

# Дерево директорий Linux

В рамках данной главы производится обзор наиболее часто используемых директорий из **дерева директорий файловой системы Linux**. Также данная глава может служить доказательством утверждения о том, что в Unix-системе все является файлом.

## Стандарт иерархии файловой системы

Многие дистрибутивы Linux частично следуют **Стандарту иерархии файловой системы (Filesystem Hierarchy Standard)**. Данный стандарт может оказаться полезным для будущего процесса стандартизации деревьев директорий файловых систем Unix/Linux. Стандарт **FHS** доступен в сети по адресу <http://www.pathname.com/fhs/>, причем на данном ресурсе мы можем прочитать: "Стандарт иерархии файловой системы был создан с целью его использования разработчиками дистрибутивов Unix, разработчиками пакетов для распространения программного обеспечения и разработчиками операционных систем. Однако, данный стандарт является в большей степени справочным материалом, нежели руководством по работе с файловой системой Unix или с иерархиями директорий."

## Страница руководства `man hier`

Существуют некоторые различия в иерархиях файловых систем различных **дистрибутивов Linux**. Для того, чтобы ознакомиться с информацией об иерархии файловой системы вашей машины, используйте команду **`man hier`**. На данной странице руководства будут приведены пояснения относительно структуры дерева директорий системы, установленной на вашем компьютере.

## Корневая директория /

Структуры директорий всех систем Linux начинаются с **корневой директории**. Корневая директория обозначается с помощью символа **прямого слэша**, а именно, `/`. Все файлы, которые существуют в

вашей системе Linux, находится ниже данной корневой директории в дереве директорий. Давайте рассмотрим содержимое этой корневой директории.

```
[paul@RHELv4u3 ~]$ ls /
bin  dev  home  media  mnt  proc  sbin  srv  tftpboot
usr
boot  etc  lib  misc  opt  root  selinux  sys  tmp
var
```

## Директории для хранения бинарных файлов

Бинарные файлы являются файлами, содержащими скомпилированный исходный код (или машинный код). Бинарные файлы могут **исполняться** на компьютере. Иногда бинарные файлы также называются **исполняемыми** файлами.

### Директория **/bin**

Директория **/bin** содержит **бинарные** файлы, которые могут использоваться всеми пользователями. В соответствии со спецификацией FHS, директория **/bin** должна содержать исполняемые файлы **/bin/cat** и **/bin/date** (помимо других исполняемых файлов).

В примере ниже вы можете увидеть список исполняемых файлов, являющихся реализациями таких команд, как **cat**, **cp**, **cpio**, **date**, **dd**, **echo**, **grep** и т.д. Многие из упомянутых команд будут рассмотрены в рамках данной книги.

```
paul@laika:~$ ls /bin
archdetect      egrep           mt
setupcon
autopartition   false           mt-gnu          sh
bash            fgconsole       mv
sh.distrib
bunzip2         fgrep           nano            sleep
bzcat           fuser           nc
stralign
bzipcmp         fusermount      nc.traditional  stty
bzdiff          get_mounthptions netcat          su
bzegrep         grep            netstat         sync
bzexe           gunzip          ntfs-3g         sysfs
bzfgrep         gzexe           ntfs-3g.probe   tailf
```

bzgrep	gzip	parted_devices	tar
bzip2	hostname	parted_server	
tempfile			
bzip2recover	hw-detect	partman	touch
bzless	ip	partman-commit	true
bzmore	kbd_mode	perform_recipe	
ulockmgr			
cat	kill	pidof	
umount			
...			

## Другие директории /bin

Вы можете обнаружить **поддиректорию /bin** во многих других директориях. Например, пользователь с именем **serena** может разместить свои собственные приложения в поддиректории **/home/serena/bin**.

Файлы некоторых приложений, обычно в случае установки путем непосредственной сборки из исходного кода, устанавливаются в директорию **/opt**. К примеру, при установке **сервера samba** для хранения бинарных файлов может быть использована поддиректория **/opt/samba/bin**.

## Директория /sbin

Директория **/sbin** содержит бинарные файлы, предназначенные для настройки операционной системы. Многие из **бинарных файлов** для настройки системы требуют наличия привилегий пользователя **root** для выполнения определенных задач.

В примере ниже приведен список **бинарных файлов** для настройки системы, предназначенных для изменения IP-адреса, работы с разделами жестких дисков и создания файловой системы ext4.

```
paul@ubu1010:~$ ls -l /sbin/ifconfig /sbin/fdisk
/sbin/mkfs.ext4
-rwxr-xr-x 1 root root 97172 2011-02-02 09:56 /sbin/fdisk
-rwxr-xr-x 1 root root 65708 2010-07-02 09:27
/sbin/ifconfig
-rwxr-xr-x 5 root root 55140 2010-08-18 18:01
/sbin/mkfs.ext4
```

## Директория **/lib**

Бинарные файлы из директорий **/bin** и **/sbin** обычно используют разделяемые библиотеки, расположенные в директории **/lib**. В примере ниже приведен список некоторых файлов из директории **/lib**.

```
paul@laika:~$ ls /lib/libc*
/lib/libc-2.5.so      /lib/libcfont.so.0.0.0
/lib/libcom_err.so.2.1
/lib/libcap.so.1      /lib/libcidn-2.5.so
/lib/libconsole.so.0
/lib/libcap.so.1.10   /lib/libcidn.so.1
/lib/libconsole.so.0.0.0
/lib/libcfont.so.0    /lib/libcom_err.so.2
/lib/libcrypt-2.5.so
```

## Поддиректория **/lib/modules**

Обычно **ядро Linux** загружает модули из директории **/lib/modules/\$версия-ядра/**. Содержимое этой директории будет подробно описано в главе, посвященной ядру Linux.

## Директории **/lib32** и **/lib64**

На данный момент осуществляется медленная миграция с **32-битных** на **64-битные** системы. По этой причине вы можете обнаружить в своей системе директории с именами **/lib32** и **/lib64**, которые указывают на размеры регистров, использованные в процессе компиляции расположенных в этих директориях разделяемых библиотек. 64-битный компьютер может содержать некоторые 32-битные бинарные файлы и библиотеки, используемые для достижения совместимости с устаревшими приложениями. В примере ниже утилита **file** используется для демонстрации описанных различий между разделяемыми библиотеками.

```
paul@laika:~$ file /lib32/libc-2.5.so
/lib32/libc-2.5.so: ELF 32-bit LSB shared object, Intel
80386, version 1 (SYSV), for GNU/Linux 2.6.0, stripped
paul@laika:~$ file /lib64/libcap.so.1.10
/lib64/libcap.so.1.10: ELF 64-bit LSB shared object, AMD
x86-64, version 1 (SYSV), stripped
```

Формат ELF (**формат исполняемых и компокуемых файлов - Executable and Linkable Format**) используется практически во

всех Unix-подобных операционных системах с момента выпуска **System V**.

## Директория **/opt**

Директория **/opt** предназначена для хранения **вспомогательного** программного обеспечения. В большинстве случаев данное программное обеспечение устанавливается не из репозитория дистрибутива. В многих системах директория **/opt** пуста.

При установке пакета программного обеспечения большого объема файлы из него могут копироваться в поддиректории **/bin**, **/lib**, **/etc** директории **/opt/\$имя-пакета/**. Например, в том случае, если пакет программного обеспечения носит имя **wp**, файлы из него будут устанавливаться в директорию **/opt/wp**, при этом бинарные файлы будут устанавливаться в поддиректорию **/opt/wp/bin**, а файлы страниц руководств - в поддиректорию **/opt/wp/man**.

## Директории для хранения файлов конфигурации

### Директория **/boot**

Директория **/boot** содержит все файлы, необходимые для загрузки компьютера. Эти файлы не изменяются очень часто. В системах Linux в данной директории обычно можно обнаружить поддиректорию **/boot/grub**. Директория **/boot/grub** содержит файл **/boot/grub/grub.cfg** (на более старых системах также может использоваться файл **/boot/grub/grub.conf**), в рамках которого описывается меню загрузки, отображаемое перед загрузкой ядра ОС.

### Директория **/etc**

Все специфичные для машины **конфигурационные файлы** должны быть расположены в директории **/etc**. Изначально имя директории **/etc** была образовано от слова **etcetera** (и так далее), но сегодня люди часто расшифровывают его как **Editable Text Configuration** (директория с редактируемыми текстовыми файлами конфигурации).

Во многих случаях имена конфигурационных файлов совпадают с именами приложений или протоколов, а в качестве расширений этих файлов используется строка **.conf**.

```
paul@laika:~$ ls /etc/*.conf
/etc/adduser.conf          /etc/ld.so.conf
/etc/scrollkeeper.conf
```

```

/etc/brltty.conf      /etc/lftp.conf
/etc/sysctl.conf
/etc/ccertificates.conf /etc/libao.conf
/etc/syslog.conf
/etc/cvs-cron.conf    /etc/logrotate.conf
/etc/ucf.conf
/etc/ddclient.conf    /etc/ltrace.conf
/etc/uniconf.conf
/etc/debconf.conf     /etc/mke2fs.conf
/etc/updatedb.conf
/etc/deluser.conf     /etc/netscsid.conf
/etc/usplash.conf
/etc/fdmount.conf     /etc/nsswitch.conf
/etc/uswsusp.conf
/etc/hdparm.conf      /etc/pam.conf
/etc/vnc.conf
/etc/host.conf        /etc/pnm2ppa.conf
/etc/wodim.conf
/etc/inetd.conf       /etc/povray.conf
/etc/wvdial.conf
/etc/kernel-img.conf  /etc/resolv.conf
paul@laika:~$

```

В директории **/etc** также можно обнаружить большое количество других важных файлов.

### Поддиректория **/etc/init.d/**

Во многих дистрибутивах Unix/Linux имеется директория **/etc/init.d**, которая содержит сценарии для запуска и остановки **демонов**. Эта поддиректория может исчезнуть в процессе перехода дистрибутивов Linux на системы инициализации, которые заменят старую систему инициализации **init**, используемую для запуска всех **демонов**.

### Поддиректория **/etc/X11/**

Управление системой вывода графики осуществляется средствами программного обеспечения от организации X.org Foundation (а именно, сервера оконной системы **X Window System** или просто **X**). Файл конфигурации для вашего сервера оконной системы носит имя **/etc/X11/xorg.conf**.

## Поддиректория /etc/skel/

Содержимое директории **каркаса /etc/skel** копируется в домашнюю директорию при создании учетной записи пользователя. Она обычно содержит такие скрытые файлы, как сценарий **.bashrc**.

## Поддиректория /etc/sysconfig/

Данная директория, не упомянутая в спецификации FHS, содержит большое количество файлов конфигурации компонентов дистрибутива **Red Hat Enterprise Linux**. Впоследствии мы будем более подробно рассматривать некоторые из этих файлов. В примере ниже приведен список файлов директории **/etc/sysconfig** дистрибутива RHELv4u4 в случае установки всех связанных с данной директорией программных компонентов.

```
paul@RHELv4u4:~$ ls /etc/sysconfig/
apmd                firstboot           irda                 network
saslauthd
apm-scripts         grub                irqbalance          networking
selinux
authconfig          hidd                keyboard             ntpd
spamassassin
autofs              httpd               kudzu                openib.conf
squid
bluetooth           hwconf              lm_sensors           pand
syslog
clock               il8n                mouse                pcmcia
sys-config-sec
console             init                mouse.B              pgsql
sys-config-users
crond               installinfo         named                prelink
sys-logviewer
desktop             ipmi                netdump              rawdevices
tux
diskdump            iptables            netdump_id_dsa       rhn
vncservers
dund                iptables-cfg        netdump_id_dsa.p     samba
xinetd
paul@RHELv4u4:~$
```

Файл **/etc/sysconfig/firstboot** сообщает агенту настройки дистрибутива Red Hat о том, что он не должен запускаться после

загрузки системы. В том случае, если вы желаете использовать агент настройки дистрибутива Red Hat после следующей перезагрузки, вам следует просто удалить данный файл и выполнить команду **chkconfig --level 5 firstboot on**. Агент настройки дистрибутива Red Hat позволяет устанавливать последние обновления системы, создавать учетные записи пользователей, пользоваться функциями портала Red Hat Network, а также выполнять другие действия. После запуска он снова создаст файл `/etc/sysconfig/firstboot`.

```
paul@RHELv4u4:~$ cat /etc/sysconfig/firstboot
RUN_FIRSTBOOT=NO
```

Файл `/etc/sysconfig/harddisks` содержит дополнительные параметры настройки жестких дисков. Формат файла описан в самом файле.

Вы можете ознакомиться с описанием программного обеспечения, обнаруженного утилитой **kudzu**, которое сохраняется в файле `/etc/sysconfig/hwconf`. Kudzu является приложением от компании Red Hat, предназначенным для автоматического обнаружения и настройки аппаратного обеспечения.

Тип клавиатуры и таблица соответствия символов устанавливаются в файле `/etc/sysconfig/keyboard`. Для получения дополнительной информации о настройках клавиатуры в консоли следует обратиться к следующим страницам руководств **keymaps(5)**, **dumpkeys(1)**, **loadkeys(1)**, а также к содержимому директории `/lib/kbd/keymaps/`.

```
root@RHELv4u4:/etc/sysconfig# cat keyboard
KEYBOARDTYPE="pc"
KEYTABLE="us"
```

Файлы для настройки сетевых устройств из данной директории будут обсуждаться в главе, посвященной настройке сети.

## Директории для хранения данных

### Директория `/home`

Пользователи могут хранить персональные данные и данные проектов в директории `/home`. Обычно (но не всегда в соответствии со спецификацией FHS) имя домашней директории пользователя устанавливается в соответствии с полным именем пользователя в формате `/home/$имя_пользователя`. Например:

```
paul@ubu606:~$ ls /home
geert annik sandra paul tom
```



Помимо выделения каждому пользователю (или каждому проекту или группе) места для хранения персональных данных в рамках домашней директории, в рамках этой же директории выделяется место для хранения данных профиля пользователя. Типичный профиль пользователя Unix содержит множество скрытых файлов (файлов, имена которых начинаются с точки). Скрытые файлы пользовательского профиля Unix содержат параметры, установленные для данного пользователя.

```
paul@ubu606:~$ ls -d /home/paul/.*
/home/paul/.                /home/paul/.bash_profile
/home/paul/.ssh
/home/paul/..              /home/paul/.bashrc
/home/paul/.viminfo
/home/paul/.bash_history   /home/paul/.lessht
```

### Директория **/root**

Во многих системах директория **/root** является стандартной директорией для хранения персональных данных и данных профиля **пользователя root**. В том случае, если ее не существует по умолчанию, администраторы могут создать ее самостоятельно.

### Директория **/srv**

Вы можете использовать директорию **/srv** для хранения данных, которые **обрабатываются вашей системой**. Спецификация FHS позволяет хранить в этой директории данные **cvs**, **rsync**, **ftp** и **www**. Кроме того, спецификация FHS подтверждает возможность использования таких административных имен для поддиректорий, как **/srv/project55/ftp** и **/srv/sales/www**.

В системах Sun Solaris (или Oracle Solaris) для этой цели используется директория **/export**.

### Директория **/media**

Директория **/media** служит точкой монтирования для таких устройств для работы со **съёмными носителями**, как приводы CD-ROM, цифровые камеры, а также различные устройства, подключаемые по шине USB. Так как директория **/media** является достаточно новой в мире систем Unix, вы с высокой вероятностью можете встретить системы, не использующие данную директорию. К примеру, несмотря на то, что система Solaris 9 не имеет рассматриваемой директории, эта директория присутствует в системе Solaris 10. Боль-

шинство дистрибутивов Linux на сегодняшний день монтирует все съемные носители в директорию **/media**.

```
paul@debian5:~$ ls /media/  
cdrom  cdrom0  usbdisk
```

### Директория **/mnt**

Директория **/mnt** должна быть пустой и использоваться исключительно для создания временных точек монтирования файловых систем (в соответствии со спецификацией FHS).

Администраторы систем Unix и Linux обычно создают в данной директории множество поддиректорий, таких, как **/mnt/something/**. Вы, скорее всего, столкнетесь с системами с более чем одной директорией, созданной и/или смонтированной в рамках директории **/mnt** для работы с различными локальными и удаленными файловыми системами.

### Директория **/tmp**

Приложения и пользователи должны использовать директорию **/tmp** для хранения временных данных при необходимости. Данные, хранимые в директории **/tmp**, могут в реальности храниться как на диске, так и в оперативной памяти. В обоих случаях обслуживание хранилища временных данных осуществляется средствами операционной системы. Никогда не используйте директорию **/tmp** для хранения данных, которые являются важными или которые вы желаете архивировать.

## Директории в оперативной памяти

### Директория **/dev**

Файлы устройств из директории **/dev** выглядят как обычные файлы, но на самом деле не являются обычными файлами, размещенными на жестком диске. Директория **/dev** заполняется файлами в процессе определения устройств средствами ядра операционной системы.

### Стандартные физические устройства

Стандартные устройства, такие, как жесткие диски, представлены файлами устройств в директории **/dev**. В примере ниже приведен список файлов устройств SATA ноутбука, а также устройств IDE настольного компьютера. (Подробное описание назначения этих файлов устройств будет приведено ниже.)

#

```
# Устройства SATA или SCSI или USB
```

```
#
```

```
paul@laika:~$ ls /dev/sd*
```

```
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sdb  
/dev/sdb1 /dev/sdb2
```

```
#
```

```
# Устройства IDE или ATAPI
```

```
#
```

```
paul@barry:~$ ls /dev/hd*
```

```
/dev/hda /dev/hda1 /dev/hda2 /dev/hdb /dev/hdb1  
/dev/hdb2 /dev/hdc
```

Помимо представления физических устройств, некоторые файлы устройств выполняют специальные функции. Эти специальные файлы устройств могут оказаться очень полезными.

### Файлы устройств **/dev/tty** и **/dev/pts**

К примеру, файл устройства **/dev/tty1** представляет терминал или консоль, соединенную с системой. (Не стоит ломать голову над точными значениями терминов 'терминал' или 'консоль', так как в данном случае имеется в виду интерфейс командной строки системы.) При вводе команд в эмуляторе терминала, поставляемом в составе такого графического окружения рабочего стола, как Gnome или KDE, ваш терминал будет представлен файлом устройства **/dev/pts/1** (вместо числа 1 может использоваться другое число).

### Файл устройства **/dev/null**

В Linux вы можете обнаружить и другие файлы специальных устройств, такие, как файл устройства **/dev/null**, которое может рассматриваться как черная дыра; хотя соответствующее устройство и имеет неограниченную емкость, после записи из него не могут быть прочитаны никакие данные. Говоря техническим языком, любые записанные на представленное файлом **/dev/null** устройство данные будут просто отброшены. Представленное файлом **/dev/null** устройство может быть использовано для отбрасывания ненужного вывода различных команд. *Помните о том, что представленное файлом **/dev/null** устройство не является удачным местом для хранения ваших резервных копий данных ;-).*

## Директория /proc и взаимодействие с ядром ОС

Директория **/proc** является другой специальной директорией, которая содержит файлы, кажущиеся на первый взгляд обычными файлами, но не занимающие места на диске. На самом деле содержимое данной директории является представлением ядра ОС, а точнее, используемых ядром ОС структур данных и предназначено для непосредственного взаимодействия с ядром ОС. В директорию **/proc** монтируется специальная файловая система procfs.

```
paul@RHELv4u4:~$ mount -t proc  
none on /proc type proc (rw)
```

При выводе содержимого директории **/proc** можно обнаружить множество директорий с именами, представленными числовыми значениями (в любой системе Unix), а также некоторые интересные файлы (в Linux).

```
mul@laika:~$ ls /proc
```

1	2339	4724	5418	6587	7201	cmdline
mounts						
10175	2523	4729	5421	6596	7204	cpuinfo
mtrr						
10211	2783	4741	5658	6599	7206	crypto
net						
10239	2975	4873	5661	6638	7214	devices
pagetypeinfo						
141	29775	4874	5665	6652	7216	diskstats
partitions						
15045	29792	4878	5927	6719	7218	dma
sched_debug						
1519	2997	4879	6	6736	7223	driver
scsi						
1548	3	4881	6032	6737	7224	execdomains
self						
1551	30228	4882	6033	6755	7227	fb
slabinfo						
1554	3069	5	6145	6762	7260	filesystems
stat						
1557	31422	5073	6298	6774	7267	fs
swaps						

1606	3149	5147	6414	6816	7275	ide
sys						
180	31507	5203	6418	6991	7282	interrupts
sysrq-trigger						
181	3189	5206	6419	6993	7298	iomem
sysvipc						
182	3193	5228	6420	6996	7319	ioports
timer_list						
18898	3246	5272	6421	7157	7330	irq
timer_stats						
19799	3248	5291	6422	7163	7345	kallsyms
tty						
19803	3253	5294	6423	7164	7513	kcore
uptime						
19804	3372	5356	6424	7171	7525	key-users
version						
1987	4	5370	6425	7175	7529	kmsg
version_signature						
1989	42	5379	6426	7188	9964	loadavg
vmcore						
2	45	5380	6430	7189	acpi	locks
vmnet						
20845	4542	5412	6450	7191	asound	meminfo
vmstat						
221	46	5414	6551	7192	buddyinfo	misc
zoneinfo						
2338	4704	5416	6568	7199	bus	modules

Давайте обратим внимание на свойства файлов из директории **/proc**. При рассмотрении даты и времени изменения данных файлов можно отметить, что эти параметры соответствуют текущим значениями даты и времени, из чего можно сделать вывод, что содержимое данных файлов постоянно обновляется (для предоставления доступа к актуальному содержимому используемых ядром ОС структур данных).

```
paul@RHELv4u4:~$ date
```

```
Пн янв 29 18:06:32 EST 2007
```

```
paul@RHELv4u4:~$ ls -al /proc/cpuinfo
```

```
-r--r--r-- 1 root root 0 ноя 29 18:06 /proc/cpuinfo
```

```
paul@RHELv4u4:~$
```

```
paul@RHELv4u4:~$ ...через некоторое время...
```

```
paul@RHELv4u4:~$
```

```
paul@RHELv4u4:~$ date
```

```
Пн янв 29 18:10:00 EST 2007
```

```
paul@RHELv4u4:~$ ls -al /proc/cpuinfo
```

```
-r--r--r-- 1 root root 0 ноя 29 18:10 /proc/cpuinfo
```

Размер большинства файлов из директории `/proc` равен 0 байт, но при этом файлы из данной директории содержат данные, а иногда большие объемы данных. Вы можете ознакомиться с этими данными, используя команду `cat` по отношению к таким файлам, как файл **`/proc/cpuinfo`**, который содержит информацию о центральном процессоре.

```
paul@RHELv4u4:~$ file /proc/cpuinfo
```

```
/proc/cpuinfo: empty
```

```
paul@RHELv4u4:~$ cat /proc/cpuinfo
```

```
processor      : 0
vendor_id     : AuthenticAMD
cpu family    : 15
model         : 43
model name    : AMD Athlon(tm) 64 X2 Dual Core
Processor 4600+
stepping      : 1
cpu MHz       : 2398.628
cache size    : 512 KB
fdiv_bug      : no
hlt_bug       : no
f00f_bug      : no
coma_bug      : no
fpu           : yes
fpu_exception : yes
cpuid level   : 1
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic
mtrr pge...
```

bogomips : 4803.54

*Для сравнения ниже приведено содержимое файла /proc/cpuinfo системы Sun Sunblade 1000...*

```
paul@pasha:~$ cat /proc/cpuinfo
cpu : TI UltraSparc III (Cheetah)
fpu : UltraSparc III integrated FPU
promlib : Version 3 Revision 2
prom : 4.2.2
type : sun4u
ncpus probed : 2
ncpus active : 2
Cpu0Bogo : 498.68
Cpu0ClkTck : 000000002cb41780
Cpu1Bogo : 498.68
Cpu1ClkTck : 000000002cb41780
MMU Type : Cheetah
State:
CPU0: online
CPU1: online
```

Большая часть файлов из директории /proc предназначена исключительно для чтения, причем для чтения некоторых из них требуются привилегии пользователя root; в некоторые файлы могут записываться данные, причем в директории **/proc/sys** таких файлов большинство. Давайте поговорим о некоторых файлах из директории /proc.

### Файл **/proc/interrupts**

В системе архитектуры x86 в файле **/proc/interrupts** содержится информация о запросах прерываний.

```
paul@RHELv4u4:~$ cat /proc/interrupts
          CPU0
 0:   13876877    IO-APIC-edge  timer
 1:         15    IO-APIC-edge  i8042
 8:          1    IO-APIC-edge  rtc
 9:          0    IO-APIC-level  acpi
12:         67    IO-APIC-edge  i8042
```

```

14:      128      IO-APIC-edge  ide0
15:    124320      IO-APIC-edge  ide1
169:   111993      IO-APIC-level  ioc0
177:    2428      IO-APIC-level  eth0
NMI:         0
LOC:   13878037
ERR:         0
MIS:         0

```

При использовании машины с двумя центральными процессорами данный файл выглядит следующим образом.

```
paul@laika:~$ cat /proc/interrupts
```

```

          CPU0      CPU1
 0:      860013      0  IO-APIC-edge  timer
 1:       4533      0  IO-APIC-edge  i8042
 7:         0      0  IO-APIC-edge  parport0
 8:   6588227      0  IO-APIC-edge  rtc
10:       2314      0  IO-APIC-fastehoi  acpi
12:       133      0  IO-APIC-edge  i8042
14:         0      0  IO-APIC-edge  libata
15:       72269      0  IO-APIC-edge  libata
18:         1      0  IO-APIC-fastehoi  yenta
19:   115036      0  IO-APIC-fastehoi  eth0
20:   126871      0  IO-APIC-fastehoi  libata,
ohci1394
21:    30204      0  IO-APIC-fastehoi  ehci_hcd:usb1,
uhci_hcd:usb2
22:    1334      0  IO-APIC-fastehoi  saa7133[0],
saa7133[0]
24:   234739      0  IO-APIC-fastehoi  nvidia
NMI:         72      42
LOC:   860000    859994
ERR:         0

```

#### Файл /proc/kcore

Физическая память представлена файлом **/proc/kcore**. Не пытайтесь использовать команду `cat` по отношению к этому файлу;



вместо этого при необходимости исследования содержимого оперативной памяти используйте отладчик. Размер файла `/proc/kcore` совпадает с объемом вашей оперативной памяти плюс четыре байта.

```
paul@laika:~$ ls -lh /proc/kcore
-r----- 1 root root 2.0G 2007-01-30 08:57 /proc/kcore
paul@laika:~$
```

## Директория `/sys` для работы с системой горячего подключения устройств ядра Linux 2.6

Директория `/sys` была создана в процессе разработки версии 2.6 ядра Linux. С момента выпуска версии 2.6 ядро Linux использует файловую систему **sysfs** для реализации механизма горячего подключения устройств, использующих шины **usb** и **IEEE 1394 (FireWire)**. Обратитесь к страницам руководств `udev(8)` (данная подсистема пришла на смену подсистеме **devfs**) и `hotplug(8)` для получения дополнительной информации (или посетите ресурс <http://linux-hotplug.sourceforge.net/>).

По существу, директория `/sys` содержит файлы с информацией об используемом аппаратном обеспечении.

## Директория системных ресурсов Unix `/usr`

Несмотря на то, что имя директории `/usr` напоминает слово `user` (пользователь), не следует забывать о том, что на самом деле оно расшифровывается как **Unix System Resources** (директория системных ресурсов Unix). Иерархия поддиректорий директории `/usr` должна содержать разделяемые данные приложений, доступные только для чтения. Некоторые системные администраторы осуществляют монтирование файловой системы `/usr` в режиме только для чтения. В этом случае данная директория должна быть расположена на отдельном разделе жесткого диска или на разделяемом ресурсе NFS.

## Директория `/usr/bin`

Директория `/usr/bin` содержит множество реализаций команд.

```
paul@deb508:~$ ls /usr/bin | wc -l
1395
```

(В системе Solaris директория `/bin` является символьной ссылкой на директорию `/usr/bin`.)

## Директория **/usr/include**

Директория **/usr/include** содержит общедоступные заголовочные файлы для языка программирования C.

```
paul@ubu1010:~$ ls /usr/include/
aalib.h          expat_config.h    math.h
search.h
af_vfs.h         expat_external.h  mcheck.h
semaphore.h
aio.h           expat.h           memory.h
setjmp.h
AL              fcntl.h           menu.h
sgtty.h
aliases.h       features.h        mntent.h
shadow.h
...
```

## Директория **/usr/lib**

Директория **/usr/lib** содержит разделяемые библиотеки, которые не используются непосредственно пользователями или сценариями.

```
paul@deb508:~$ ls /usr/lib | head -7
4Suite
ao
apt
arj
aspell
avahi
bonobo
```

## Директория **/usr/local**

Директория **/usr/local** может использоваться системным администратором для локальной установки программного обеспечения.

```
paul@deb508:~$ ls /usr/local/
bin  etc  games  include  lib  man  sbin  share  src
paul@deb508:~$ du -sh /usr/local/
128K   /usr/local/
```

## Директория `/usr/share`

Директория `/usr/share` содержит независимые от архитектуры данные. Как вы можете заметить, данная директория имеет значительный размер.

```
paul@deb508:~$ ls /usr/share/ | wc -l
```

```
263
```

```
paul@deb508:~$ du -sh /usr/share/
```

```
1.3G    /usr/share/
```

Обычно данная директория содержит поддиректорию `/usr/share/man`, предназначенную для хранения файлов страниц руководств.

```
paul@deb508:~$ ls /usr/share/man
```

```
cs  fr          hu          it.UTF-8  man2  man6
pl.IS08859-2  sv
de  fr.IS08859-1  id          ja          man3  man7
pl.UTF-8      tr
es  fr.UTF-8      it          ko          man4  man8
pt_BR        zh_CN
fi  gl          it.IS08859-1  man1        man5  pl    ru
zh_TW
```

Также данная директория содержит поддиректорию `/usr/share/games`, предназначенную для хранения всех статических данных игр (таким образом, в данной директории не могут находиться файлы со списками рекордов или журналами игрового процесса).

```
paul@ubu1010:~$ ls /usr/share/games/
```

```
openttd  wesnoth
```

## Директория `/usr/src`

Директория `/usr/src` является рекомендуемой директорией для хранения файлов исходного кода ядра ОС.

```
paul@deb508:~$ ls -l /usr/src/
```

```
итого 12
```

```
drwxr-xr-x  4 root root 4096 2011-02-01 14:43 linux-headers-2.6.26-2-686
```

```
drwxr-xr-x 18 root root 4096 2011-02-01 14:43 linux-headers-2.6.26-2-common
```

```
drwxr-xr-x  3 root root 4096 2009-10-28 16:01 linux-  
kbuild-2.6.26
```

### Директория для изменяемых данных /var

Файлы заранее неизвестного размера, такие, как файлы журналов, файлы кэша и файлы очереди печати должны сохраняться в директории **/var**.

### Директория /var/log

Директория **/var/log** выполняет функции центрального хранилища всех файлов журналов.

```
[paul@RHEL4b ~]$ ls /var/log
```

acpid	cron.2	maillog.2	quagga
secure.4			
amanda	cron.3	maillog.3	radius
spooler			
anaconda.log	cron.4	maillog.4	rpmpkgs
spooler.1			
anaconda.syslog	cups	mailman	rpmpkgs.1
spooler.2			
anaconda.xlog	dmesg	messages	rpmpkgs.2
spooler.3			
audit	exim	messages.1	rpmpkgs.3
spooler.4			
boot.log	gdm	messages.2	rpmpkgs.4
squid			
boot.log.1	httpd	messages.3	sa
uucp			
boot.log.2	iiim	messages.4	samba
vbox			
boot.log.3	iptraf	mysqld.log	scrollkeeper.log
vmware-tools-guestd			
boot.log.4	lastlog	news	secure
wtmp			
canna	mail	pgsql	secure.1
wtmp.1			
cron	maillog	ppp	secure.2
Xorg.0.log			
cron.1	maillog.1	prelink.log	secure.3
Xorg.0.log.old			

## Файл `/var/log/messages`

Стандартным файлом, к которому следует обратиться в первую очередь при диагностике дистрибутива от компании Red Hat (и производных дистрибутивов), является файл `/var/log/messages`. По умолчанию данный файл должен содержать информацию о событиях, которые происходят в рамках системы. Файл, выполняющий аналогичные функции в дистрибутивах Debian и Ubuntu, носит имя `/var/log/syslog`.

```
[root@RHEL4b ~]# tail /var/log/messages
Jul 30 05:13:56 anacron: anacron startup succeeded
Jul 30 05:13:56 atd: atd startup succeeded
Jul 30 05:13:57 messagebus: messagebus startup succeeded
Jul 30 05:13:57 cups-config-daemon: cups-config-daemon
startup succeeded
Jul 30 05:13:58 haldaemon: haldaemon startup succeeded
Jul 30 05:14:00 fstab-sync[3560]: removed all generated
mount points
Jul 30 05:14:01 fstab-sync[3628]: added mount point
/media/cdrom for...
Jul 30 05:14:01 fstab-sync[3646]: added mount point
/media/floppy for...
Jul 30 05:16:46 sshd(pam_unix)[3662]: session opened for
user paul by...
Jul 30 06:06:37 su(pam_unix)[3904]: session opened for
user root by paul
```

## Директория `/var/cache`

Директория `/var/cache` может содержать **кэшированные данные** некоторых приложений.

```
paul@ubu1010:~$ ls /var/cache/
apt          dictionaries-common  gdm          man
software-center
binfmts      flashplugin-installer  hald         pm-utils
cups         fontconfig           jockey       pppconfig
debconf      fonts                ldconfig     samba
```

## Директория **/var/spool**

Директория **/var/spool** обычно содержит поддиректории для хранения файлов с сообщениями электронной почты и данными задач **cron**, причем она также может быть родительской директорией для других файлов очередей (например, файлов очередей печати).

## Директория **/var/lib**

Директория **/var/lib** содержит файлы с данными состояния приложений.

Дистрибутив Red Hat Enterprise Linux, к примеру, хранит файлы, относящиеся к менеджеру пакетов **rpm**, в поддиректории **/var/lib/rpm/**.

## Другие директории **/var/...**

Директория **/var** также содержит файлы с идентификаторами процессов в поддиректории **/var/run** (которая в недалеком будущем будет заменена на директорию **/run**), временные файлы, которые не должны удаляться при перезагрузке, в поддиректории **/var/tmp**, а также файлы блокировок в поддиректории **/var/lock**. Далее в данной книге будут приведены дополнительные примеры использования директории **/var** для хранения данных.