтобы изменения настройки сетевых интерфейсов вступили в силу, следует дать команду:

```
# /etc/init.d/networking restart
```

или

```
$ sudo /etc/init.d/networking restart
```

Всё, после этого настройки вступают в силу для всех сетевых интерфейсов.

## 5.3 Как узнать MAC адрес сетевой карты

### 5.3.1 Временная смена MAC-адреса в Linux

Для этого набираем



```
$ sudo ifconfig eth0 hw ether xx:xx:xx:xx
```

при этом MAC-адрес изменится и будет оставаться таковым до перезагрузки. Если нужно сделать так, чтобы MAC-адрес был таким всегда, нужно придумать в соответствующий бу-бен системный конфиг.

### 5.3.2 Постоянный изменённый MAC-адрес в Linux

Для этого идём в `/etc/network/if-pre-up.d/` и создаём там от-рута файл, например имени `pre-up`.

В этот файл пишем следующее:

```
#!/bin/sh
ifconfig eth0 hw ether 00:00:00:00
```

Нули замените своим мак-адресом. Чтобы изменения всту-пили сразу же, без перезагрузки, даём команду:

```
# /etc/init.d/networking restart
```

При этом в первый раз (особенно при работающем интер-фейсе) может вылезти такое:

```
# /etc/init.d/networking restart
Setting up IP spoofing protection: rp_filter .
Reconfiguring network interfaces ...
SIOCDELRT: No such process
ifup: interface lo already configured
SIOCSIFHWADDR: Device or resource busy
run-parts: /etc/network/if-pre-up.d/pre-up
exited with return code 1
done.
```

Повторяем команду, на этот раз всё должно быть без шума и пыли:

```
notebeast:/home/beast# /etc/init.d/networking restart
Setting up IP spoofing protection: rp_filter.
Reconfiguring network interfaces ... ifup:
interface lo already configured
done.
```

Всё, MAC изменён и останется таковым и после перезагрузки.

### 5.3.3 Другие способы смены MAC адреса

Сменить MAC-адрес в Линукс можно разными способами - на то он и Линукс. Например, можно заменить содержимое `/etc/network/interfaces` на следующее:

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp
hwaddress ether 00:00:00:00:00:00
```

Нули, как и ранее, заменяем на требуемый MAC-address. В Debian всё проще: в описание интерфейса в `/etc/network/interfaces` добавляем:

```
hwaddress ether XX:XX:XX:XX:XX:XX
```

Другие способы описаны в комментариях к этому посту ниже.

### 5.3.4 Узнать MAC-адрес сетевой карты в Linux

Чтобы узнать текущий MAC-адрес просто: набираем

```
$ sudo ifconfig
```

И видим нечто вроде:

```
eth0 Link encap:Ethernet
HWaddr 00:0A:E4:53:AA:2D
inet addr:10.26.49.77 Bcast:10.26.63.255
Mask:255.255.240.0
UP BROADCAST RUNNING MULTICAST MTU:1500
    Metric:1
RX packets:208554 errors:0 dropped:0
overruns:0 frame:0
TX packets:125071 errors:0 dropped:0
overruns:0 carrier:0
    collisions:0 txqueuelen:1000
RX bytes:40664531 (38.7 MiB)
TX bytes:45919980 (43.7 MiB)
Interrupt:21 Base address:0x4c00

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:18511 errors:0 dropped:0
overruns:0 frame:0
TX packets:18511 errors:0 dropped:0
overruns:0 carrier:0
    collisions:0 txqueuelen:0
RX bytes:537155 (524.5 KiB)
TX bytes:537155 (524.5 KiB)
```

Красное полужирное выделение и есть MAC-адрес (в данном случае, сетевой карты моего ноутбука).

## 5.4 Настройка и использование SSH в Linux

### 5.4.1 Установка SSH в Linux на примере Debian

Итак, всё, что нам нужно для установки полного комплекта удалённого управления компьютером (SSH-клиент и SSH-

сервер) давным-давно лежит в репозитории. Лёгким движением ставим пакет:

```
# apt-get install ssh
```

и ждём несколько мгновений, когда оно настроится. После этого мы получим возможность SSH доступа в систему и управления ей. Так как технология эта кросс-платформенная, то можно управлять по SSH Linux или FreeBSD можно и из Windows. Для этого есть putty, SSH Windows клиент.

### 5.4.2 На стороне клиента

Теперь надо поправить настройки, которые лежат в каталоге /etc/ssh - конфиг для клиента называется ssh-config, конфиг для сервера, соответственно, sshd-config. На своей, клиентской, стороне, настраиваем возможность приёма X11Forward, ищем и меняем ключи на:

```
ForwardX11 yes  
ForwardX11Trusted yes
```

Клиентская машина теперь может запускать удалённо графические приложения на сервере. Настройка SSH на стороне клиента закончена, теперь идём к админу далёкого сервера... В принципе, можно на клиентской стороне ничего не менять, а логиниться на удалённую машину так:

```
$ ssh -X user@server1.mydomain.com
```

или

```
$ ssh -X user@192.168.x.x
```

если лезть в конфиги на своей стороне не хочется, но у меня это почему-то не работало.

### 5.4.3 На стороне сервера

Теперь нужно настроить SSH сервер: в конфигах машины-сервера, к которой будем подсоединяться (у вас ведь есть её рутовый пароль, так ведь?) в `/etc/ssh/sshd-config` ищем и меняем ключи на:

```
X11Forwarding yes
X11DisplayOffset 10
X11UseLocalhost yes
```

Этим мы разрешаем серверу запускать удалённо графические приложения и отправлять их на клиентскую машину. Перестартуем сервис:

```
$ sudo /etc/init.d/ssh restart
```

Теперь мы сможем заходить на машину не только в консольном режиме, но и с запуском иксовых приложений.

Если хочется разрешить вход только с определённых машин, нужно подправить строки в конфиге `/etc/ssh/sshd_config`

```
AllowUsers hacker@*

AllowUsers *@192.168.1.*
```

Впрочем, это уже для более продвинутых товарищей.

### 5.4.4 SSH в действии

Всё готово, и теперь я приведу несколько команд SSH для примера. Открываем консольку и пишем в ней:

```
$ ssh имя_пользователя_удалённой_машины@ip_адрес
_или_сетевое_имя_удалённой_машины
```

Например, в моём случае, когда я захожу удалённо на ноутбук, пишу `ssh beast@192.168.1.5` - так как у меня не настроен

сервер имён, пишу адреса. Опять-таки, доступ по SSH может быть не только из Linux или FreeBSD, но и из Windows - при помощи putty.

## Глава 6

# Принтеры

### 6.1 Сетевая печать из Linux на Windows принтер используя SAMBA

Пример: есть компьютер под Windows XP по адресу 192.168.1.1, на нём по SAMBA виден расшаренный принтер LaserJet 1200, который имеет сетевое имя HPLaserJ. С других машин под Windows печать на него идёт, но через CUPS при попытке отправить задание на печать вылезает ошибка.

При использовании SMB протокола пишет:

Connection failed with error NT\_STATUS\_ACCESS\_DENIED

Если пытаюсь печатать через IPP, получаю:

Unable to connect to SAMBA host, will retry in 60 seconds...

Пароля на принтеры нет. **Решение проблемы:** добавляем в веб-форме CUPS принтер, выбираем способ подключения Windows printer via Samba. Далее в строке адреса пишем:

`smb://guest@192.168.1.1/HPLaserJ`

И всё! Печать пошла! А если авторизация через домен, то надо

```
smb://username:userpassword@
```

```
domainname/computername/printername
```

## 6.2 Сетевая печать: как печатать из Linux в Linux

Машина-сервер, к которой подключён принтер: 192.168.1.1

Машина-клиент, с которой будем печатать на сервер: 192.168.1.2

### 6.2.1 На стороне сервера

Идём править конфиг `/etc/cups/cupsd.conf`, в котором разрешаем печатать не только локальной машине, но и тем, кому надо (здесь - всем машинам подсети 192.168.1.\*):

```
Allow From 127.0.0.1
Allow From 192.168.1.*
```

Это надо будет вписать / раскомментировать где-то в конце конфига. Листайте его внимательно, он длинный. Когда конфиг поправите, надо перезапустить сервер печати:

```
# /etc/init.d/cupsys restart
```

Всё, с сервером разобрались, теперь идём к клиенту.

### 6.2.2 На стороне клиента

Здесь надо поправить другой конфиг, `/etc/cups/client.conf`, в котором указываем IP-адрес сервера, на который будем печатать. Это достигается указанием адреса здесь:

```
ServerName 192.168.1.1
```



Не забываем сохранить конфиг. Всё готово - теперь, если вы зайдёте на машине-клиенте в админ-панель CUPS в браузере, указав адрес **http://localhost:631** вы должны увидеть принтер, который на сервере:

### 6.3 Как настроить принтеры с помощью CUPS

CUPS - это Общая Система Печати в UNIX (Common Unix Printing System). Ставим следующие пакеты:

```
# aptitude install cupsys gs—esp foomatic—bin foo2zjs
```

вместе со всеми зависимостями. Важно поставить не только сам CUPS, но и программы, которые он использует, имеющие не вполне очевидные названия.

Кстати, весьма распространённая ошибка: ставят только cupsys, или забывают foo2zjs, в результате чего при отправке задания на печать из принтера ничего не вылезает, а в логах вот такие записи:

```
I [23/Aug/2005:14:41:46 +0400] Adding end banner page
"none" to job 7.
I [23/Aug/2005:14:41:46 +0400] Job 7 queued on
'HPDeskJet930C' by 'penta4'.
E [23/Aug/2005:14:41:46 +0400] Unable to convert file 0 to
printable format for job 7!
I [23/Aug/2005:14:41:46 +0400] Hint: Do you have ESP
Ghostscript installed?
I [23/Aug/2005:14:41:46 +0400] Hint: Try setting the LogLevel
to "debug".
```

#### 6.3.1 В печать!

Для того, чтобы добавить новый принтер (или менять опции уже установленного), запускаем любой браузер и в адресной

строке набираем:

```
localhost :631
```

Страница CUPS может долго загружаться, если указаны много других сетевых подключений или неправильно настроен DNS. Попробуйте на время вырубить сетевой интерфейс

```
# ifconfig eth0 down
```

### 6.3.2 Печать из GiMP

Для этого потребуется установить ещё несколько пакетов и пожертвовать несколько мегабайт дискового пространства:

```
# aptitude install cupsys-driver-gimpprint gimp-print
```

поставится и настроится. Теперь запускаем GiMP, открываем любое изображение и жмём Файл - Печать.

## Глава 7

# Ядро

### 7.1 Управление ядром Linux реального времени

Для управления приоритетами ядра есть утилиты из пакета schedutils.

#### 7.1.1 Привязка задачи к процессору

Если у вас многоядерная или многопроцессорная машина, есть смысл некоторые задачи раскидать по процессорам принудительно. Для этих целей есть программа taskset.

```
# taskset -p 1 5907
```

Получаем в ответ:

```
pid 5907's current affinity mask: 1  
pid 5907's new affinity mask: 2
```

Задача, таким образом, перекинута на второй процессор.

### 7.1.2 Изменение приоритета процессов

Чтобы задача получила приоритет реального времени, следует выполнить команду:

```
# chrt -p 6091
```

В ответ должны получить:

```
pid 6091's current scheduling policy : SCHED_OTHER
pid 6091's current scheduling priority : 0
```

Это значит, что задача теперь будет выполняться неотложно, с минимально возможными задержками.

### 7.1.3 Изменение приоритета ввода-вывода

Много задач требуют не только точного интервала исполнения, но и чтобы данные перекачивались как можно быстрее. Для этого можно отдать приоритет ввода-вывода наиболее критичной задаче. Возможно выставить следующие типы приоритетов: Бездействие (Idle), Наилучшая работа (Best effort) и Реальное время (Real time).

### 7.1.4 Приоритет Idle

Программа с приоритетом idle получит доступ к диску только тогда, когда никакие другие программы к диску не обращаются в течение определённого времени.

#### Приоритет Best effort

Этот приоритет выбирается по умолчанию для любого процесса, который не требует иного приоритета ввода-вывода. Программы наследуют установки процессорного приоритета для приоритета ввода-вывода, можно так же передавать аргумент от 0 (высокий приоритет) до 7 (низкий приоритет). Ресурсы между приложения с одинаковым приоритетом best effort распределяются по принципу "каждой сестре по серьге".

### 7.1.5 Приоритет Real time

Задачам с приоритетом реального времени выделяется доступ к диску в первую очередь и вне зависимости от происходящего в системе. Поэтому приоритет реального времени следует использовать осторожно, чтобы не перекрыть кислород другим процессам. Как и в случае приоритета best effort, предусмотрено 8 уровней приоритета, от 0 (высокий приоритет) до 7 (низкий приоритет).

Для выбора приоритета следует указывать параметр -с и значения:

```
-c1 — реальное время (real time)
-c2 — наилучшая работа (best effort)
-c3 — бездействие (idle)
```

Для указания параметра приоритета используется ключ -пМ, где М - число от 0 (высокий приоритет) до 7 (низкий приоритет).

Например, чтобы задать приоритет реального времени с высшим приоритетом для ввода-вывода приложения с идентификатором

```
# ionice -c2 -p0 -p5812
```

После этого приложению уже никто не мешает монопольно использовать ввод-вывод.

## 7.2 Как собрать ядро в Линукс

Если вы столкнулись с необходимостью собрать ядро Linux, наложить патч для ядра Линукс или просто вы уже забыли, как это делается - здесь приводятся рекомендации и ссылки для "быстрого старта".

### 7.2.1 Как собрать или пересобрать ядро Линукс?

Прежде всего, заполучите статью "Ставим ядро 2.6, или Ядерная физика для домохозяйки" (тут - новая версия этой статьи). Не надо оскорбляться таким названием - это исчерпывающая статья по сборке ядра, её автор, Михаил Зарипов, перевёл на русский почти все опции ядра!

### 7.2.2 Какие могут быть проблемы при сборке ядра?

Проблем при сборке обычно не бывает, только если вы не намудрили с компилятором или недоустановили его. В процессе сборки проблем не так много, они обычно возникают после сборки ядра по следующим причинам:

1. вы не включили в ядро то, что очень нужно, и теперь система не загружается или пропала возможность работы с некоторым оборудованием;
2. вы включили в ядро то, что не надо (особенно с надписью EXPERIMENTAL) и теперь система работает нестабильно или очень медленно или ядро вываливается в kernel panic;
3. вы взяли нестабильное ядро и/или компилятор в бета-версии/выставили дикие флаги оптимизации - вас предупреждали

### 7.2.3 С чего начать?

С того, что к сборке ядра нужно подготовиться: это займёт много времени, особенно если вы это делаете в первый раз. У меня пересборка первого ядра заняла 4 часа - вдумчивое чтение статьи, поиск параметра, включение или выключение и так далее. У вас это может занять времени меньше - затем я это и пишу. Хорошо бы взять конфиг из вашего дистрибутива

- дистрибьютеры в сборке ядер понимают, как правило, больше вас, так что не стесняйтесь. Копируем в каталог с ядром и пишем

```
# make oldconfig
```

Флажки параметров ядра должны расставиться соответствующим конфигу образом, если только в вашем дистрибутиве не принято накладывать на ядро 33 заплатки. Конфиг так же можно попросить у ближайшего линуксоида - будет кого пинать и доставать своими вопросами. Если охота проявить творчество, можно попросить создать конфиг с умолчальными настройками:

```
# make defaultconfig
```

Не увлекайтесь заплатками, особенно поначалу. Каждая из них добавляет возможности в ядро, но может привести к дестабилизации системы и крайне неприятным последствиям. Обзор различных интересных патчей и веток ядер Линукс дан в этой прекрасной статье.

### 7.2.4 Где лежат большие грабли

Как и всё остальное в Линукс, тексты ядра прокомментированы и можно в процессе сборки попросить помощи. В `menuconfig` это крайняя правая кнопка HELP. Комментарии есть практически к каждой опции, более или менее внятные, и их стоит почитать, если из статьи не ясно, что это такое.

Перво-наперво, не включайте в ядро и не делайте модулем никаких функций, рядом с которыми написано слово EXPERIMENTAL или DEPRECATED! Это может вывести систему из стабильного состояния и добавить вам совершенно ненужных проблем - включайте только в том случае, если это очень необходимо и вы знаете, что делаете.

Чтобы система после перезагрузки загрузилась с новым ядром, обязательно включите поддержку следующих опций молотитно:

- файловой системы, которую вы используете на системном диске;
- файловых систем `initramfs` и `cramfs` в меню `Device drivers` — `Block devices`, иначе система не сможет считать стартовый образ ядра;
- поддержку вашего дискового контроллера в `Device drivers` — `ATA/ATAPI/MFM/RLL support`

Чтобы не ломать голову и не думать, почему не работает что-то, имейте в виду:

- То, что может быть будет подключаться к системе, полезно сделать модулем, то, чем будете пользоваться постоянно - лучше монолитно (об исключениях - ниже).
- Драйвера для звуковых карт включаются модульно в меню `Device drivers` — `Sound` — `Advanced Linux Sound Architecture` — чтобы потом их проще было конфигурировать (если для этого нет крайней необходимости, не пользуйтесь OSS).
- Если в системе есть SATA-диски, планируется подключать флешки и писать CD или DVD-диски - не выключайте поддержку SCSI в меню `Device drivers` — `SCSI device support`, так как очень многое в Линукс работает "как протокол поверх SCSI".
- Если вы планируете пользоваться кардридером, обязательно включите опцию `Device drivers` — `SCSI device support` — `Probe all LUNs on each SCSI device`, иначе будете видеть только первый слот.
- Файловые системы, которые планируется использовать редко, сделайте модулем. Обязательно включите поддержку петлевого устройства в `Device drivers` — `Block devices` — `Loopback device support`, иначе не сможете монтировать образы дисков.
- Корневую файловую систему включайте монолитно в ядро - иначе система не сможет загрузиться.
- Будьте внимательны, если используете ReiserFS и не включайте все флажки под ним подряд (лучше вообще не



каких не ключайте): один из них (указанный здесь пустым) File systems — Reiserfs support [ ] Enable reiserfs debug mode приводит к замедлению файловой системы и в логах постоянно будет появляться "WARNING! Using reiserfs in slow mode for debugging"или что-то в этом духе. Не включайте этот флажок!

- В меню Block layer — IO Schedulers отмечайте ДА на все планировщики ввода-вывода, потом можно будет на лету их менять (по ситуации).
- Для ноутбуков я бы советовал включать шину ISA в меню Bus options, так как некоторые встроенные устройства прикрепляются именно к ней.
- Включите поддержку вашей сетевой карты модулем в меню Device drivers — Network device support — Device drivers — Ethernet (10 or 100Mbit) —
- Если хочется мониторинг железа, включаем всё модулями в Device drivers — Hardware Monitoring support — и сверяемся ещё раз с постом.
- Не забываем включить поддержку USB и поддержку нужных флешек в Device drivers — USB support — USB Mass Storage support
- Так же стоит сразу включить поддержку оперативной памяти более 1Гб отметив Processor type and features — High Memory Support (off) — 4Gb

Далее для сборки набираем:  
от простого пользователя:

```
make bzImage
```

(собрать ядро)

```
make modules
```

(собрать модули от пользователя root:

```
make modules_install
```

(установить модули)

```
make install
```

(установить ядро)

После того, как ядро собрано, следует создать загрузочный образ (в Дебиане это применяется). Для этого нужно выполнить либо

```
# mkinitramfs -o /boot/initrdвашаверсияядра -  
вашаверсияядра
```

либо (для старых или специфических дистрибутивов)

```
# mkinitrd -o /boot/initrd-ваша_версия_ядра  
ваша_версия_ядра
```

То есть например:

```
# mkinitramfs -o /boot/initrd-2.6.20 2.6.20
```

После этого прописываем в загрузчике (скорее всего это будет GRUB):

```
title Debian Linux (2.6.20)  
root (hd0,0)  
kernel /vmlinuz-2.6.20 ro root=/dev/sda1  
initrd /initrd-2.6.20.img
```

Номер системы, загружаемой по умолчанию, равен 0. Чтобы определить по умолчанию загрузку вашего нового ядра, либо поместите описывающий его раздел файла, в начало, либо измените номер системы по умолчанию на нужный вам (помните, что номера начинаются с 0) Если вы используете GRUB, после редактирования файла конфигурации вам не нужно выполнять какие-либо команды. Вот все грабли, на которые я наступал и которые вспомнил. Уважаемые комментаторы приглашаются к конструктивной критике и дополнению вышеописанных подводных камней.

## Глава 8

# Продуктивность

### 8.1 GraphViz: создание графов и взаимосвязей

GraphViz это язык управления взаимосвязями - не нужно их чертить руками, их нужно логически описать на языке программирования и доверить работу по вычерчиванию графов GraphViz.

Быстрый старт: открываем свой любимый текстовый редактор и вставляем в него например вот это:

```
digraph G{
Рождение->Юность->Зрелость->Старость->Смерть;
Юность->Смерть;
Зрелость->Смерть;
```

И сохраняем в temp.dot, далее в консоли:

```
$ dot -Tpng temp.dot -o temp.png
```

### 8.1.1 Небольшой пример

Дабы минимизировать проблемы с кириллицей, стоит сохранять данные в кодировке UTF8.

```
{ //Уровни в проектах
node[shape=plaintext]; // чтобы было не видно рамок
edge[color=white]; // чтобы было не видно рамок
50.000->40.000->30.000->20.000->10.000->5.000;
}
```

Последовательность символов -> означает связь двух объектов, shape - форму объекта (рамка вокруг текста), color - соответственно цвет (здесь белый - чтобы не было видно рамки вокруг текста). Группируем данные и даём понять GraphViz, что группа принадлежит к этому уровню:

```
{ // <===== Задачи на уровне 10.000 =====
node[shape=rectangle,fontsize=10]; // порядок важен:
// описание свойств узлов должно идти раньше,
// чем сами узлы
rank=same; 10.000; "Исследования";
"Time-Frequency";
}
```

Теперь выстраиваем взаимосвязи объектов друг с другом:

```
"Time-Frequency"-> "Научные\проекты";
"Исследования"->"Time-Frequency"[dir=both];
// зависимость направлена в обе стороны
```

Тут две тонкости:

1. чтобы текст в рамке разбить на несколько строк, ставим \n
2. если нужно, чтобы стрелка была направлена в обе стороны - взаимная зависимость двух объектов - нужно рядом с зависимостью приписать [dir= both];

## 8.2 Презентации beamer

Для создания презентаций в LaTeX существует пакет beamer.

### 8.2.1 Установка

Ставим всё, что содержит слово beamer и находим:

```
# apt-get install latex-beamer
```

### 8.2.2 Создание презентаций в beamer

После установки пакета latex-beamer вы должны будете новый документ с презентацией объявить классом beamer. Вот как выглядит шапка для латеховского файла:

```
\documentclass{beamer}
\usepackage[T2A]{fontenc}
\usepackage[koi8-r]{inputenc}
\usepackage[english,russian]{babel}
\usepackage{amssymb,amsfonts,amsmath,mathtext}
\usepackage{cite,enumerate,float,indentfirst}
\usepackage[dvips]{graphicx}

\title{Итоги работы проекта}
\author{Михаил Конник и сотоварищи}
\date{лето 2007}
```

Если у вас другая раскладка - поменяйте koi8-r на свою. Кроме того, вы должны указать название презентации (title), авторов(author) и дату (date). После этого в теле документа вы можете сгенерировать титульный слайд командой `\maketitle` и таким образом получить первый слайд своей презентации. Каждый слайд находится в окружении

```
\begin{frame}
.....
```

```
\end{frame}
```

Каждый слайд неплохо дополнять комментариями, чтобы не запутаться потом и не забыть, что отображается на слайдах. Заголовок слайда делается командой

`\frametitle{что вы хотите написать в заголовке}` и отображается в соответствии с темой слайдов. Но об этом позже.

После всего сказанного выше, пробуем сделать презентацию - например, такую:

```
\begin{document}

%%титульная страница
\maketitle

%% основные моменты
\begin{frame}
\begin{center}
\frametitle{Основные достижения:}
\begin{itemize}
\item использование пространственно-некогерентного
света ртутной лампы;
\item применение RAW-конверторов, использование
полного динамического диапазона цифровой
фотокамеры;
\item усовершенствование линейных алгоритмов
деконволюции.
\end{itemize}
\end{center}
\end{frame}

\end{document}
```

Не обращаем внимания на вопли и спокойно конвертируем в PDF.

Хорошо, быстро и без долгих страданий от съехавших шрифтов и картинок офисных пакетов. Если хотим сделать презен-

тацию несколько богаче и профессиональнее по оформлению, нужно лишь указать одну из тем, наличествующих в комплекте. А в комплекте их немало:

```
beamertHEMEAntibes.sty
beamertHEMEBergen.sty
beamertHEMEBerkeley.sty
beamertHEMEBerlin.sty
beamertHEMEBoadilla.sty
beamertHEMEboxes.sty
beamertHEMECopenhagen.sty
beamertHEMEDarmstadt.sty
beamertHEMEDefault.sty
beamertHEMEDresden.sty
beamertHEMEFrankfurt.sty
beamertHEMEGoettingen.sty
beamertHEMEHannover.sty
beamertHEMEIlmenau.sty
beamertHEMEJuanLesPins.sty
beamertHEMELuebeck.sty
beamertHEMELMadrid.sty
beamertHEMELMalmoe.sty
beamertHEMELMarburg.sty
beamertHEMELMontpellier.sty
beamertHEMELPaloAlto.sty
beamertHEMELPittsburgh.sty
beamertHEMELRochester.sty
beamertHEMELSingapore.sty
beamertHEMELSzeged.sty
beamertHEMELWarsaw.sty
```

Так же много и цветовых схем оформления, всё это добро (по крайней мере в Debian) лежит в  
`/usr/share/texmf/tex/latex/beamer/themes`

Чтобы применить тему и цветовую схему оформления к вашей презентации, достаточно до `\begin{document}` написать:

```
\usetheme{Antibes}
\usecolortheme{seahorse}
```



## Глава 9

# Данные

### 9.1 EXIF теги в Linux

В различных форматах графических файлов возможность хранить дополнительные сведения, например, время выдержки, значение диафрагмы, время или ещё что-нибудь. Это обеспечивает формат EXIF (Exchangeable Image File Format). В Linux для работы с теми EXIF данными есть такой мощный пакет утилит, как `exiftools`, всю полезность которого автор этих строк испытал на себе. Вот, например, потребовалось вам сохранить какие-то пометки о том, где сделан этот снимок - тут-то вам `exiftools` и пригодится.

Например, выдираем теги из RAW-файла на примере цифровой камеры Canon EOS 400D

`exiftool IMG_0383.CR2 :`

```
exiftool IMG_0383.CR2
ExifTool Version Number : 6.57
File Name : IMG_0383.CR2
File Size : 8 MB
File Modification Date/Time : 2008:10:22 12:48:10
```

```
File Type : CR2
MIME Type : image/x-raw
Image Width : 1936
Image Height : 1288
Bits Per Sample : 8 8 8
<....>
```

Например, чтобы выцедить из RAW-файла время выдержки, сделать надо вот что:

```
exiftool -shutterspeed IMG_0383.CR2
Shutter Speed : 1
```

Для того, чтобы использовать exif-теги, надо запросить список тегов в которые возможна запись. Пишем:

```
$ exiftool -listw IMG_0383.CR2
```

Так, например из кучи тегов мы хотим писать в тег City. Для этого:

```
$ exiftool -City=Moscow IMG_0383.CR2
1 image files updated
```

И теперь:

```
$ exiftool -City IMG_0383.CR2
City : Moscow
```

Ура, теперь мы в тег City записали, что хотели. Утилита exiftools может не только читать, но и записывать информацию в EXIF-теги. Это может быть полезно и нужно для хранения в RAW-файлах информации о съёмке.

И теперь, чтобы избавить себя от нервотрёпки, делаем так: находим любой записываемый тег, и пишем туда то, что относится к эксперименту. После этого записываем нужные данные во ВСЕ файлы RAW скопом:

```
$ exiftool -City='Moscow' *.CR2
10 image files updated
```

С удовлетворением видим:

```
$ exiftool -City *.CR2
===== IMG_0383.CR2
City : Moscow
```

## 9.2 Небольшая шпаргалка по командам SVN

В этом посте я решил собрать простые и наиболее часто используемые мной команды svn, часть параметров и способов работы с репозиториями svn, которых я склонен забывать.

### 9.2.1 Наполнение репозитория Subversion

Репозиторий, созданный командой:

```
$ svnadmin create /usr/local/svn/newrepos
```

нужно наполнить свежими данными. Для этого есть команда `svn import` это быстрый способ скопировать не версионированное дерево файлов в хранилище, например:

```
$ svn import mytree
    file:///usr/local/svn/newrepos/some/project \
-m "Initial import"
Adding mytree/foo.c
Adding mytree/bar.c
Adding mytree/subdir
Adding mytree/subdir/quux.h

Committed revision 1.
```

Стоит заметить, что после завершения импорта, оригинальное дерево файлов не конвертируется в рабочую копию. Для того, чтобы начать работать вам необходимо создать новую рабочую копию (svn checkout) дерева файлов.

### 9.2.2 Добавление новых файлов и каталогов в репозиторий Subversion

Например, вы создаёте крупный документ в ЛaTeXe, и появился новый каталог с новыми текстами или картинками. Как их добавить в Subversion репозиторий? Очень просто:

```
$ svn add temp A temp A temp/Lit_obzor2_edt.tex
```

Теперь смотрим на статус наших файлов и видим, что против них стоит буква A - значит, они Added, то есть добавлены:

```
$ svn status
? image
? biblio
? optcodeimages
M draft_PhD_Konnik.dvi
! PhD_Konnik.log
A temp
A temp/Lit_obzor2_edt.tex
! PhD_Konnik.toc
M PhD_Konnik_chapter_results.tex~
! PhD_Konnik_chapter_introduction.dvi
! PhD_Konnik.aux
! PhD_Konnik.bbl
! draft_PhD_Konnik.log
! PhD_Konnik.blg
! draft_PhD_Konnik.toc
! draft_PhD_Konnik.aux
! draft_PhD_Konnik.bbl
M PhD_Konnik.dvi
! draft_PhD_Konnik.blg
```

```
M PhD_Konnik_chapter_results.tex
! PhD_Konnik_chapter_introduction.log
```

Они будут помещены в репозиторий при следующей отправке (commit).

### 9.2.3 Обновление рабочей копии до состояния репозитория

Файлы, находящиеся в репозитории свежее, чем у вас в рабочей копии. Дубовый вариант заключается в том, чтобы удалить свою рабочую копию и сделать checkout из репозитория. Это проходит, если репозиторий только ваш и вы уверены, что никаких изменений в рабочей копии вы не потеряете. Но есть вариант и получше:

```
$ svn update U
biblio/my.bib U
biblio/my.bib.bak
Updated to revision 9.
```

Это обновит вашу локальную копию до ревизии, которая находится в репозитории SVN. После этого можно приступать к работе с файлами, так как они обновлены до последней версии.

Латинские буквы перед именем файла имеют свой сакральный смысл, скрытый в недрах документации сокрушительных размеров (кстати, документация очень хороша, и есть даже частичный перевод на русский тут). Собственно, буква означает действие, которое выполнил Subversion для приведения рабочей копии в актуальное состояние:

```
U foo

Updated — Файл foo был обновлен
```

A foo

Added — Файл или директория foo были добавлены в рабочую копию.

D foo

Deleted — Файл или директория foo были удалены из рабочей копии.

R foo

Replaced — Файл или директория foo была заменена в рабочей копии; это значит, что foo был удален, а новый элемент с таким же именем был добавлен.

G foo

merGed — Файл foo получил новые изменения из хранилища, однако ваша локальная копия содержит ваши изменения. Либо изменения не пересекаются, либо они точно такие же, как ваши локальные изменения, поэтому Subversion успешно выполнил слияние изменений хранилища с файлом.

C foo

Conflicting — Файл foo получил от сервера конфликтующие изменения. Изменения с сервера пересекаются с вашими изменениями фала.

### 9.2.4 Просмотр истории ревизий в Subversion

Часто нужно посмотреть, что и как редактировалось - для этого при отправке новых данных в репозиторий (commit) есть смысл писать содержательные и осмысленные комментарии. Если это так, то можно многое узнать о том, что и когда редактировалось:

```
$ svn log --revision 1:3
```

что значит: выдать лог ревизий с первую по третью. Увидим:

```
-----
r1 | penta4 | 2008-06-03 15:49:21 +0400 (Втр, 03 Июн 2008)
    | 1 line Initial dissertation state

-----
r2 | penta4 | 2008-06-03 16:56:01 +0400 (Втр, 03 Июн 2008)
    | 3 lines Committed new editions

-----
r3 | penta4 | 2008-06-03 16:58:12 +0400 (Втр, 03 Июн 2008)
    | 2 lines incremental -----
```

Очень удобно, когда нужно кому-то отправить историю правок вашей программы или документа.

### 9.2.5 Просмотр изменений в Subversion

Например, нужно посмотреть изменения, происшедшие со второй по четвертую ревизии. Легко!

```
$ svn diff --revision 2:4 PhD_Konnik.tex >
  PhD_Konnik.tex.diff.r2-4
```

Это выведет нам в diff-файл все изменения, а ваш любимый текстовый редактор их с удовольствием подсветит для более удобного просмотра.

### 9.2.6 Отмена изменений в Subversion

Вы сидите и правите какой-то документ - и тут с ужасом понимаете, что правите *не тот* документ. Не беда - переносите изменения во временный файл, а для того документа, который вы нечаянно испортили, пишем:

```
$ svn revert
  UMNIK08-Report-01/Report-01-sub1-shortened.tex
Reverted
  'UMNIK08-Report-01/Report-01-sub1-shortened.tex'
```

После этого SVN вернёт файл Report-01-sub1-shortened.tex в то состояние, в котором он находится в репозитории.

### 9.2.7 Копирование или перемещение нескольких файлов в SVN/Subversion (multiple targets move/copy in SVN)

Странно, однако SVN не понимает символов типа \*, так что ваша попытка сделать так:

```
$ svn mv MCBC/*.jpg LinuxImages/
```

приведёт к ошибке:

```
svn: Client error in parsing arguments
```

Чтобы не перемещать все файлы по одному, воспользуемся Bash:

```
for i in *.cs ; do svn mv $i newFolder/ ; done
```

Конечно, если в вашей операционной системе есть нормальная консоль :-)



### 9.2.8 Выдача определённой правки файла из репозитория

Например, нам очень нужно посмотреть, каким был файл при второй правке. Для этого пишем:

```
$ svn cat --revision 2 PhD_Konnik.tex >
  PhD_Konnik.tex.v2
```

То есть выводим результат второй правки файла PhD\_Konnik.tex и записываем его в файл PhD\_Konnik.tex.v2 для дальнейшего просмотра.

### 9.2.9 Восстановление удалённых файлов в Subversion

Ну и наконец самое страшное: вы приказали Subversion удалить файл из репозитория, а потом кто-то вам говорит: этот файл нужно вернуть. На самом деле, файл, который вы удалили из SVN, никуда не делся: элемент исчезает из правки HEAD, но продолжает существовать в более ранних правках. Так что вопрос "как восстановить удалённый файл в Subversion" имеет довольно простой ответ: скопировать его из последней правки, где он был.

Главное вспомнить, как этот файл назывался. Если помним хотя бы частично имя файла (например, overview), то можно попросить Subversion вывести все упоминания о правках этого файла так:

```
svn log -v | grep spatialcoh -A 5 -B 5
```

Здесь параметры для grep такие: -A 5 это выдача 5 строк после (After) совпадения, -B 5 это 5 строк до (Before) совпадения. Получим длинный вывод вида:

```
spatialcoherencyexplained.eps 436 | penta4 |
2009-12-02 15:39:30 +0300 Ср(, 02 Дек2009) | 2 lines
Changed paths:
M /PhD_Konnik_autoreferat.tex
```

```

M /PhD_Konnik_chapter_opticalcoding.tex
M /PhD_Konnik_chapter_overview.tex
D /images-opticalcoding/spatialcoherencyexplained.eps
M /images-opticalcoding/sxemacorrelator3.eps
M /latex.py
M /svnstatus.tex
M /zagashnikPhD_Konnik.tex

```

```
--
```

```

M /biblio
A /formaldocs/disserDocs.tif.7z
A /images-opticalcoding/PSFdotssize.eps
M /images-opticalcoding/PSFoutofLambdaDelta
LambdaDavsfromNumberOfElementsPSFRus.ps
A /images-opticalcoding/auxiliaryImagesForDisser.svg
A /images-opticalcoding/spatialcoherencyexplained.eps
M /zagashnikPhD_Konnik.tex

```

TENGIBLE changes:

- autoreferat is improved slightly;
- Spatial coherency is improved

Красным подчеркнул я: видно, что файл `spatialcoherencyexplained.eps` удалён в правку 436, и теперь хотим его восстановить. Смотрим, в какой правке файл был удалён - в 436, значит, в 435 есть его последняя копия. Восстанавливаем:

```

svn copy --revision 435
file:///phdthesis/images-opticalcoding/spatialcoherencyexplained.eps
spatialcoherencyexplained.eps

```

Всё, теперь файл появится снова, вместе с историей правок (что важно) и вы можете его править дальше.

### 9.3 Вертикальная конкатенация двух файлов

Обработанные экспериментальные данные часто представляют собой ряды чисел, по которым затем необходимо строить графики. Иногда требуется склеить данные из двух источников так, чтобы новые данные были дополнительными столбцами в уже существующих данных.

Есть данные А:

```
$ cat RAWMAXmeasurementresult
2.7100000e+02 2.8300000e+02 2.8000000e+02
2.7300000e+02 2.8700000e+02 2.8400000e+02
2.7400000e+02 2.9600000e+02 2.9000000e+02
2.7800000e+02 3.0400000e+02 2.9800000e+02
```

Есть другой файл с данными В, полученный обработкой результата эксперимента в другой программе:

```
$ cat shutter
.00024414062500000000
.00031661600810536980
.00037652020030874656
.00048828125000000000
```

Нужно сделать так, чтобы в результате получился файл типа:

```
2.7100000e+02 2.8300000e+02 2.8000000e+02
.00024414062500000000
2.7300000e+02 2.8700000e+02 2.8400000e+02
.00031661600810536980
2.7400000e+02 2.9600000e+02 2.9000000e+02
.00037652020030874656
2.7800000e+02 3.0400000e+02 2.9800000e+02
.00048828125000000000
```

Так проще строить графики в гнуплоте, что в скриптах реализуется сравнительно просто.

Условия:

1. Ряды А и В можно перемещать, но нежелательно менять местами данные в ряде А.

2. Решение должно предполагать автоматизированную обработку (данных очень много, Эксель / ОпенОфис Calc отпадают).

3. Ряд В получается не напрямую, сделать всё в одной программе нельзя.

4. Разделители - пробелы (количество неважно).

5. Задача не надуманная.

Что требуется:

Скрипт/программа, делающая такую конкатенацию, максимально кроссплатформенная (никаких визуал васиков), ориентированная на \*nix-среду.

Решение: Изумительное по краткости и лаконичности: использование команды paste.

```
$ paste -d " " RAWMAXmeasurementresult shutter > sult
```

И вся любовь.

## 9.4 Как сделать бекап при помощи dd

Для того, чтобы сделать backup (бэкап, резервная копия) винчестера для последующего восстановления вовсе не нужны коммерческие, проприетарные утилиты - есть старый, как мир, dd. С помощью dd очень удобно и просто делать бекапы, и всё, что нужно сделать - просто правильно его использовать.

### 9.4.1 Приготовление

Загружаемся с любого LiveCD диска с Linux / BSD, залогиниваемся под root-аккаунтом. Убеждаемся, написав в консоли df, что ни один из тех разделов, что мы собираемся копировать,

не примонтирована на запись. Решаем, куда будем архивировать данные: это может быть другой раздел на винчестере, внешний жёсткий диск или по сети.

#### 9.4.2 Создаём резервную копию

В данном примере `/dev/hda` это винчестер, который архивируем, а `/dev/sda1` - это раздел внешнего винчестера или другая партиция, на которую архивируем, `hda.img.gz` - получившийся сжатый файл-образ.

##### Внешний винчестер или другой раздел диска

Монтируем устройство, куда мы будем архивировать данные

```
# mount -t vfat /dev/sda1 /mnt/sda1
```

Перегоняем данные с одного винчестера на другой. Так как данных много, полезно их сжать - и нисс-системы позволяют это делать "на лету". В примере использован GZip, но подойдёт и любой другой (например, bzip2 или p7zip). Для этого:

```
# dd if=/dev/hda conv=sync,noerror bs=64K | gzip -c >
  /mnt/sda1/hda.img.gz
```

Здесь происходит архивирование с одновременным сжатием, результат чего записывается в файл `hda.img.gz`

Параметр `"conv=sync,noerror"` указывает dd не читать блок, если при этом происходит ошибка, но записать в архивируемый файл-образ данные, чтобы размер блока не менялся.

Параметр `"bs=64K"` это размер блока в килобайтах - использование такого большого размера блока ускоряет процесс копирования. Параметр `bs` можно сделать ещё больше, а именно равным размеру кеша копируемого диска (например `bs=8m`), это сильно ускорит весь процесс.

Выходной поток передаётся по конвейеру для сжатия `gzip`, результат выводится в файл на другом диске.

## 2.2 Удалённо по сети

Всё то же самое, что в п.2.1, но с добавлением в конвейер SSH:

```
# dd if=/dev/hda conv=sync,noerror bs=64K | gzip -c | ssh
  -c blowfish user@hostname "dd of=hda.img.gz bs=64k"
```

Архивный файл появится в домашней директории пользователя user на машине hostname. Если сеть маленькая, лучше вместо hostname использовать просто ip-адрес. Опция -c blowfish заставляет SSH выбрать алгоритм шифрования Blowfish, который быстрее остальных шифрует данные.

Вариант с netcat, если данные копируются по локальной сети и шифрование не нужно:

```
# dd if=/dev/hda conv=sync,noerror bs=64K | gzip -c | nc
  192.168.1.1 3333
```

На удалённой машине должен быть запущен netcat:

```
# nc -l -p 3333 > hda.img.gz
```

Вариант с SAMBA, если копирование производится на оффтопичную машину: примонтировать удалённо сетевой ресурс

```
smbmount //192.168.1.100/shared /mnt/remoteshare -o
  username=USERNAME,password=PASSWORD
```

и далее всё то же, что и в п.2.1.

### 9.4.3 Восстанавливаем данные из бекапа

Пример тот же: /dev/hda это винчестер, который архивируем, hda.img.gz - архивный сжатый файл-образ, а /dev/sda1 - это раздел внешнего винчестера или другая партиция, с которого восстанавливаем данные.

### Внешний винчестер или другой раздел диска

Копируем и распаковываем файл-образ одновременно:

```
# gzip -x /mnt/sda1/hda.img.gz | dd of=/dev/hda  
conv=sync,noerror bs=64K
```

Все параметры в п.2 уже объяснены.

### Удалённо по сети

Сетевой вариант с восстановлением из файл-образа:

```
# dd if=hda.img.gz | ssh -c blowfish user@deadhost "gunzip  
-c | dd of=/dev/hda1 bs=64k"
```

Следует помнить, что в данном случае не следует использовать опцию `conv=sync,noerror` в опциях `dd` - если вы всё-таки включите эту опцию, в большинстве случаев это исказит записываемые данные (это заставляет `dd` не дожидаться данных, приходящих по сети, если целый блок недоступен).

## 9.5 Subversion - быстрый старт

Вы пишете программу на любом языке программирования или просто новую статью в латехе. В порыве вдохновения лихо правите текст, забывая о бекапах между изменениями. И тут вспоминаете, что часть текста вчера удалили зря. Резервных копий нет, а текст был хорош... И тут время ставить систему контроля версий - например, Subversion. О том, как поставить и настроить Subversion в Linux, написан этот пост.

### 9.5.1 Как быстро поставить и использовать систему контроля версий?

На самом деле, в этом нет ничего сложного. Итак, возьмите вашу латеховскую статью / скрипт / исходник программы и

попробуйте. Для того, чтобы быстро поставить Subversion (она же SVN) и воспользоваться всеми её благами, нужно выполнить несколько простых шагов.

### Ставим Subversion и оболочку для неё

Хотя с системой контроля версий удобнее работать через консоль, графическая оболочка не помешает (можно визуально наблюдать изменения, сравнивать правки и прочее). Ставим:

```
# aptitude install kdesvn svn
```

и все зависимости (здесь рассматривается svn v.1.4.2 из Debian GNU/Linux Etch). Через мгновение у нас уже будет установлена Subversion. Полдела сделано!

### Начало работы с Subversion - импорт файлов в репозиторий

Первым делом нужно создать новый репозиторий - простой, без особых затей и канделябров:

```
$ svnadmin create phdthesis
```

В текущем каталоге появится подкаталог phdthesis следующего содержания:

```
drwxr-xr-x 2 penta4 penta4 32768 2008-06-03 15:48 conf
drwxr-xr-x 2 penta4 penta4 32768 2008-06-03 15:48 dav
drwxr-xr-x 5 penta4 penta4 32768 2008-06-03 18:55 db
-r--r--r-- 1 penta4 penta4 2 2008-06-03 15:48 format
drwxr-xr-x 2 penta4 penta4 32768 2008-06-03 15:48 hooks
drwxr-xr-x 2 penta4 penta4 32768 2008-06-03 15:48 locks
-rw-r--r-- 1 penta4 penta4 229 2008-06-03 15:48
  README.txt
```



Красным подсвечены важные для нас каталоги: `conf` содержит конфигурационный файл репозитория (сейчас мы его немного допилим), а `db` содержит версии наших данных.

Наш репозиторий предполагается локальным, то есть доступа по сети к репозиторию нет и не надо. Тогда нужно конфиг немного допилить. Редактируем файл `phdthesis/conf/svnserve.conf` для того, чтобы у нас не было проблем с записью туда новых файлов - для этого ищем строку `anon-access` и правим:

```
anon-access = write
```

Теперь любой аноним имеет права на запись в наш репозиторий - то есть вам не нужно ломать себе голову авторизацией и паролями (если, конечно, с проектом работаете вы один).

Так, репозиторий создали, теперь его нужно наполнить данными. Для этого переходим в каталог, где хранятся нужные данные, и даём команду:

```
$ svn import -m "Initial dissertation state"
file:///mnt/flash/other/phdthesis/
```

Здесь указывается полный путь до репозитория. Каждый раз указывать его не нужно - скоро мы поднимем SVN-сервер одной левой, и наша жизнь станет ещё веселее.

При импорте файлов Subversion будет выводить в консоль имена добавляемых файлов:

```
Adding chapter.dtx
Adding kvoptions.ins
Adding (bin) draft_PhD_Konnik.dvi
Adding PhD_Konnik_chapter_introduction.tex
Adding PhD_Konnik_chapter_introduction.tex.backup
Adding (bin) PhD_Konnik_chapter_introduction.dvi
Adding PhD_Konnik.tex
```

```
Committed revision 1.
```

То есть команда `svn import` берёт содержимое текущего каталога и помещает его в репозиторий, а так же все вложенные каталоги, если они есть. При этом выводится номер ревизии - это 1, наша первая правка.

### Извлекаем и загружаем изменённые файлы в репозиторий

Для того, чтобы начать работать с файлами, переходим в какой-нибудь другой каталог и просим Subversion выдать нам последнюю копию из репозитория. Делается это так:

```
$ svn checkout file:///mnt/flash/other/phdthesis/
```

в результате чего в текущем каталоге получим локальную копию данных из репозитория.

После внесения нужных правок, загружаем результаты обратно в репозиторий, для чего выполняем команду в каталоге, где находится рабочая копия данных, с которыми работали:

```
$ svn commit --message "minor edition in tex-files"
```

После ключа `--message` пишем в кавычках короткое сообщение, чтобы отличать одну правку от другой. В процессе загрузки изменений Subversion будет выводить имена файлов, которые загружаются:

```
Sending PhD_Konnik.tex
Sending draft_PhD_Konnik.tex
Transmitting file data ..
Committed revision 2
```

Готово! Ревизия изменённых файлов зарегистрирована в репозитории. Вот так легко и просто начать работать с Subversion.

## Глава 10

# Цифровые камеры

### 10.1 Экспозиция более 30 секунд на цифровой камере: Long-time remote shooting

Задача: требуется фотографировать с выдержками более 30 секунд ("от руки") и есть необходимость этот процесс автоматизировать. Решение: при помощи паяльника, доступных электрических компонентов и скрипта на `bash` можно снимать с длительными выдержками без необходимости стоять с камерой и держать кнопку.

#### 10.1.1 Исходные данные

Имеется цифровая фотокамера Canon EOS 400D, Debian-powered ноутбук и необходимость снимать фотографии с выдержкой более 30 секунд. Если немного погуглить, то можно найти замечательную схему, которую предлагает Michael A. Covington [здесь](#). На всякий случай дублирую её тут:

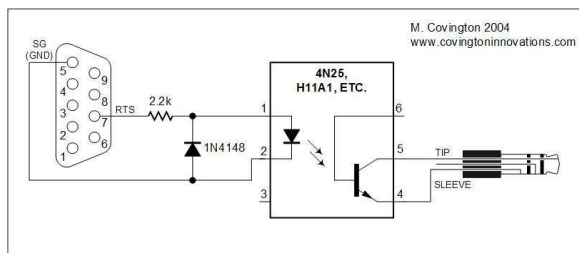
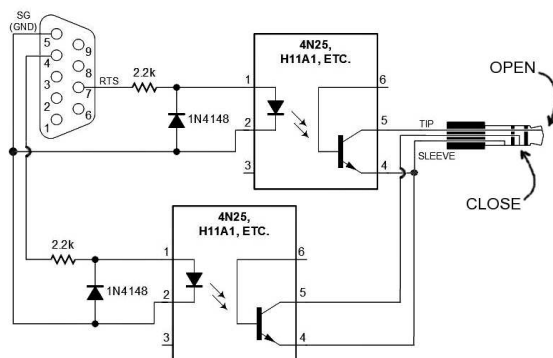


Схема прекрасная, но она подходит для Canon EOS 300D, а для 400D нет - если её спаять и попробовать управлять 400D, то затвор будет только подниматься, но не опускаться. То есть нет возможности остановить экспозицию в нужный момент, кроме как выключить камеру.

### 10.1.2 Схема для Canon EOS 400D

Долго ли, коротко, однако я с коллегой, Алексеем Ропяным, экспериментальным путём выяснили, что закрывает затвор фотоаппарат по сигналу с другого контакта, и Алексей предложил схему изменить вот так:



И это сработало: наш лабораторный Canon EOS 400D стал открывать и закрывать затвор по команде с компьютера!

### 10.1.3 Необходимые компоненты

Для того, чтобы спаять шнурок управления камерой, требуется некоторое количество четырёхжильного кабеля (от аудиоаппаратуры или простой телефонный провод), jack на 2.5мм (он же 3/32 inch), указанные на схеме радиоэлементы (можно купить в Chip&Dip), 9-контактный COM-порт и соответственно переходник USB-COM (чтобы можно было использовать и на современных компьютерах, и на старых), а также паяльник и немного терпения.

Переходник USB-COM лучше всего брать на чипе Profilic 2303 - это самый распространённый чип и в Линукс работает "из коробки".

### 10.1.4 Программное обеспечение

Для управления камерой требуется одна маленькая утилита, которую мне подсказали на Астрофоруме (обсуждение велось в теме "Выдержка более 30 сек. через комп?" тут и "Чем Linux полезен ЛА?" здесь).

Программа называется `setSerialSignal` и её исходник находится здесь. Скомпилировать её GCC, который входит в любой дистрибутив \*nix-систем:

```
$ gcc -o setSerialSignal setSerialSignal.c
```

Проверялось на системе Debian GNU/Linux v4.0 r.0 "Etch gcc version 4.1.2 20061115 (prerelease) (Debian 4.1.1-21).

### 10.1.5 Посылаемые сигналы

Программу собираем и даём ей необходимые права на выполнение, а ниже - сигналы, посылаемые в порт программой, и их передача `setSerialSignal`:

```
Сигнал DTR
setSerialSignal /dev/ttyS0 1 0
Сброс сигнала DTR
setSerialSignal /dev/ttyS0 0 0
Сигнал RTS
setSerialSignal /dev/ttyS0 0 1
Сброс сигнала RTS
setSerialSignal /dev/ttyS0 1 1
```

Всё это обеспечивает вам возможность управления камерой: открытие происходит по сигналу DTR, закрытие - сигналу RTS.

### 10.1.6 Скрипт управления камерой

Дальше удобно воспользоваться простым скриптом, который написал Евгений Ромас aka BrainBug, но для случая с 400D я скрипт немного поправил, и теперь он выглядит так:

```
#!/bin/bash

for i in `seq $3`; do
{
setSerialSignal /dev/ttyUSB0 0 0 &&
sleep $1 && setSerialSignal /dev/ttyUSB0 0 1 &&
sleep 0.3 && setSerialSignal /dev/ttyUSB0 0 0 &&
sleep $2 && setSerialSignal /dev/ttyUSB0 1 1 &&
    echo "One_more_image_captured!" &&
sleep $4;
}
done

echo "Done!"
```

Параметры скрипта:

- 1: задержка поднятия зеркала
- 2: время выдержки в секундах
- 3: количество кадров
- 4: задержка между кадрами

Пример:

```
$ make_captures 4 60 30 2
```

Скрипт рассчитан на работу с переходником USB-COM, и если у вас другой номер порта - естественно, скрипт нужно немного поправить.

### 10.1.7 Как это работает

Итак, шнурок спаян, переходник USB-COM включён в USB или сам шнурок вставлен в COM-порт, софт скомпилирован и наделён необходимыми правами. Изложение ориентировано на любую \*nix-систему, в данном случае это Debian GNU/Linux.

\* Включаем камеру, переводим её в режим BULB, выстав-  
ляем значение диафрагмы и ISO, а так же другие параметры  
при необходимости.

\* Вставляем jack в разъём фотокамеры, а противополож-  
ный конец в COM-порт или COM-USB переходник и далее в  
USB-порт компьютера.

\* Убеждаемся (в случае USB), что переходник опознан и  
работает: dmesg должен выдать что-то вроде:

```
usb 2-1: new full speed USB device using uhci_hcd and
address 2
usb 2-1: configuration #1 chosen from 1 choice
drivers/usb/serial/usb-serial.c: USB Serial support registered
for pl2303
pl2303 2-1:1.0: pl2303 converter detected
usb 2-1: pl2303 converter now attached to ttyUSB0
usbcore: registered new interface driver pl2303
drivers/usb/serial/pl2303.c: Prolific PL2303 USB to serial
adaptor driver
```

\* Всё готово, теперь снимаем при помощи команды:

```
$ make_capture 1 5 2 3
```

В этом примере будет сделано 2 кадра с экспозицией 5 се-  
кунд, задержка между кадрами 3 секунды, задержка поднятия  
зеркала 1 секунда.

Следует так же отметить, что таким способом можно от-  
секать выдержки от 1 секунды и больше по Exif. Физически  
передавать сигнал можно и быстрее, однако в Exif при этом  
пишутся неправильные данные. Точно так же, выдержки в  
данном случае отсекаются с точностью 1 секунда. Впрочем,  
для установки, которая должна отсекай длинные выдержки,  
это приемлемо.

Данное устройство с успехом применяется в оптическом  
эксперименте для съёмки изображений с длинной выдержкой.



Конкретно: применяется в оптико-цифровом корреляторе. Кроме того, может быть полезно астрономам-любителям для фотографирования астрономических объектов с длительной экспозицией.

Работает на камерах по крайней мере с firmware 1.0.4, имеются частные сообщения, что оригинальная схема должна работать с 400D. Выясняется в настоящее время.

## 10.2 Настройка веб-камеры Logitech QuickCam Messenger в Linux

Давно хотел приобрести подобное устройство (webcam), но было лень и не доходили руки. Пока мой давний друг не подарил мне сие чудо на день рождения. О том, как с полпинка настроить веб камеру в Linux, будет говориться здесь.

### 10.2.1 Подготовка

Несколько предварительных замечаний - если вы не собирали сами, можно пропустить эту часть. Нужно учесть, что видеокамера в Linux - такое же точно видеоустройство, как и TV-Tuner. Поэтому все замечания о настройке TV-Tuner в силе. Если у вас дистрибутивное ядро - там всё уже включено, а для любителей самосборных ядер нужно включить:

- \* v4l и все его протоколы (само v4l монолитом, протоколы - модулями)

- \* i2c и его протоколы (так же, фактически как при установке ТВ-тюнеров)

Если это в ядре есть, дальнейшее не составит проблем.

### 10.2.2 Установка

На коробке пишут: установка проста, как раз-два-три. В общем, у меня так и получилось:

### Устанавливаем программное обеспечение

Запускаем с балкона два прилагающихся диска и вместо этого ставим из репозитория:

```
$ sudo aptitude install gspca-source camorata
```

В каталоге `/usr/src` появится архив с исходниками `gspca-source.tar.bz2` - распаковываем его прямо там. После этого переходим в каталог `/usr/src/modules/gspca` и там выполняем команду:

```
# make
```

Модуль должен собраться без ошибок. Далее пишем:

```
$ sudo make install
```

Это установит наш модуль в каталог к другим модулям ядра. На всякий случай, можно после этого написать:

```
# sudo depmod -a
```

Всё, время загрузить модуль ядра для вебкамеры:

```
$ sudo modprobe gspca
```

Теперь всё готово. Врубаем камеру и запускаем софт.

### Подключаем камеру

Подключаем USBшный штекер камеры, она мигнёт зелёным светодиодом. На всякий случай смотрим выводы `dmesg`.

Диагностические выводы показывают, что чип опознан. Теперь должен появиться файл устройства `/dev/video0` - это и есть наша вебкамера. После этого запускаем программу по работе с камерами (её мы установили на предыдущем шаге):

```
$ camorata
```

Если всё было сделано верно, вы увидите видеоизображение в окошке программы и простой интерфейс управления изображением (яркость/контраст, насыщенность, фильтры и прочее).

## Глава 11

# Навигация

### 11.1 Закачка карт в GPS навигатор Garmin GPSMap 60CSx

Не так давно решил разориться на GPS-навигатор, так как по многим местам приходится мотаться и ходить по отечественным картам уже нет никакой возможности.

Итак, вариантов закачки карт в GPS-навигатор, собственно, два: через эмуляцию `/dev/ttyUSB0` по закрытому протоколу Garmin и по Mass Storage.

#### 11.1.1 Закачка карт в GPS-навигатор по протоколу Garmin

Этот вариант оказался самым жизнеспособным. Если ваше ядро собрано с поддержкой `garmin_gps`, то после подключения навигатора к USB штепселю ядро нам должно написать в логах `dmesg` так:

```
usb 4-1: new full speed USB device using uhci_hcd and
address 3
usb 4-1: configuration #1 chosen from 1 choice
drivers/usb/serial/usb-serial.c: USB Serial support registered
for Garmin GPS usb/tty
garmin_gps 4-1:1.0: Garmin GPS usb/tty converter detected
usb 4-1: Garmin GPS usb/tty converter now attached to
ttyUSB0
usbcore: registered new interface driver garmin_gps
drivers/usb/serial/garmin_gps.c: garmin gps driver v0.28
```

Выделенная строчка нам говорит о том, что у нас появился новый порт имени `/dev/ttyUSB0` куда мы будем закидывать наши карты.

Где взять карты для GPS-навигаторов Garmin? Это философский вопрос, и в одном посте на него не ответить. Но для начала можно воспользоваться открытыми картами OpenStreetMaps (после регистрации имеем возможность экспортировать нужные вам куски карт в XML формат).

Уже собранные карты для Garmin, на всякий случай, есть здесь. После распаковки будет много файлов, нам нужен только `008XXX.img`

### 11.1.2 Программы для закачки карт в GPS-навигаторы для Linux

Софта под Linux для закачки gps-карт в навигатор немного. Из опробованного мной могу сказать про sendmap. Скажу сразу: версия `sendmap17Lstatic`, которая `Stable version of sendmap with RS support only`, не работает у меня никак. При попытке отправить в навигатор карту ругается:

```
./sendmap17Lstatic /dev/ttyUSB0 -s57600 moscowregion.img
GPS SendMap & GetMap version 17, Kozicki@gps.chrisb.org

Checking input files
Preparing ...
```

```
Failure to communicate with GPS or no maps can be upload
into GPS
```

Зато работает новая версия, которая на сайте обозначена как Free sendMap20 rev 4.2 BETA for Linux with experimental USB support. С её-то помощью и удалось закачать карту, сделанную для московской области:

```
./sendmap20 -t/dev/ttyUSB0 -s9600 moscowregion.img
Detected : GPSMap60CSX Software Version 3.60
Available memory : 120 MB
Max number of maps: 2025
Final map size : 486 kB
Maps uploaded!
```

После этого карта появляется в списке карт навигатора (на всякий случай - у Garmin в устойстве карта может быть только одна, загрузка новой стирает предыдущую).

Параметры можно менять, например, изменить скорость зачатки (-s115200) и присвоить карте другое отображаемое имя (-mMOSCOW )

```
./sendmap20 -t/dev/ttyUSB0 -mMOSCOW -s115200
63240001.img
```

После этого навигатор карту видит и может с ней работать. Правда, на той карте географическая привязка была какая-то странная...

### Закачка карт в GPS-навигатор через Mass Storage

Если выбрать в меню навигатора этот вариант, он станет большой флешкой и вы сможете на него закачивать карты:

```
usb 4-1: new full speed USB device using uhci_hcd and
address 4
usb 4-1: configuration #1 chosen from 1 choice
scsi0 : SCSI emulation for USB Mass Storage devices
```

```
usb-storage: device found at 4
usb-storage: waiting for device to settle before scanning
scsi 0:0:0:0: Direct-Access Garmin 60x microSD 1.00 PQ: 0
        ANSI: 5
sd 0:0:0:0: [sda] 246016 512-byte hardware sectors (126 MB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Mode Sense: 23 00 00 00
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] 246016 512-byte hardware sectors (126 MB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Mode Sense: 23 00 00 00
sd 0:0:0:0: [sda] Assuming drive cache: write through
sda: sda1
sd 0:0:0:0: [sda] Attached SCSI removable disk
sd 0:0:0:0: Attached scsi generic sg0 type 0
usb-storage: device scan complete
```

По идее, монтируем `/dev/sda1` и наступает нам счастье. После этого, по идее, нужно создать каталог `garmin` и закачать туда вашу карту с названием `gmapsupp.img`

Но здесь есть нюанс: если вы это делаете в первый раз, закачать карты можно, но навигатор карт не видит. Это связано, как я понял, с регистром символов: каталог должен называться `Garmin` - так, как его создаёт программа `sendmap` при первой заливке карт.

## 11.2 Garmin 60CSx: технические особенности устройства

В этом посте мне хотелось собрать раскиданные там и сям разные технические особенности GPS-навигатора Garmin 60CSx. Кроме того, несколько замечаний о картах для Garmin 60CSx. Собственно, о них далее.

Где скачать карты для Garmin? Собственно, на торрентах или пойти и честно воспользоваться картами OpenStreetMaps. Уже готовые карты OSM по городам есть здесь. Или можно

отправиться на сайт OpenStreetMaps, зарегистрироваться там и скачать карту нужной области честно и бесплатно.

Для этого заходим на <http://www.openstreetmap.org/> и регистрируемся. Далее выделяем область карты, которую хотим экспортировать (для последующей закидки в навигатор), жмём на Экспорт и просим карту в OSM формате - через некоторое время получаем файл `map.osm`. Теперь скачиваем `mkmap` отсюда, как завещал ув.тов. jethex в своей бессмертной статье, оно на джаве и требует сановскую джаву версии старше 1.5. С её помощью компилируем в полученный OSM-файл в `img` формат. Получившийся файл заливаем в гармин с помощью `sendmap`.

Однако с картами OpenStreetMap есть одна проблема: не дают скачать сразу большую карту (я с жадности запросил у сервера векторную карту всей Москвы с областью). Если нужны большие карты, например, всей РФ, можно податься на [Cloudmade.com](http://cloudmade.com) и скачать оттуда. Этот сайт содержит уже скачанные карты местностей в формате OSM (которые мы потом перегоняем в `img` для Гармина) и собственно файл `IMG`, который можно закачать в навигатор.

### 11.2.1 Закачивание карт в Garmin 60CSx

По сообщениям разных осведомлённых орлов, во внутренней памяти (`internal`) может находиться до 3 карт, тогда как на флешке (`microSD`) - только одна. Хотя, как совершенно справедливо попенял многоуважаемый комментатор `diaword`, на самом деле, никто не мешает скачать несколько карт и потом собрать из них одну. Это можно сделать с помощью утилиты `Mkgmap`, а её ключик `-gmapsupp` позволяет собрать результирующую карту из нескольких отдельных и потом залить в навигатор. Файл `gmapsupp.img` является архивом из нескольких карт `*.img`, причем их можно как упаковывать, так и обратно распаковывать с помощью `sendmap`.

А такой финт, как закачка четырёх файлов карт сразу:

```
Gmapbmap.img -- базовая карта
```



```
Gmapsupp.img -- ваша карта
Gmapsup2.img -- другая карта
Gmapprom.img -- другая карта
```

на Garmin 60CSx не проходит, к сожалению.

Размер одной карты (img-файл) не может превышать 2Gb, хотя карта может состоять из кусков, число этих кусков не может быть больше 2025.

### 11.2.2 Память и подключение

Подключение происходит через miniUSB кабель как в режиме MassStorage, так и по специальному протоколу. И тот, и другой вариант поддерживаются в Linux.

В комплекте идёт карта microSD на 64Mb, максимальная поддерживаемая ёмкость карты 4GB micro SDHC.

## 11.3 Маршруты и треки с помощью GPS-навигаторов Garmin

Отслеживание перемещений по GPS (tracking) это довольно интересная и полезная возможность современных навигаторов. Например, вы пошли в поход или просто в парк, и долго блуждали, пока искали правильный путь - со включённым отслеживанием вы потом сможете дома посмотреть, как можно было бы срезать путь. Или, что тоже важно, по этим данным можно найти обратную дорогу.

Собственно, на навигаторе всё довольно просто: в зависимости от модели, трекинг включается одной-двумя кнопками (в Garmin GPSmap 60CSX - главное меню - Track - ON). После этого на экране должны оставаться точки, которые обозначают ваш путь по местности. Эти данные в абсолютных координатах хранятся в самом навигаторе, и скачать их оттуда можно в Linux довольно просто.

### 11.3.1 Скачиваем треки в гармине под Linux

Для этого нам понадобится консольная утилита `gpsbabel`. Её исходники нужно скачать с официального сайта (или посмотреть в репозиториях своего любимого дистрибутива) и собрать. Далее в нашем распоряжении появляется собственно `gpsbabel`, с помощью которого треки и скачиваем.

Можно загрузить трек и точки маршрута с устройства при помощи программы `gpsbabel`, из командной строки. Если устройство подключено к `/dev/ttyUSB0`, а нужен трек с точками в формате GPX, то получить его можно так:

```
$ gpsbabel -t -w -i garmin -f /dev/ttyUSB0 -o gpx -F
мойтрек.gpx
```

Другой способ получить эту информацию ? скопировать с устройства GPX-файл в режиме USB Mass Storage, а потом уже его конвертировать-преобразовывать... Этот файл с названием вида ГГГГММДД.gpx появляется на карточке памяти только если включена соответствующая функция в приборе.

## Глава 12

# Мультимедиа

### 12.1 МОСР

Время от времени нужно выходить в консоль и гасить иксы - для изменения иксовых же настроек или обновлений, связанных с иксовыми пакетами. И сидеть в это время в тишине совсем не хочется. Конечно, можно запустить XMMS на ноутбуке и проигрывать музыку оттуда, но есть решение лучше: МОСР.

Управляется горячими клавишами, и основные клавиатурные сокращения такие:

1. ENTER - переключиться в директорию и/или начать воспроизведение песни
2. o - проиграть URL (например, интернет-радиостанции)
3. p или пробел - пауза
4. s - стоп
5. l - переключиться между одно- и двухпанельным режимом
6. a - добавить текущий файл в список воспроизведения

7. A - добавить текущую директорию в список воспроизведения
8. C - очистить список воспроизведения
9. q - отключить консольный интерфейс (сервер МОС будет продолжать воспроизведение, к нему можно подключиться снова, набрав тоср)
10. Q - отключение МОС-сервера (прекращение воспроизведения)

Так же можно воспользоваться более широкими возможностями по управлению плеером с помощью следующих клавиатурных сокращений:

1. n - проиграть следующую песню из списка
  2. b - проиграть предыдущую песню из списка
  3. S - воспроизведение в произвольном порядке
  4. R - повторяет выбранную песню циклически
  5. u - перемещает элемент списка воспроизведения вверх
  6. j - перемещает элемент списка воспроизведения вниз
  7. g или / - ищет заданную строку в именах фалов песен
  8. r - перечитать директорию заново
  9. f - переключить режим отображения с имён файлов на теги
  10. ТАВ - переключение между списком воспроизведения и файлами песен
  11. Н - показывает скрытые файлы
  12. Ctrl-t - показать или убрать длительность песни в списке
  13. Ctrl-f - показать или убрать формат песни
  14. U - переместиться на уровень выше в директории
  15. a - добавить файл к списку воспроизведения
  16. A - добавить каталог к списку воспроизведения
  17. C - очистить список воспроизведения
  18. V - сохранить список воспроизведения
  19. d - удалить отмеченные песни из списка воспроизведения
- Наконец, управление громкостью

1. < - уменьшить громкость на 1%
2. , - уменьшить громкость на 5%
3. >- увеличить громкость на 1%
4. . увеличить громкость на 5%

Если вы запомнили какую-нибудь команду, то получить справку по клавиатурным сокращениям можно клавишей "?"

Язык осин и берёз МОСР по умолчанию отображает славяно-греко-латинской мешаниной, что легко поправить. Для этого копируем пример конфига отсюда:

```
$ cp /usr/share/doc/moc/examples/config.example.gz ~/.moc/
$ cd ~/.moc
# chown $USER config.example.gz
```

Где **\$USER** - имя пользователя в системе. После этого разархивируем конфиг:

```
$ gunzip config.example.gz
$ mv config.example config
```

И правим его своим любимым текстовым редактором. Чтобы заставить МОСР дружить с кириллицей в ID3v1-заголовках, необходимо найти строку:

```
#ID3v1TagsEncoding = WINDOWS-1250
```

и превратить ее в

```
ID3v1TagsEncoding = WINDOWS-1251
```

И всё, играй, гармонь!

## 12.2 Немного о lossless кодировании видео в Linux

Некоторое время назад была потребность в том, чтобы сжать видеофайл без потерь в Linux (lossless compression) и воспроизводить сжатый фильм без пропуска кадров.

### 12.2.1 Сжимаем видео с помощью Mencoder

Для этого мы отобрали пару кодеков с lossless и начали сжимать наш видеоролик. Исходно в нашем распоряжении был видеоролик размером 341 Мб, в котором полутонная картинка перемещалась по экрану. Нужно было сжать без потерь ролик так, чтобы он и места занимал немного, и воспроизводился без торможений.

#### Сжатие FFV1

Сначала сжимали кодеком FFV1 командой:

```
$ mencoder -ovc lavc -lavcopts vcodec=ffv1 video-before.avi
-o video-compressed-ffv1.avi
```

Сжалось до 18Мб, что очень хорошо. Однако при попытке такой сжатый файл воспроизвести, возникли проблемы: Mplayer (как и другие) сильно грузит процессор и видео не успевает воспроизводиться с заданным FPS. О том, какие страдания были с воспроизведением и как мы их героически решали, будет сказано ниже.

#### Сжатие huffyuv

Ещё один метод сжатия без потерь, но жмёт он хуже, чем FFV1, зато быстрее воспроизводится и меньше грузит процессор. При этом использовали mencoder для сжатия huffyuv с параметрами:

```
$ mencoder -ovc lavc -lavcopts
vcodec=ffvhuff:vstrict=-1:pred=2:context=1:format=422p
video-before.avi -o video-compressed-ffvhuff.avi
```

Получилось видео на 99Мб, что тоже неплохо. Однако при воспроизведении оказалось, что происходит конвертирование в другое цветовое пространство 420p и мы решили отказаться от такого финта:

```
$ mencoder -ovc lavc -lavcopts
vcodec=ffvhuff:vstrict=-1:pred=2:context=1
video-before.avi -o video-compressed-ffvhuff-420.avi
```

при этом сэкономили ещё почти 20 Мб - фильм стал занимать 80Мб и воспроизводиться чуть глаже.

Остановившись на сжатии huffyuv без опции `format=422p`, мы стали перебирать варианты воспроизведения видео с минимальными тормозами. История на этом не кончается...

### 12.2.2 Воспроизведение видео с помощью Mplayer

Видео у нас сжато слабо, воспроизводить надо без задержек и железо скажем сразу весьма чахлае. Как проигрывать видео в таких условиях?

Перво-наперво, используем замечательные возможности Mplayer по выводу видео куда и на что угодно: поэкспериментировать с параметром `-vo` было здоровой идеей, и через некоторое время мы остановились на варианте `-vo gl` и `-vo sdl` для нашей задачи. Однако иксы - это хорошо, но хочется выводить видео на чёрный экран без всяких засветок.

Далее, чтобы отвязаться от дисковой подсистемы как можно больше, используем возможность Mplayer кешировать видео:

```
$ mplayer -vo gl -cache 65535 -cache-min 98
video-compressed-ffvhuff.avi
```

Это значит, что мы просим сделать Mplayer кеш на 64Мб и заполнять его не менее чем на 98%. Это сильно помогло делу, но это ещё не всё.

Условия нашей задачи в меру экзотические: нужно проигрывать видео без масштабирования, "как есть и желательно на чёрном экране без фона, с максимально возможной плавностью. Всё осложняется тем, что компьютер, на котором видео

будет воспроизводиться, управляется Windows, и Linux там можно задействовать только в режиме LiveCD.

Идём дальше и привлекаем для наших нужд Knoprix. В арсенале его параметров загрузки есть неприметный параметр fb1280x1024 для работы с framebuffer. Замечательное дело, и мы решили этим воспользоваться. У нас теперь появилась графическая консоль, чёрная и мрачная, как полярная ночь.

Теперь, чтобы воспроизвести сжатое нами видео, копируем видео в память на Knoprix в каталог Desktop. После этого воспроизводим наше многострадальное видео в консоли так:

```
$ mplayer -vo fbdev -cache 65535 -cache-min 98 -quiet
video-compressed.avi
```

Ключик -quiet нужен для того, чтобы при воспроизведении Mplayer не выводил статистику в бегущей строке. Всё, после этого получаем фильм на чёрном экране, который воспроизводится плавно и без затей.

### 12.2.3 Скорость воспроизведения в Mplayer

На всякий случай, если потребуется запускать фильм с другой скоростью, всегда можно попросить Mplayer изменить число кадров в секунду (вплоть до 1 кадра в секунду) так:

```
$ mplayer -fps 1 video-compressed.avi
```

Эта команда будет воспроизводить фильм со скоростью 1 кадр в секунду.

## 12.3 Видеозахват и сжатие видео в Linux

С появлением устройств видеозахвата возникает необходимость в программном обеспечении, которое способно записывать видео или отдельные кадры через определённый промежуток времени. Всё это может быть решено открытыми программами, входящими в Debian Linux.



### 12.3.1 Запись видеопотока с устройств захвата на примере ТВ Тюнера

Большинство программ просмотра телепередач, например tvtime, умеют только просматривать программы и снимать скриншоты. Но время от времени требуется и записывать передачи, особенно когда они идут в неподходящее время. Для этого, в соответствии с UNIX way, есть другие утилиты. Одна из них называется streamer.

```
# aptitude install streamer
```

После того, как пакет streamer будет установлен, он сможет писать и снимать скриншоты из /dev/video0 и /dev/dsp в картинки jpeg, png и в видеофайлы со звуком. И ролики пишет, и скриншоты делает (в том числе несколько через равные промежутки времени). Программа, естественно, консольная, так что завернуть её в скрипт труда не составит.

Для подробной справки у streamer есть очень хороший ключик, который называется -h. Справка написана очень толково, с примерами и простым языком. Программа пишет в файл заданного имени данные с указанного тюнера, в формате raw и с разными типами компрессии, с заданным числом кадров в секунду и проч.

Вот, например, мой скрипт на bash:

```
#!/bin/sh

if [ -z $1 ]; then
    kmvTimeToRecord=50;
else
    kmvTimeToRecord=$1;
fi

streamer -o ~/movie.avi -r 24 -f mjpeg -j 90 -s 720x520 -F
mono8 -t $kmvTimeToRecord:00 -p2
```

Скрипт пишет в файл `movie.avi` 120 минут телепрограммы с тюнера (если время не передано скрипту явно) 24 кадра в секунду с `mpeg`-компрессией и в полном формате 720x520 и восьмибитным звуком. Включение скрипта в нужное время осуществляется командой `at`, позволяющей выполнять любые задания в указанное пользователем время. Это ещё раз иллюстрирует мощь UNIX-way: несколько мелких программ можно связывать в цепочку для получения нужного результата.

### 12.3.2 Сжатие видеоданных

Если вы будете захватывать в `jpeg`-сжатии для экономии места, качество будет посредственным, а если в `rawvideo rgb15`, то минута видео займёт гигабайт. И тут потребуются сжать видеопоток... Естественное желание - всё это как-то скомпрессировать. Есть несколько программ данного класса: например `ffmpeg` и `mencoder`.

### 12.3.3 Пример с `ffmpeg`

Установим `ffmpeg` из стандартного дистрибутива Дебиан. После её установки у нас появляется возможность задавать множество параметров для пережатия захваченного видео с тюнера в MPEG4 с хорошим сжатием и неплохим качеством. Дабы не утомлять читателя `man ffmpeg` (а он там действительно впечатляющий), привожу свой башевский скрипт, настройки которого более или менее разумны:

```
#!/bin/sh
ffmpeg -i movie.avi -f avi -vcodec mpeg4 -b 800 -g 300 -bf
2 new.avi
```

Этот скрипт вызывает `ffmpeg` и указывает сжать файл `movie.avi` в новый, `new.avi`, с кодеком `mpeg4` и хорошим качеством. Пережатие - долгая операция, так что можно за это время выпить чашечку любимого напитка или даже принять ванну.

### 12.3.4 Пример с mencoder

Входящая в комплект вместе с Mplayer, утилита mencoder так же очень здорово сжимает видеоданные, в том числе с ТВтюнера. Вот пример с настройками, дающие удовлетворительное качество видеоизображения для просмотра:

```
$ mencoder movie.avi -o new.avi -oac copy -ovc lavc
-lavcopts vcodec=mpeg4:mbd=1:vbitrate=1800 -ofps 24
```

При этом перекодировается входной файл movie.avi в новый файл, new.avi с визуально удовлетворительным качеством. Чтобы каждый раз не вбивать параметры в командную строку добавил профили в mencoder.conf, например так:

```
[ppc]
profile -desc="ppc"
vf=scale=320:240
oac=mp3lame=1
lameopts=mode=3:cbr=1:br=48
ovc=lavc=1
lavcopts=vcodec=mpeg4:vbitrate=270
fourcc=DIVX
```

Для уменьшения размера с точными пропорциями можно vf=scale=320:240 заменить на

```
vf=scale=1
zoom=1
xy=320
```

Теперь видео будет иметь ширину 320, а высота будет прочитана автоматически с соблюдением пропорций.

## 12.4 Установка Mplayer в Linux

Собрать Mplayer в Linux на примере Debian Linux и установить кодеки для Mplayer.

### 12.4.1 Подготовка

Идём на официальный сайт и скачиваем там архив с программой и, кроме того, архивы с кодеками (они обычно называются `essential*.tar.bz2` или вроде того). После скачивания всего добра оттуда готовим систему для установки: нужен компилятор и несколько дополнительных пакетов. Для сборки плеера из исходников нам нужно поставить пакеты:

```
# apt-get install gcc g++ make
```

После этого ставим необходимые для MPlayer пакеты.

Некоторые пакеты могут быть уже установлены, другие потребуют обновлённых версий, но набор их остаётся прежним. Далее переходим в каталог `/usr/src` и создаём там каталог `mplayer`, где и будем его собирать. Всё почти готово...

### 12.4.2 Сборка MPlayer в Debian Linux - консольный плеер

Переходим в каталог с исходниками MPlayer и разархивируем их в каталог:

```
# cd /usr/src/mplayer/ && tar -jxvf * && cd *
```

После этого в этом же каталоге набираем:

```
# ./configure
```

Поползуют выводы конфигулятора: что у вас есть и с поддержкой чего будет компилироваться MPlayer. Если его на этом этапе что-то не устроит - он вылетит и перед кончиной скажет, чего ему не хватило для полного счастья. Недостающее надо поставить, после чего собрать-таки пакет прописыванием следующего заклинания:

```
# DEB_BUILD_OPTIONS="--disable-runtime-cpu-detection" fakeroot debian/rules binary
```

Вот прямо так и пишите, как здесь, и должно быть вам счастье. По экрану поползут и замелькают строчки компилятора, в которые вчитываться не обязательно. Апофеозом его деятельности будет появившийся в каталоге `/usr/src/mplayer` заветный файл `mplayerVersion.deb`, который мы ставим утилитой `dpkg` следующим манером:

```
# dpkg -i mplayerVersion.deb
```

Но это ещё не всё: у нас только основные кодеки, а дополнительных нет, и разные неправильные ролики мы не увидим. А хотим. Но разве напрасно мы качали кодеки к MPlayer вида `all-*.tar.bz2` и `essential-*.tar.bz2`? Сейчас мы их поставим.

### 12.4.3 Установка Mplayer кодеков в Linux

Отвечаю на вечный вопрос убунтуйцев и сусеводов - как поставить кодеки Mplayer. Скачиваем архив с кодеками Mplayer, разархивируем кодеки Mplayer и все их сваливаем в каталог `/usr/lib/win32` - если такого каталога нет, создаём его. Теперь (заделываясь рутом или его полномочиями через `sudo`) идём в каталог `/etc` и редактируем там файл `ld.so.conf` вписыванием в него строк:

```
/usr/lib  
/usr/lib/win32
```

Вот как например этот файл выглядит у меня (это только пример!)

```
# cat /etc/ld.so.conf  
  
/usr/X11R6/lib  
/usr/lib/atlas  
/usr/lib  
/usr/lib/win32
```

Вот только после этого действия у меня появились кодеки - и увидел их не только MPlayer, но и Xine, что приятно вдвойне. Так что теперь кодеками Mplayer будут пользоваться оба плеера.

#### 12.4.4 Использование консольного MPlayer - основные команды

Сейчас вы освоите такую экзотику, как консольный плеер. Это просто и при некотором навыке гораздо быстрее графического. Организую это в виде FAQ:

##### Как посмотреть фильм, который находится в каталоге /home/vasya/porno.avi?

В графической консолюке (Терминал, xterm или как она у вас называется) набираем: `mplayer /home/vasya/porno.avi`

##### Фильм идёт, но много жрёт процессорного времени

(вариант: только в центральной части экрана, а когда разворачиваешь fullscreen - фильм продолжает идти не на весь экран) - что можно сделать? Попробовать другие методы видеовывода, указав плееру параметр: `mplayer -vo xv /home/vasya/porno.avi` И попробовать другие ключи:

```
-vo sdl
-vo xv -vo x11
-vo gl
-vo gl2
```

а так же другие, которыми плеер поделится, если его попросить: `mplayer -vo help`

**Фильм идёт рывками, пропуск кадров виден нетрезвым/невооружённым глазом, что делать?**

а) попробовать предыдущий рецепт (п.2) и поменять параметр видеовывода, особенно помогает `sdl`; б) указать параметр `-framedrop` или `-hardframedrop` при которых плеер пытается воспроизводить фильм в максимально пристойном качестве с пропуском кадров; в) подымать `framebuffer`, выгружать иксы и смотреть фильмы в чистой консоли (для упёртых фанатов) или г) обновить своё железо.

**Хочется выводить фильм на второй монитор (который в режиме `clone/dualhead`), а оно играет только на одном мониторе - куда копать?**

В тот же пункт.2 и менять видеовывод указанием параметра видеовывода. Рецептов тут дать нельзя - только перебором возможных вариантов. Например, для ЗлаХП и видеокарты ATi RadeonX300 одновременный вывод на двух мониторах наблюдался при указании: `mplayer -vo gl /home/vasya/rogn.avi` Вывод фильма был, а ЗлоМедиаПлеер нервно курил в стороне :-)

**Хочу подкрутить яркость/контраст/насыщенность, а на какие кнопки жать в консольном плеере?**

На клавиатурные, натурально, о чём, кстати, в `man mplayer` подробно растолковано. Но для тех, кто мануалов не читает, привожу перевод:

```
Кнопки 1 и 2 понижают/повышают контраст
Кнопки 3 и 4 понижают/повышают яркость
Кнопки 5 и 6 понижают/повышают оттенки
Кнопки 7 и 8 понижают/повышают насыщенность
```

Не все эти операции поддерживаются различными методами видеовывода.

### **Видео и звук рассинхронизируются - как компенсировать?**

Снова мучаем клавиатуру:

```
кнопка + и - компенсировать назад/вперёд  
десинхронизацию на 0.1 секунд  
или прописать параметр -autosync 0
```

### **У видеофайла неправильное соотношение сторон (не правильный aspect ratio) - как компенсировать?**

В консоли запускаем mplayer с параметром -monitoraspect X:Y где вместо X и Y подбираем правильные соотношения экспериментально. Полная строка может выглядеть так: mplayer -monitoraspect 4:4 1984.mpg

## **12.5 Мультимедийные клавиши и кнопки в Linux**

Многие современные клавиатуры комплектуются разного рода мультимедийными клавишами и мультимедийными кнопками для управления плеерами и броузерами.

### **12.5.1 Коды мультимедийных клавиш**

Прежде, чем мы начнём настраивать мультимедийные кнопки, нужно узнать, видят ли иксы эти самые мультимедийные клавиши - потом нам эта информация очень пригодится. Сделать это просто, стандартными средствами иксов - программой xev.



Запускаем консольку xterm (или другую, которая больше нравится), пишем в ней хев (от пользователя). Появится белое окошко, которое будет регистрировать всё, что происходит с мышью (в нём можно перетаскивать чёрный прямоугольник, но это детали). Пробуем нажимать клавиши и смотрим, что получается. Например, жмём на кнопку Down, должны увидеть вот что:

```
KeyRelease event, serial 32, synthetic NO, window 0xe00001,
  root 0x4c, subw 0x0, time 8095640, (-57,293),
  root:(753,322), state 0x0, keycode 104 (keysym 0xff54,
  Down), same_screen YES, XLookupString gives 0 bytes:
```

То, что написано после keycode, и есть код клавиши (выделение - моё). Пробуем нажать и отпустить мультимедийную кнопку - если видим какой-нибудь keycode, радуемся. Осталось только поставить соответствующий софт, чем и займёмся.

### 12.5.2 Софт для использования мультимедийных кнопок

Чтобы зашедшие сюда ассы меня не закидали тухлыми овощами, скажу сразу: да, есть метод, который позволяет подключить горячие клавиши через правку xmodmaprc - но я пугать новичков им не буду.

Итак, в Debian Linux есть единственная программа, которая позволяет комфортно использовать в Иксах мультимедийные кнопки:

```
hotkeys — A hotkeys daemon for your Internet/multimedia
  keyboard in X
```

Ставим:

```
# apt-get install hotkeys
```

Отлично, теперь запускаем её, указывая параметр клавиатуры:

```
$ hotkeys -t btc8190
```

Предопределённых конфигов довольно много, но они друг от друга отличаются только тем, какие коды клавиш определены для каких действий. Поэтому идём в каталог `/usr/share/hotkeys` и смотрим, какой конфиг (по содержимому) нам ближе. Его можно и нужно редактировать под свою клавиатуру и потребности. На код клавиши можно поставить всё, что угодно. Например, мне удобно управлять только плеером с клавиатуры (у меня Dialog KP-105UH), поэтому я напярк хев на предмет кодов клавиш мультимедиа, поправил конфиг и вот что получилось: А вот собственно текст конфига:

```
<?xml version="1.0"?>
<definition>
<config model="BTC_Smart_Office_(8190)">
<NextTrack keycode="153"/>
<PrevTrack keycode="144"/>
<Play keycode="162"/>
<Stop keycode="164"/>
<VolUp keycode="176" adj="2"/>
<VolDown keycode="174" adj="2"/>
<Mute keycode="160"/>
<userdef keycode="231"
  command="/usr/bin/myscript/cdem">CDchange</userdef>
<userdef keycode="223"
  command="/usr/bin/myscript/cdem">CDchange</userdef>
<userdef keycode="222"
  command="/usr/bin/myscript/cdem">CDchange</userdef>
</config>
<contributor>
```

```
<name>Petrisor Eddy Marian</name>  
<email>marian.petrisor@siemens.com</email>  
</contributor>  
  
</definition>
```

### 12.5.3 Другие программы использования мультимедийных кнопок в Linux

На этом история не заканчивается - есть и другие приёмы работы с мультимедийными кнопками. На никс-форуме советовали для борьбы с клавиатурами обратить внимание на следующие программы: 1) keytouch - это, думаю, лучшее, что есть, но, возможно, вы не найдёте среди поддерживаемых свою клавиатуру; 2) sven - глючная, собирать из исходников надо, когда устанавливаешь. Правда, включает в себя многие вкусности, которые у меня не заработали. 3) xmodmap - подключаем клавиатуры через xmodmap /xmodmaprc, в /xmodmaprc в свою очередь прописываем сопоставления функций кодам клавиш, все замечательно. Один из подходов к использованию мультимедийных кнопок с использованием xmodmap дан здесь.

## Глава 13

# Изображения

### 13.1 Чтение МАТ-файлов MATLAB в `pir2`

Теперь замечательный графический анализатор `pir2` полностью поддерживает чтение МАТ-файлов, в которых MATLAB записывает данные рабочего окружения с двойной точностью (`double precision`).

#### 13.1.1 Для чего нужно открывать МАТ-файлы в `pir2`

Упреждая возможные вопросы "а зачем нам проприетарный `mat` когда есть `csv`" отвечаю: запись в открытый формат CSV в матлабе происходит несколько нетривиальным образом и можно записать только одну переменную.

#### 13.1.2 Просмотр МАТ-файлов MATLAB в `pir2`

Здесь всё довольно просто: в диалоге открытия явно указываем, что хотим просмотреть MATLAB's `mat-files` и далее откры-

ваем такой файл. Пример на скриншоте - легендарная Лена.

А теперь примерчик посложнее - загружаем изображение усреднённых темновых шумов цифровой камеры и берём от них Фурье-преобразование (на том же скриншоте открыта в левом нижнем углу неудачная фотография от скоростной съёмки):

Так что наши данные теперь можно просмотреть без томительного ожидания, пока MATLAB откроет эти файлы. Кстати о данных в форматах CSV и MAT...

### 13.1.3 Данные в CSV и MAT-форматах: особенности записи MATLAB и просмотр `pip2`

Эту заметку я хотел сделать отдельным постом, но решил выложить здесь же. Появился тут вопрос: как сохраняет MATLAB данные с плавающей точкой в разных форматах. Для этого мы провели маленькое расследование...

#### 13.1.4 Подготовка

Ответ состоит из нескольких частей. Да, Матлаб может сохранять данные в своём формате `mat`, и его может прочесть та же Octave (теперь и `pip2` с использованием библиотеки `matio`). Для проверки сгенерируем матрицу

```
>> a = magic(5)

a =

17 24 1 8 15
23 5 7 14 16
4 6 13 20 22
10 12 19 21 3
11 18 25 2 9
```

```
>> b = a./3

b =

5.6667 8.0000 0.3333 2.6667 5.0000
7.6667 1.6667 2.3333 4.6667 5.3333
1.3333 2.0000 4.3333 6.6667 7.3333
3.3333 4.0000 6.3333 7.0000 1.0000
3.6667 6.0000 8.3333 0.6667 3.0000

>> b(2,2)=1.09878432753084573248523475
>> b(2,3)=2^20
```

Видим только первые четыре числа после запятой - это потому, что по умолчанию MATLAB отображает числа в коротком формате. Изменить этот формат можно командой `format type`.

Чтобы сделать задачу более наглядной, в матрицу записали два числа: одно с разными числами после запятой, а второе - очень большое ( $2^{20}$ ). Теперь сохраняем в MAT-файл и CSV командами:

```
save('magic5x5doubleafterformatlong.mat', 'b');
    ЭтоСохраняетсяфайл MAT
csvwrite('magic5x5double-afterformatlong.csv',b);
    ЭтоСохраняетсяфайлCSV
```

Теперь открываем их и сравниваем с исходниками...

### 13.1.5 Данные в MAT-файле.

В MAT-файлах всё чисто: он сохраняет так же, как и открывает.

```
>> load('magic5x5doubleafterformatlong.mat')
>> b
b =
```

```

1.0e+06 *
Columns 1 through 4
0.00000566666667 0.00000800000000 0.00000033333333
    0.00000266666667
0.00000766666667 0.00000109878433 1.04857600000000
    0.00000466666667
0.00000133333333 0.00000200000000 0.00000433333333
    0.00000666666667
0.00000333333333 0.00000400000000 0.00000633333333
    0.00000700000000
0.00000366666667 0.00000600000000 0.00000833333333
    0.00000666666667
Column 5
0.00000500000000
0.00000533333333
0.00000733333333
0.00000100000000
0.00000300000000
>> b(2,2)
ans =
1.09878432753085
>> b(2,3)
ans =
1048576

```

### 13.1.6 Данные в CSV-файле.

А вот тут начинаются приключения. MATLAB записывает в CSV формат данные с точностью, такой же, как при отображении. То есть, например, при отображении используется вывод только первых 4 чисел после запятой - следовательно, сохранены они в CSV будут так же.

```

>> b(2,2)

ans =

```

```
1.0988
```

С большими числами тоже самое: они сохраняются в экспоненциальной форме с точностью, установленной для отображения (по умолчанию 4 знака). Это так же видно в CSV-файле (записывает с той точностью, с которой отображает). У одной и той же матрицы, сохранённой в CSV и MAT сходный размер.

```
>> b(2,3)=2^20

b =

1.0e+06 *

0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 1.0486 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000

>> b(2,3)

ans =

1048600

>> 2^20

ans =

1048576
```

Именно так оно и происходит по умолчанию: MATLAB просто округляет числа при записи в CSV до четырёх значащих чисел. Для того, чтобы запись происходила с нужной нам точностью в CSV, следует использовать команду `dlmwrite` и явно



указывать точность. Например так:

```
>> dlmwrite('magic5x5double1—dlmwrite—precision12f.csv', b,  
            'precision', '12.12f')
```

Это записывает матрицу в CSV формат с переменной `b` и обеспечивает точность 12 значащих чисел. Так что стоит аккуратнее обходиться с командами записи: `csvwrite` вам запишет, конечно, переменную, но с малой точностью. Это может стать источником трудноуловимых проблем.

### 13.1.7 Отображение данных в MAT и CSV форматах в `nir2`

Теперь, возвращаясь к теме поста, в последних версиях (`nir2` и `vips > 7.18.2`) замечательного графического анализатора `nir2` можно просматривать не только данные в CSV, но и в MAT-файлах. Это очень и очень удобная возможность: например, усреднённое изображение в MAT-файле можно проанализировать и посмотреть без необходимости загружать лишний раз MATLAB. Особенно это актуально, если изображение в MAT-файле больших размеров. Точность просмотра CSV и MAT-данных составляет 5 значащих цифр.

## 13.2 Установка и впечатления новой версии графического анализатора `NIP2`

Не так давно вышла новая версия библиотеки обработки изображений `VIPS`, а вместе с ней и графическая оболочка `nir2`. Несколько слов о том, что изменилось и как новую версию быстро собрать и использовать.

### 13.2.1 Фильтры

В составе `Nir2` теперь улучшенные фильтры морфологического анализа (пока только бинарная морфология), многие филь-

тры получили возможность точной настройки (custom kernels), а так же включён шумоподавляющий фильтр GREYstoration, основанный на анизотропной диффузии. GREYstoration интересен тем, что реализует подавление шума при помощи весьма изощрённого "сглаживания" изображения. В результате это важные детали не размываются, а шума становится намного меньше. Техника anisotropic diffusion известна довольно давно, однако реализация в Nip2 достаточно быстрая и нетребовательная к памяти, что важно при работе с большими изображениями.

### 13.2.2 Именованние рядов

Это полезная функция, если требуется анализировать много изображений. Тогда можно задать один workspace и просто заменять одно изображение - остальное пересчитается автоматически. Именованные ряды позволяют проще отличать конечные результаты анализа от промежуточных подсчётов. Просто впишите в пустую клетку:

```
Mean=Math_stats_item.Mean_item.action A1
```

и вы получите именованную ячейку, содержимое которой есть среднее значение изображения в ячейке A1. Именовывать по-русски пока нельзя, только на английском.

### 13.2.3 Горячие клавиши на элементы меню

Очень и очень полезная возможность! Открываем меню, доходим до нужной нам функции, подсвечиваем её и нажимаем к примеру CTRL+M - и теперь на функцию повешена горячая кнопка. Теперь, чтобы посчитать среднее значение на изображении, просто выделяем изображение и нажимаем CTRL+M.

### 13.3 Продвинутое средства анализа изображений в pir2

Некоторое время назад я уже писал об этом замечательном графическом редакторе pir2 для огромных изображений - сейчас я хочу записать методы обработки изображений в нём. По роду текущей деятельности приходится иметь дело с 12-битными изображениями (конвертированные RAW-файлы при помощи dcrw в полностью документальный режим), так что просматривать и работать с такими картинками в обычных редакторах (типа GIMP или Krita) не удобно. Зато в pir2 можно и просматривать изображения любой битности, и выполнять весьма изощрённые методы обработки. Об этом ниже и будет говориться.

#### 13.3.1 Изменение масштаба яркости при просмотре

Для того, чтобы изменить масштаб значений для отображения, следует дважды щёлкнуть по миниатюре и перетащить ползунки масштаба для лучшего масштабирования. Это не изменяет само изображение, а лишь масштабирует его для просмотра. Закрываем окно, в котором меняли масштаб просмотра, что сохраняет настройки для текущего рабочего окружения.

#### Быстрое масштабирование

Для того, чтобы изменить масштаб, совсем не обязательно искать на панели инструментов какую-то кнопку: достаточно, удерживая клавишу CTRL на клавиатуре, покрутить колёсиком мыши - и изображение в окне будет менять свой масштаб. Естественно, на оригинальном файле это не отразится.

### Быстрое выделение области

Если нужно быстро выделить область интереса на изображении, достаточно зажать клавишу CTRL на клавиатуре и начать выделять мышью нужную область. Тут же будет создана новая область с названием, соответствующем текущему ряду и последнему свободному номеру ячейки (например, если ряд В и ячейка 14 последняя - новая будет называться В15).

### Быстрое перемещение по изображению

Колёса продолжают рулить! Колесом мыши можно скроллить изображение вверх-вниз, а если зажать SHIFT и использовать колесо мыши - скроллинг будет в направлениях "влево-вправо". Очень удобно перемещаться так без использования полос прокрутки.

### Горячие клавиши в pir2

Если вы часто используете какую-то функцию, есть смысл поставить на неё горячую клавишу. Для этого открываем меню, доходим до нужной нам функции кнопками клавиатуры, подсвечиваем её (или нажимаем её кнопкой мыши и держим для подсветки) и нажимаем к примеру сочетание клавиш CTRL+M - и теперь эту функцию можно вызвать по нажатию CTRL+M.

## 13.3.2 Анализ изображений в pir2

С помощью pir2 можно проводить довольно сложный анализ изображений: Фурье-анализ, корреляционный анализ, свёртка, low-pass/high-pass фильтры и прочее.

### Фурье-анализ в pir2

Часто бывает необходимо видеть Фурье-спектр изображения, особенно тогда, когда к нему применяются методы обработки. Для этого идём в Toolkits - Math - Fourier - Forward для

прямого фурье-преобразования. Считается оно в первый раз довольно долго, зато потом будет пересчитываться быстро.

Там же, в `pir2`, можно выполнить и обратное фурье-преобразование. Для этого идём в `Toolkits - Math - Fourier - Reverse` и получаем назад своё изображение.

### Гистограмма изображения в `pir2`

Гистограмма это зависимость количества пикселей одного уровня яркости от яркости изображения - она даёт представление о том, пикселей какой яркости на изображении больше или меньше. Функция чрезвычайно полезная при анализе изображений, и, разумеется, она присутствует в `pir2`. Для этого выделяем изображение, которое собираемся анализировать, и идём в меню `Toolkits - Histogram - Find - One Dimension`. В результате, как на скриншоте выше, имеем красивую и информативную гистограмму изображения.

### 13.3.3 Изменение изображений в `pir2`

Некоторые привычные операции, такие как кадрирование, в `pir2` есть, просто они могут выглядеть, согласно философии `pir2`, несколько непривычно...

#### Кадрирование

Есть и эта операция, причём её можно делать и визуально, и имея точные координаты. Точное кадрирование можно осуществить, либо когда вам известны координаты области, либо используя `pir2` в поточном режиме (для этого следует использовать команду `viprs` и мануал к ней). Отмечаем ячейку с изображением, которое необходимо кадрировать, и идём в `Toolkits - Image - Crop`. После этого появится ещё одна ячейка ниже, и потребуется указать координаты среза.

Визуальное кадрирование можно применить к изображению, открыв изображение на просмотр в окне, идём в меню `File - New - New Region`. Теперь меняем размер области и её

положение по вкусу. А можно сделать и ещё проще: для кадрирования в `pip2` достаточно, зажав клавишу `CTRL`, выделить желаемую область на изображении - и в следующей ячейке сразу же появится желаемая область выделения.

### Порог

Казалось бы, простая вещь - есть в любом уважающем себя графическом редакторе. В `pip2` это тоже есть, но не так очевидно. Мне пришлось некоторое время поломать голову и проявить немного сообразительности: порог, как выясняется, можно сделать в два этапа. В `pip2` есть простые статистические операции: среднее, минимум, максимум и прочее. Выделяем изображение и находим, например, среднее (`Toolkits - Math - Statistics - Mean`). В следующей ячейке появится число: Теперь выделяем, зажав `Shift`, сначала ячейку с числом, а потом ячейку с картинкой. После чего идём в математику (`Toolkits - Math`) и ищем `Relational` (Соотношения). Выбираем `Less than` - это ли не порог? Отлично, в следующей ячейке появится чёрно-белое изображение с порогом. На скриншоте выделено полупрозрачным. Обновлено: оказывается, всё проще - в `pip2` недавно появилась специальная функция порога, которая упрятана в меню `Toolkits - Image - Select - Threshold`.

### Склеивание изображений в `pip2`.

Чтобы склеить несколько изображений в одно, вовсе не нужен фотопоп - с этим прекрасно и быстро справляется `pip2`. Причём справляется тем лучше, чем больше изображений или фотографий нужно склеить. Например, если у вас имеются снимки со сканирующего микроскопа и нужно склеить десяток снимков - это лучше сделать в `pip2`. Для этого идём в меню `Toolkits - Image - Join - Left to Right` если хотим склеить изображения по горизонтали (левый край к правому краю) или `Top to Bottom` (если нужно склеить верхний край изображения к нижнему краю). Вот что при этом получается:

Пользуясь Toolbox - Image - Join, легко склеить несколько больших изображений в одно для последующего просмотра и анализа.

### **Корректировка перекоса яркости (tilt brightness)**

Следует отметить, что при научных съёмках часто на изображениях появляется перекося яркости: когда одна часть изображения освещена сильнее другой (меняющаяся яркость от края изображения к середине). Этот достаточно неприятный эффект можно устранить в pir2 так: Tools - Filters - Tilt brightness. Это позволит восстановить освещённость на изображении. А используя табличное свойство pir2, вы получаете возможность оперативно перерисовать большое изображение с учётом скорректированной яркости.

## Глава 14

# Интернет

### 14.1 настроить сетевую карту в Linux

Несмотря на то, что во многих современных дистрибутивах есть красивые графические утилиты по настройке сетевой карты, в ряде случаев проще настроить всё руками в конфигах. Это очень просто - вписать несколько строчек в нужные конфиги.

#### 14.1.1 Как посмотреть текущие настройки сетевой карты в Линукс

Просмотреть текущие настройки можно, набрав в консоли с привилегиями рута команду:

```
# ifconfig
```

или

```
$ sudo ifconfig
```



В результате в консоль будет выдано что-то вроде:

```
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:104 errors:0 dropped:0 overruns:0 frame:0
TX packets:104 errors:0 dropped:0 overruns:0 carrier:0
collisions :0 txqueuelen:0
RX bytes:9387 (9.1 KiB) TX bytes:9387 (9.1 KiB)

eth0 Link encap:Ethernet HWaddr 00:0A:E4:53:AA:2D
inet addr:192.168.1.5 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions :0 txqueuelen:1000
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
Interrupt:21 Base address:0x2c00
```

Отсюда видно, что настроены два сетевых интерфейса: петлевой (loopback) под названием lo, и обычная сетевая карта (которая Ethernet) называемая eth0. Причём если в интерфейс eth0 вставлен сетевой провод, идущий в свитч, будет написано RUNNING, то есть:

```
inet addr:192.168.1.5 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::20e:2eff:feec:ef9f/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500
Metric:1
```

Простая настройка сетевого интерфейса возможна командой:

```
# ifconfig eth0 192.168.1.5
```

Чтобы заглушить сетевой интерфейс, набираем:

```
# ifdown eth0
```

### 14.1.2 Настройка сетевого соединения в Линукс

Для этого потребуются права администратора и текстовый редактор (nano, vim, emacs). Так же нужно представлять, какой IP-адрес следует присвоить сетевой карте, адрес шлюза и прочие параметры.

Сначала нужно выяснить, как называется сетевой интерфейс, соответствующий вашей сетевой карте. Делается это так: в консоли пишем

```
$ dmesg | grep -i Eth
```

В результате должны получить что-то вроде (выделение моё):

```
8139too Fast Ethernet driver 0.9.28
eth0: RealTek RTL8139 at 0xdf822c00, 00:15:f2:51:ad:da, IRQ
      21
eth0: Identified 8139 chip type 'RTL-8101'
```

Так, похоже, что называется он eth0 (потому, что ethernet, 0 - нулевое устройство - довольно логично). В этом примере дальше будет предполагаться, что интерфейс - eth0.

### 14.1.3 Настройка сетевого соединения со статическим IP

Для этого следует отредактировать файл:

```
# nano /etc/network/interfaces
```

или

```
# sudo nano /etc/network/interfaces
```

Перед тем, как его править, следует решить, какой IP-адрес назначается машине. Если вы подключаетесь к уже настроенной сети - хорошо бы посмотреть, какие настройки есть там. Например, если подключение происходит к CorvettTelecom, настройки вам дадут. Если же сеть самопальная, например из двух компьютеров, нижеследующий пример подойдёт.

В общем, при назначении статических адресов нужно указать: IP-адрес (*address*), маску подсети (*netmask*) и адрес шлюза (*gateway*). Для локальной сети файл `/etc/network/interfaces` нужно изменить так:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.5
netmask 255.255.255.0
gateway 192.168.1.1
```

В этом примере назначен IP-адрес 192.168.1.5.

#### 14.1.4 Настройка сетевого соединения с динамическим IP

Редктируем тот же файл, отвечающий в Линукс за сетевые настройки:

```
# nano /etc/network/interfaces
```

или

```
$ sudo nano /etc/network/interfaces
```

Здесь всё проще: нужно лишь привести файл к виду:

```
auto lo
```

```
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

Так же можно настроить и из консоли:

```
# dhcpcd eth0
```

Для этого в системе должен быть установлен пакет `dhcpc3-client`

### 14.1.5 Два слова о беспроводных сетях

Чтобы изменить тип интерфейса

```
# wlanconfig ath0 destroy
# wlanconfig ath0 create wlandev wifi0
# wlanmode ath0
```

Для того, чтобы посмотреть список сетей, набираем:

```
# iwlist ath0 scan
```

или

```
# wlanconfig ath0 list ap
```

Если есть открытая сеть и нужно к ней подключиться, набираем от рута:

```
# iwconfig ath0 essid SomeESSID
```

Для того, чтобы соединение шифровалось с использованием WEP, пишем:

```
# iwconfig ath0 key ...
```

### 14.1.6 Настройка адресов серверов имён (DNS, nameserver)

Если вы не настроите это, то сможете подключаться к ресурсам сети только по IP-адресам, что несколько неудобно. Для того, чтобы в Линукс прописать адреса серверов имён (DNS address), нужно отредактировать файл `/etc/resolv.conf` или создать его, если его нет.

```
# nano /etc/resolv.conf
```

или

```
$ sudo nano /etc/resolv.conf
```

И прописать адреса следующим образом:

```
nameserver 192.168.1.1
nameserver 192.168.2.1
```

столько, сколько их нужно. Слово `nameserver` добавлять обязательно.

Чтобы изменения настройки сетевых интерфейсов вступили в силу... .. можно перезагрузиться, но это не наш метод. Без перезагрузки следует дать команду:

```
# /etc/init.d/networking restart
```

или

```
$ sudo /etc/init.d/networking restart
```

Всё, после этого настройки вступают в силу для всех сетевых интерфейсов.

## 14.2 Wget - повелитель закачек, или Как использовать wget

Ниже приводятся наиболее часто используемые мною опции.

Нужно просто скачать файл и/или продолжить загрузку, которая оборвалась. Это очень и очень легко сделать, написав в терминале несколько строчек.

Просто скачать файл wget-ом

```
$ wget ftp://vasya.pupkin.com/film.avi
```

Ещё вариант: часть файла скачана, загрузка оборвалась.

Для продолжения файла загрузки пишем:

```
$ wget -c ftp://vasya.pupkin.com/film.avi
```

или

```
$ wget --continue ftp://vasya.pupkin.com/film.avi
```

Вместо -c можно написать -continue, так как каждый ключ имеет короткую и длинную формы: длинные ключи проще запомнить, но дольше писать. Можно легко смешивать различные формы написания.

Чтобы выкачать файлы из списка, содержащего прямые ссылки:

```
wget -i pupkinlist
```

или

```
wget --input-file=pupkinlist
```

Здесь указывается только файл, в котором содержатся ссылки. Файл может так же быть HTML-страницей, в которой есть ссылки. Они будут выкачаны указанной выше командой.

### 14.2.1 Использование а-ля "Teleport Pro for Linux"

При скачивании веб-сайтов возможностей больше, и поэтому требуется больше ключей. Опять-таки, запоминать их все не обязательно, можно сделать скрипт (а лучше несколько - под разные случаи) и вызывать их. Так вот, если имеется веб-сайт, и хотелось бы иметь его локальную копию на компьютере, чтобы, отключившись от сети, можно было не торопясь его почитать.

Зеркалирование сайтов на локальную машину:

```
wget -m http://www.vasyapupkin.com/
```

Замечу, что при этом ссылки останутся абсолютными - то есть, будут указывать на Интернет-адреса, и удобно просматривать на локальной машине будет затруднительно.

Копирование сайта для локального просмотра:

```
wget -r -l0 -np -k http://www.vasyapupkin.com/
```

При этом будет включена рекурсивная выгрузка (ключ -r, -recursive), то есть не только файлы с главной страницы, но и все остальные, на которые ведут ссылки (ключ -l0 бесконечная вложенность ссылок). Имена ссылок будут переконвертированы в локальные для удобства просмотра (ключ -k). Так же при помощи ключа -np (no-parent) можно запретить wget подниматься выше начального адреса при рекурсивной загрузке, то есть если вы копируете `http://home.vasyapupkin.com/` то по ссылкам с основного сайта `http://www.vasyapupkin.com/` скопированы не будут.

### 14.2.2 И несколько других полезных ключей

Включение и исключение файлов при загрузке

```
-A acclist | -accept acclist
```

```
—R rejlis | —reject rejlis
```

Задаёт разделяемые запятыми шаблоны имён файлов, которые следует загружать (acclist) или игнорировать (rejlis).

```
—k | —convert—links
```

Преобразует абсолютные ссылки (типа `http://www...`) в относительные (типа `file:///home/vasya/www/index.html`) для удобства локального просмотра. Чтобы локально в браузере просмотреть скачанный сайт, открываете файл `index.html` в браузере и бродите по ссылкам точно так же, как если бы вы были подключены к Интернету.

```
—H —span—hosts
```

Разрешает `wget` скачивать данные с любого адреса, на который есть ссылка в запрашиваемом документе.

```
—p —page—requisites
```

Загружать все файлы, которые нужны для отображения страниц HTML. Например: рисунки, звук и каскадные стили. После завершения загрузки конвертировать ссылки в документе для просмотра в автономном режиме. Это касается не только видимых ссылок на другие документы, а ссылок на все внешние локальные файлы.

Ограничение канала загрузки `-limit-rate=значение в кб`. Например:

```
$ wget —limit—rate=50k http://www....
```

Это приведет к тому, что скорость загрузки не будет превышать указанных в примере пятидесяти килобайт/сек.

Siteget - простой скрипт для скачивания сайтов Если вам надо все эти премудрости с ключами `wget` и нужно просто



скачать сайт, то `siteget` вам определённо понравится. Это довольно простой скрипт с необходимыми параметрами для загрузки сайта. Взять скрипт можно [тут](#), использование очень простое:

```
$ siteget http://www.vasyapupkin.com/
```

Большое спасибо за ссылку [jetxee](#), который писал об этом [здесь](#).

## Глава 15

# Научное

### 15.1 Научная поисковая система на вашем Linux-десктопе

Некоторое время назад я писал о SWISH++, и вот теперь мне хотелось бы обобщить написанное в более краткой форме. Проблема поиска нужной статьи, если не помнишь точно где она находится в каталогах, действительно серьезная.

#### 15.1.1 Поиск дубликатов

Итак, когда я накачал нужных мне статей из Интернета, я прежде всего смотрю, есть ли они у меня в коллекции с помощью `fdupes`. Эта программа ищет дубликаты файлов:

```
$ fdupes -rd .
```

Точка в конце говорит о том, что искать дубликаты `fdupes` будет, начиная с текущего каталога, поэтому можно в каталоге со статьями сделать подкаталог `1/` и набросать туда скачанные статьи.

Fdupes сравнивает файлы как побайтово, так и с помощью подсчёта md5-суммы, работает невероятно быстро, сортировка производится в полуавтоматическом режиме: программа выводит в консоли дублирующиеся файлы и спрашивает, какой из них оставить. Об этом уже говорилось в статье о подходах к сортировке PDF-файлов.

### 15.1.2 Индексация текста в PDF-файлах

Сначала из документов pdf вытаскиваем текст, для чего просим утилиту find найти все файлы PDF и для каждого найденного выполнить pdftotext без выдачи предупреждений и без вставки символов разрыва страницы:

```
# find -name '*.pdf' -exec pdftotext -npgbrk -q {} \;
```

Надо отметить, что pdftotext хорошо работает только для английского языка, на котором, в общем, все статьи и написаны.

Теперь индексируем все текстовые файлы - в этом нам поможет собрат SWISH++ по имени index++ для индексации текстовых файлов:

```
# index++ -e "text:*.txt" .
```

Точка в конце означает, что поиск ведётся с текущего каталога. Подробнее об индексации и автоматизации этого процесса сказано здесь.

В результате в текущем каталоге появится файл swish++.index где, собственно, лежит информация об индексировании файлов. Теперь всё готово для поиска.

### 15.1.3 Ищем в архиве статей нужное

Для поиска используем search++ которая найдёт по нашему запросу в индексированной базе swish++.index файлы. Вот пример поиска статьи по математической морфологии, в которых нет упоминания про медицину:

```
$ search++ morphology and erosion and dilation not medicine
```

Мгновение спустя вижу результат (вывод сокращён):

```
# results: 125
99 ./Krylov2.txt 3771 Krylov2.txt
49 ./13300407.txt 3103 13300407.txt
46 ./morph1.slides.printing .6.txt 4369
    morph1.slides.printing .6.txt
37 ./lecture_morphology_sara.txt 6746
    lecture_morphology_sara.txt
30 ./SIGGRAPH2002_Sketch-Mitchell.txt 5308
    SIGGRAPH2002_Sketch-Mitchell.txt
26 ./MorphologicalImageProcessing.txt 7642
    MorphologicalImageProcessing.txt
25 ./phdsymp2002_ledda.txt 8298 phdsymp2002_ledda.txt
23 ./lab2_manual.txt 9313 lab2_manual.txt
23 ./Project 1.txt 9946 Project 1.txt
22 ./morphology.txt 11212 morphology.txt
22 ./edg/morphology.txt 11212 morphology.txt
22 ./slides-6-geometry.txt 11717 slides-6-geometry.txt
22 ./V1BFOGG8.txt 10797 V1BFOGG8.txt
18 ./71650638.txt 13978 71650638.txt
```

Первая колонка - релевантность, вторая - расположение файла относительно текущей директории, третья - размер файла, четвёртая - имя. Просто и понятно.

Теперь в консоли набираю `kdpf ./Krylov2.pdf` и смотрю нужную мне статью. И всё.

## 15.2 Совместная работа над документами $\text{\LaTeX}$

Когда вы сидите и тихо работаете над своей задачей - всё хорошо и здорово. Ровно до тех пор, пока не возникнет необходимость в совместной работе над статьёй, отчётом или другим

крупным документом в  $\text{\LaTeX}$ . Особенно этот процесс становится захватывающим при использовании разных кодировок и операционных систем.

### 15.2.1 Совместная работа над документами $\text{\LaTeX}$

Для того, чтобы взять под контроль процесс правок при совместной работе, перво-наперво нужно выбрать систему контроля ревизий.

### 15.2.2 Правки и комментарии в документах $\text{\LaTeX}$

Приведу свои результаты проб и ошибок в деле коллективной работы над латеховскими документами. Вот тут, кстати, начинаешь немного тосковать по Word и OpenOffice.Writer, в которых есть встроенные средства для коллективной работы (правки, исправления, пометки, заметки на полях). Некое подобие этого я реализовал в  $\text{\LaTeX}$ и с некоторым успехом применял для правок дипломов у своих студентов.

#### Дробление документов

С одним большим документом работать просто невозможно и жутко неудобно - так что главы дипломов, отчётов и крупных монографий лучше разделить на файлы и вставлять директивой `\input`. Это облегчит работу и системе контроля версий при обновлении рабочих копий - чтобы правки были неперекрывающиеся.

#### Зачёркивания и пометки цветом

Чтобы иметь возможность зачеркивать слова в  $\text{\LaTeX}$ , включаем в преамбуле документа пакет `ulem`. И далее в тексте для зачёркивания используем команду `\sout{чтобы зачёркивать текст}`.

Альтернатива пакету `ulem` - пакет `soul`, где для зачёркивания используется команда `\st{зачеркнуть это}`. Рядом обычно пишется вопрос или предложение фразы для соавтора - это лучше помечать цветом (при этом лучше сразу договориться, какой цвет выберет каждый соавтор, чтобы потом с ума не сойти). Для выделения цветом в  $\text{\LaTeX}$  можно использовать пакет `xcolor`, который подключаем в преамбуле. И далее помечаем в тексте слова `\textcolor{red}{красным шрифтом}`.

### Заметки на полях и комментарии в $\text{\LaTeX}$

Иногда хочется выразить своё мнение или особо обратить внимания соавтора на что-то - в этом случае стоит использовать заметки на полях. Для этого есть стандартная команда `\marginpar{Замечание соавтору сбоку на полях.}`

Но иногда комментарий длинный или хочется втянуть соавтора в письменную дискуссию - тогда нам придёт на помощь небольшой трюк. Можно определить команду комментирования, и тот текст, который будет в комментарии, не будет виден в откомпилированном документе, но в коде присутствовать будет. Для этого в преамбуле определяем новую команду так: `\newcommand{\comment}[1]{}` и в тексте можно дать волю своим полемическим способностям `\comment{, потому как этот текст всё равно не будет виден в откомпилированном  $\text{\LaTeX}$ -документе}`.

Эти в общем-то нехитрые приёмы, часть из которых со временем выветривается из моей просторной головы, помогают поддерживать адекватный уровень совместной работы.

## Глава 16

# Debian-специфичное

### 16.1 Поиск пакета по меткам в Debian GNU/Linux: debtags

В репозитории Debian GNU/Linux пакетов очень и очень много. Для поиска пакетов можно использовать много методов, однако всегда интересно посмотреть, а может быть есть полезная и нужная программа, которую пропустил. Для этого есть в каждом пакете теги (они же метки), а поиск по меткам лучше всего удаётся замечательной программе debtags.

Можно иметь под рукой огромный репозиторий и не знать, что в нем есть. Начиная с Debian Etch, появилась такая замечательная программа, как debtags. Она при умелом пользовании должна сильно помочь при поиске программ для конкретных задач.

#### 16.1.1 Поиск по аналогии

Например, поищем программы для просмотра изображений. Скажем, используем программу gqview - поищем её аналоги в

репозитории:

```
$ debtags related gqview
```

Результат:

```
pornview — Image and movie viewer/manager
gtksee — GTK-based clone of ACDSee (an image viewer)
```

Продолжаем искать, уже зная о трёх программах. Для этого последовательно увеличиваем такой критерий поиска, как "дистанция характеризующий похожесть искомых пакетов на указанный.

```
$ debtags related -d 1 gqview,pornview,gtksee
feh — imlib2 based image viewer
gthumb — an image viewer and browser
paul — Yet another image viewer
```

Таким образом нашли ещё три программы для просмотра изображений, feh, gthumb и paul. Используем их названия для дальнейшего поиска:

```
$ debtags related -d 2 gqview,pornview,
gtksee,feh,gthumb,paul
```

Получаем:

```
xsane — GTK+-based X11 frontend for SANE
showimg — A feature-rich image viewer
djview — Viewer for the DjVu image format
eog — Eye of Gnome graphics viewer program
xli — command line tool for viewing images in X11
```

Тут результат уже не точен, но 3 из 5 программ нам подходят. Среди горы пакетов найдено за пару минут целых 5 просмотрщиков.



### 16.1.2 Поиск по конкретным тэгам

Выбираем подходящие теги для поиска:

```
$ aptitude show gqview eog showing
```

И, наконец, ищем:

```
$ debtags search "use::viewing && works-with::image &&
interface::x11"
```

Получаем список из 51 пакета. Просмотрев его по диагонали, и выясняем, есть ещё несколько ранее не обнаруженных программ по просмотру изображений:

```
gimageview
xzgv
gwenview
glib
imgseek
kview
qiv
kuickshow
```

Осталось только выбрать самый подходящий.

### 16.1.3 "Умный" поиск

Теперь посмотрим, что из содержимого репозитория Дебиан умеет работать с изображениями:

```
$ debtags smartsearch images
```

В результате нас попросят уточнить область поиска. Ладно, хотим посмотреть на программы для растровых изображений.

```
Tag selection :
```

- |                                                                                                                                                                                                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"><li>1) works-with::image (420/15850)</li><li>2) works-with::image:raster (293/15850)</li><li>3) works-with-format::png (35/15850)</li><li>4) role :: program (4851/15850)</li><li>5) interface :: x11 (2436/15850)</li><li>6) hardware::camera (44/15850)</li><li>7) x11::application (2175/15850)</li></ol> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Так можно найти много интересного в репозитории!

## Глава 17

# ЛaTeX

### 17.1 Мелкие тонкости презентаций в Beamer

Есть ряд мелких, но противных тонкостей при работе с Beamer, на которые мне пришлось нарваться при работе над очередной презентацией.

#### 17.1.1 Как выровнять текст по ширине в Beamer?

Для этого используем возможности пакета `ragged2e` и в преамбуле документа презентации пишем:

```
\usepackage{ragged2e}
\justifying
```

#### 17.1.2 Куча слайдов в Beamer и как их структурировать

На самом деле, в Beamer никто не запрещает использование `\section` и `\subsection` - в тексте презентации они никак не

отражаются (за исключением, может быть, некоторых тем), а структурировать презентацию это сильно помогает.

В презентации много текста, надо бы впихнуть туда ещё немного - как? Для этого можно использовать такую конструкцию:

```
\begin{frame}[shrink=5]
...
\end{frame}
```

Это уменьшит содержимое слайда на 5% (можно поставить и больше, если надо).

### 17.1.3 Как включить/выключить нумерацию рисунков?

Чтобы включить нумерацию рисунков, в преамбуле пишем:

```
\setbeamertemplate{caption}[numbered]
```

Чтобы выключить навигационные символы на презентации в Beamer, используем строчку в преамбуле:

```
\setbeamertemplate{navigation symbols}{}
```

Для того, чтобы вообще убрать слово "РИС" в презентации и при этом не использовать пакет caption (он с Beamer конфликтует), в преамбуле пишем:

```
\addto\captions<your language>{
\renewcommand{\tablename}{bar}}
```

Или, если не используется babel, пишем:

```
\renewcommand{\figurename}{foo}
```

Если нужно, чтобы слайды разбивались на несколько автоматически, употребляем тег:

```
\begin{frame}[allowframebreaks,Литературные]{ источники}
.....
\end{frame}
```

Теперь содержимое слайда будет автоматически разбиваться Beamer на несколько слайдов.