

## Лекция 2. Пользовательский интерфейс ОС UNIX. Файловая система

### 1. Работа на компьютере с ОС UNIX

Работа пользователя с компьютером, на котором установлена ОС UNIX (Linux) начинается с регистрации. Для этого системный администратор создает в базе данных необходимую учетную запись и определяет дисциплину доступа и режим работы пользователя. Работа может производиться в трех основных режимах: терминальном, графическом и облачном.

#### 1.1. Терминальный режим

Терминальный режим работы с компьютером, управляемым ОС UNIX (Linux), подразумевает прямое/сетевое подключение терминала/компьютера пользователя к этому компьютеру (Рис. 1).

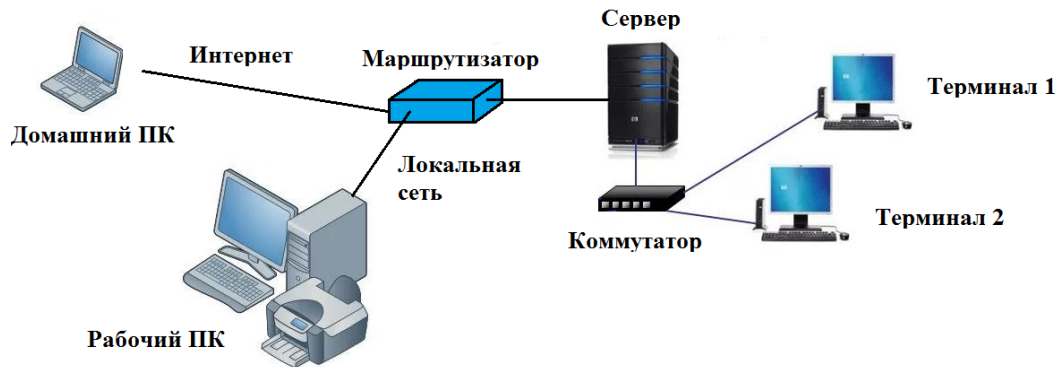


Рисунок 1. Схема подключения пользователей к компьютеру с ОС UNIX.

Работа пользователя в терминальном режиме начинается с процедуры авторизации на сервере организации с помощью программ telnet, rlogin, rsh, ssh. В случае правильности всех необходимых данных (логин – специальное имя пользователя, пароль, ключ доступа) пользователь получает удаленное терминальное окно, обслуживаемое командным процессором shell.

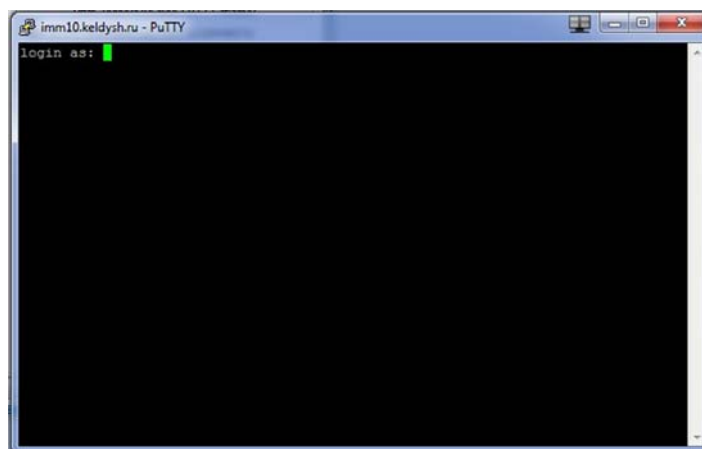


Рисунок 2. Вид удаленного терминального окна перед входом.

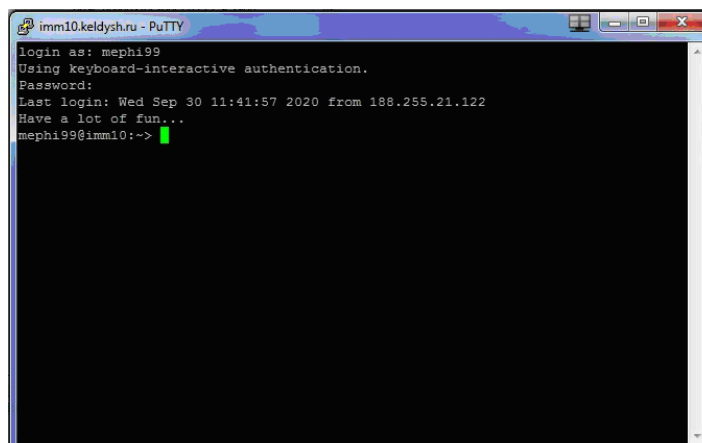


Рисунок 3. Вид удаленного терминального окна после авторизации.

## 1.2. Графический режим

Графический режим реализуется в случаях, когда пользователь работает непосредственно с UNIX-компьютером или когда удаленный интерфейс позволяет получить доступ непосредственно к графическому рабочему столу. После подключения и завершения процедуры авторизации пользователь получает графический функционал. Пример такого функционала показан на Рис. 4. В современной ситуации дальнейшая работа с графическим столом аналогична работе пользователей на компьютерах с ОС MS Windows или Apple Mac. При этом использование терминального режима реализуется путем открытия на рабочем столе отдельных терминальных окон.

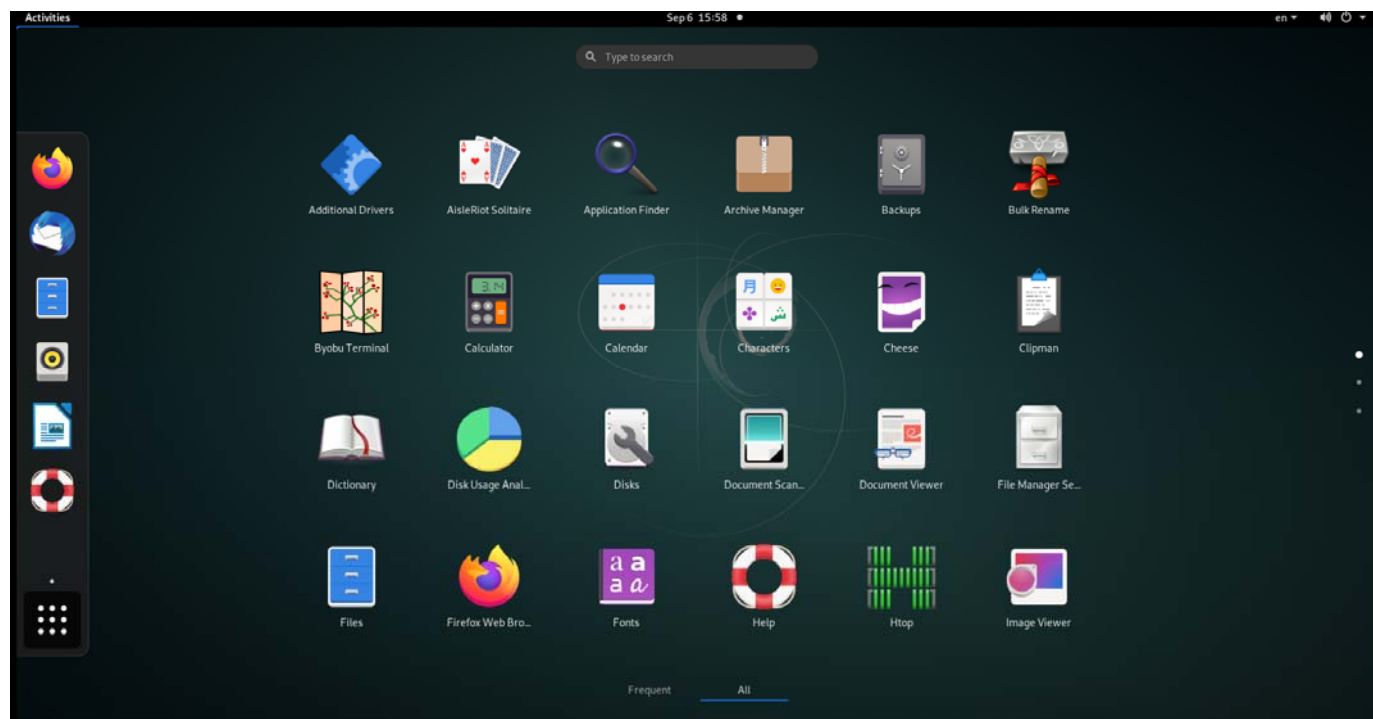


Рисунок 4. Графический рабочий стол.

## 1.3. Облачный режим

Облачный режим подразумевает чаще всего работу с UNIX-компьютером на основе веб-интерфейса с использованием альтернативного механизма авторизации. В режиме ограниченного доступа (гостевой аккаунт) авторизация обычно не требуется, и пользователь подключается к компьютеру посредством веб-браузера по протоколу HTTP. В режиме полного доступа используется протокол HTTPS и специальные процедуры авторизации с помощью секретных ключей. После подключения в первом случае пользователь может только просматривать определенные html-страницы. Во втором случае возможно работать с так называемым виртуальным графическим рабочим столом, который может быть близок по функционалу к реальному рабочему столу.

## 1.4. Программы и команды

Работа пользователя на компьютере с ОС UNIX состоит в выполнении команд и программ, а также в разработке новых программ. Дадим некоторые простые пояснения.

**Программа** – это набор инструкций для компьютера, хранящийся либо на внутреннем или внешнем носителе в виде исполняемого файла. Программы, которые могут быть выполнены компьютером без предварительной трансляции, называются исполняемыми программами или **командами**. Пользователю системы UNIX доступно множество стандартных программ и инструментальных средств (пакетов программ). Если система UNIX используется для разработки новых программ, то пользователю предоставляется базовая система программирования на языках C, Assembler, Born Shell, а также возможность использования практически любых системных вызовов (функций ядра ОС и драйверов устройств), программ и инструментальных средств, входящих в программное обеспечение ОС и прикладных сервисов. Любая написанная программа может быть переведена в разряд команд.

Базовый компонент системы UNIX образуют программы и инструментальные средства ОС, разделенные на категории функционально. Эти функции включают:

**программное окружение** – программы системы UNIX, устанавливающие дружественное программное окружение, обеспечивающее интерфейсы между системой, пользователем, системой (языками) программирования, обслуживающими программами;

**обработка и подготовка текстов** – строковый и экранный редакторы для создания и изменения текстов, программы проверки текстов на наличие ошибок и осуществляющие форматирование;

**система поддержки работы с файлами** – множество программ, которые позволяют вам создавать, организовывать и удалять файлы и каталоги;

**компоненты системы программирования** – трансляторы и интерпретаторы текстов, написанных на специализированных языках программирования (C, C++, Fortran, Python и т.д.);

**обслуживающие программы** – инструментальные средства, создающие графику и выполняющие вычисления;

**сетевые программы** – программы передачи данных другим пользователям системы и в другие системы на базе UNIX и других ОС (например, mail – почтовый клиент).

Ввод и анализ команд пользователя в терминальном режиме осуществляют совместно строковый редактор и командный процессор (shell), имеющий несколько разновидностей: sh, csh, tcsh, zcsh, bash и др.. Командная строка при этом должна иметь следующий вид (синтаксис):

**> command option(s) argument(s) <CR>**

Для каждой командной строки необходимо ввести как минимум два компонента: имя команды и клавишу <RETURN> или <ENTER>. (Обозначение <CR> используется в документации как инструкция для нажатия этой клавиши). Командная строка может также содержать ключи и аргументы. В указанном примере синтаксиса командной строки:

command – это имя программы, которую вы хотите выполнить;

option – ключи, которые указывают как запустить команду;

argument – указывает на данные, которые эта команда обрабатывает, обычно это имя каталога или файла.

В пакетном режиме используется непосредственно командный процессор, анализирующий команды при чтении так называемого пакетного файла (скрипта, файла сценария), содержащего командные строки в вышеуказанном формате. В графическом режиме функцию ввода команды может заменить клик манипулятором мышь по иконке исполняемого файла или пакетного файла.

## 1.5. Язык и команды shell

Главная задача shell – анализ команд пользователя и передача их на выполнение ядру ОС, а также обработка пакетных файлов. В последнем случае shell выступает в качестве интерпретатора команд специального языка. Язык shell содержит стандартные средства для объявления переменных и функций, организации циклов обработки данных, ветвлений сценария и т.п. Файлы со сценариями являются текстовыми и часто имеют расширение ".sh". Язык shell часто используется в UNIX-подобных системах при создании различных сценариев работы, в частности, сценариев автоматического конфигурирования исходных кодов программ перед их компиляцией.

Отличительная особенность языка shell – многие операции, которые в традиционных языках программирования являются встроенными, выполняются с помощью вызова внешних программ. Поэтому все команды shell можно подразделить на внутренние и внешние.

**Внутренние команды shell** – это команды, которые выполняются непосредственно самим командным процессором. **Внешние команды shell** – остальные команды, доступные пользователю не только через командный процессор. Внешние команды подразделяются на **системные** (реализованные в текущей версии операционной системы и доступные всем пользователям) и **пользовательские** (разработанные самим пользователем). Например, команда справки "help" является внутренней (дает справку только по shell), а команда "man sh" является внешней, так как дает справку по любой команде ОС.

Как уже сказано выше, о внутренних командах shell можно узнать с помощью системы справки:

```
>help
```

Из выведенной командой help информации можно узнать о конструкциях языка shell. По команде

```
>man sh
```

выводится информация, из которой видно, что shell это тоже команда с ключами и аргументами.

Примерами других внешних команд служат:

pwd, chdir, ls, ... – команды навигации пользователя по файловой системе и обзора содержимого конкретных каталогов;

expr используется для вычисления арифметических выражений;  
test используется для сравнения чисел и строк, а также для определения наличия или атрибутов файлов;  
sed, tr, awk, head, tail, cut и другие – для работы с текстом.

Краткая сводка часто используемых внешних команд дана в тексте Семинара 1.

## 2. Устройство пользовательских аккаунтов

Данные об учетных записях пользователей хранятся в специальных файлах базы данных ОС. Информацию о пользователе можно получить при выполнении следующих команд:

```
>cat /etc/passwd
ivanov:x:1000:1001:Ivanov Ivan Ivanovich:/home/ivanov:/bin/bash

>finger ivanov
Login: ivanov                               Name: Ivanov Ivan Ivanovich
Directory: /home/ivanov                     Shell: /bin/bash

>ls -al ~ivanov
total 28
drwxr-xr-x  3 ivanov  users 4096 Jan 20  2022 .
drwxr-xr-x 24 root    root  4096 Sep 10 19:19 ..
-rw-----  1 ivanov  users   19 Jan 20  2022 .bash_history
-rw-r--r--  1 ivanov  users  220 Feb 25  2020 .bash_logout
-rw-r--r--  1 ivanov  users 3771 Feb 25  2020 .bashrc
-rw-r--r--  1 ivanov  users  807 Feb 25  2020 .profile
```

Все данные пользователя изначально защищаются ОС. Для этого не каждый пользователь может выполнить команды finger и ls при просмотре чужих каталогов. Также пароли пользователей шифруются, и даже системный администратор не имеет доступа к содержанию паролей. Дополнительно домашние каталоги пользователей и файлы в них могут быть скрыты от чтения, записи и исполнения. Поддержка данной возможности осуществляется посредством системы прав. Она состоит в том, что все файлы в системе имеют только одного владельца (пользователя), а доступ к файлам разделен на три класса: владелец; группа, в которую входит владелец; владельцы из других групп.

Наглядно это устройство демонстрирует команда ls с ключами -al. У файлов Иванова И.И. владельцем является он сам (для этого используется его системное имя ivanov). Также Иванов входит в группу users. Права доступа для файлов Иванова показаны в левой колонке. Они кодируются тремя символами: "r", "w", "x" – обозначающими возможность чтения, записи и исполнения. Для каждого класса устанавливается своя комбинация прав. Дополнительно слева указано, является ли файл каталогом. Для этого используется символ "d".

Внутри домашнего каталога возможно создание целого дерева дочерних каталогов, где пользователь может хранить любые данные в виде файлов различного назначения и объема. Пользователь может изменять права доступа к ним посредством команды chmod.

## 3. Устройство и содержимое файловой системы

### 3.1. Стандарт иерархии файловой системы

Многие дистрибутивы UNIX и 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.

### 3.2. Страница руководства man hier

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

### 3.3. 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, находятся ниже данной корневой директории в дереве директорий. Давайте рассмотрим содержимое этой корневой директории.

```
>ls /
```

```
bin etc home lib lost+found opt run selinux sys usr boot dev lib64 media mnt proc root sbin srv tmp var
```

### 3.4. 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.

```
>ls /bin
```

```
arch          df            gzip          mkdir         pwd           spawn_login
awk           dmesg        hostname     mknod        readlink     stat
basename     dnsdomainname ifconfig     mksh         resizecons   stty
bash         domainname   initviocons  mktemp       rm           su
cat          dumpkeys    ip           more         rmdir        sync
chgrp        echo         kbinfo      mount        route        systemctl
chmod        ed           kbd_mode    mv           rpm          systemd
chown        egrep       kbdrate     netstat     screendump   systemd-ask-password
chvt         ex           keyctl      nisdomainname sed           tar
clrunimap   false       kill        openvt      setfont      tcsh
cp           fgconsole   ksh         outpsfheader setkeycodes  touch
cpio        fgrep       lksh        pdksh       settleds     tracepath
csh         fillup      ln          pgrep       setlogcons   tracepath6
date        find        loadkeys    pidof       setmetamode  true
dbus-cleanup-sockets findmnt     loadunimap  ping        setpalette   umount
dbus-daemon  fsync       logger      ping6       setvesablank uname
dbus-monitor fuser       login       pkill       setvtrgb     unicode_start
dbus-send    gawk        ls          plymouth    sh           unicode_stop
dbus-test-tool getkeycodes lsblk       ps          showconsolefont usleep
dbus-update-activation-environment getunimap  lsmod      psfaddtable showkey      vi
dbus-uuidgen grep         mail        psfgettable sleep         vim
dd          guess_encoding mapscrn     psfstrietable sort          ypdomainname
deallocvt   gunzip      md5sum      psfxtable   spawn_console zcat
```

#### 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.

## Директория /lib64

Бинарные файлы из директорий /bin и /sbin обычно используют разделяемые библиотеки, расположенные в директории /lib64. В примере ниже приведен список некоторых файлов из директории /lib64.

```
>ls /lib64/libc*  
/lib64/libc-2.26.so /lib64/libcidsn.so.1 /lib64/libcom_err.so.2.1 /lib64/libcrypt.so.1  
/lib64/libcidsn-2.26.so /lib64/libcom_err.so.2 /lib64/libcrypt-2.26.so /lib64/libc.so.6
```

## Поддиректория /lib64/modules

Обычно ядро Linux загружает модули из директории /lib64/modules/\$версия-ядра/. Содержимое этой директории будет подробно описано в главе, посвященной ядру Linux.

## 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.

## 3.5. 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 было образовано от слова etcetera (и так далее), но сегодня люди часто расшифровывают его как Editable Text Configuration (директория с редактируемыми текстовыми файлами конфигурации). 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 останова 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 содержит большое количество файлов конфигурации компонентов дистрибутива Linux. Впоследствии мы будем более подробно рассматривать некоторые из этих файлов.



### 3.6. 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:

```
>ls /home
```

```
aeix boldar2 dboykov factor informer khaitaliev nikitin polyakov sergepol usachev vikhrov  
boldar bondalex evdokiya gennadiy kataev lira olxa puzyrkov tarasov victoria
```

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.

```
>ls -d /home/factor/.*
```

```
/home/factor/. /home/factor/.bashrc /home/factor/.fonts /home/factor/.inputrc /home/factor/.xim.template  
/home/factor/.. /home/factor/.config /home/factor/.gnu-emacs /home/factor/.local  
/home/factor/.xinitrc.template /home/factor/.bash_history /home/factor/.emacs /home/factor/.i18n  
/home/factor/.profile
```

#### 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.

#### Директория /media

Директория /media служит точкой монтирования для таких устройств для работы со съемными 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. Большинство дистрибутивов Linux на сегодняшний день монтирует все съемные носители в директорию /media.

```
>ls /media/
```

```
cdrom cdrom0 usbdisk
```

#### Директория /mnt

Директория /mnt должна быть пустой и использоваться исключительно для создания временных точек монтирования файловых систем. 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.

### 3.7. 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.

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

```
>ls /dev/sd*
```

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

##### # Устройства IDE или ATAPI

```
>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.

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

К примеру, файл устройства /dev/tty1 представляет терминал или консоль, соединенную с системой. 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. Любые записанные на представленное файлом /dev/null устройство данные будут уничтожены. Представленное файлом /dev/null устройство может быть использовано для удаления ненужного вывода различных команд.

#### 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. 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).

```
>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

```

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).

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.

```
>file /proc/cpuinfo
```

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

```
>cat /proc/cpuinfo
```

```

processor       : 0
vendor_id     : GenuineIntel
cpu family    : 6
model        : 79
model name    : Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz
stepping     : 1
microcode    : 0xb000038
cpu MHz      : 2394.566
cache size   : 35840 KB
physical id  : 0
siblings: 28
core id     : 0
cpu cores  : 14
apicid     : 0
initial apicid : 0
fpu       : yes
fpu_exception : yes
cpuid level : 20
wp        : yes
flags    : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx
fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma
cx16 xtptr pdcm pcid dca sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm
abm 3dnowprefetch cpuid_fault epb cat_l3 cdp_l3 invpcid_single pti intel_ppin ssbd ibrs ibpb stibp tpr_shadow
vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm cqm rdt_a rdseed adx
smap intel_pt xsaveopt cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local dtherm ida arat pln pts md_clear
flush_lld
bugs          : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf mds swaps taa itlb_multihit
bogomips     : 4789.13
clflush size : 64
cache_alignment : 64
address sizes : 46 bits physical, 48 bits virtual
power management:

```

```

processor       : 1
vendor_id     : GenuineIntel
cpu family    : 6

```

```

model          : 79
model name    : Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz
stepping      : 1
microcode     : 0xb000038
cpu MHz       : 2394.566
cache size    : 35840 KB
physical id   : 0
siblings: 28
core id       : 1
cpu cores     : 14
apicid        : 2
initial apicid : 2
fpu           : yes
fpu_exception : yes
cpuid level   : 20
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx
fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma
cx16 xtrp pdcm pcid dca sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm
abm 3dnowprefetch cpuid_fault epb cat_l3 cdp_l3 invpcid_single pti intel_ppin ssbd ibrs ibpb stibp tpr_shadow
vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm cqm rdt_a rdseed adx
smap intel_pt xsaveopt cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local dtherm ida arat pln pts md_clear
flush_l1d
bugs          : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf mds swapsg taa itlb_multihit
bogomips     : 4789.13
clflush size  : 64
cache_alignment : 64
address sizes : 46 bits physical, 48 bits virtual
power management:
...

```

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.

### 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.

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

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.

### **Директория /usr/include**

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

### **Директория /usr/lib**

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

### **Директория /usr/local**

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

### **Директория /usr/share**

Директория /usr/share содержит независимые от архитектуры данные. Данная директория имеет значительный размер. Обычно данная директория содержит поддиректорию /usr/share/man, предназначенную для хранения файлов страниц руководств. 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).

### **Директория /usr/src**

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

### **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.

### **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.

### **Директория /var/cache**

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

### **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.