Лабораторна робота № 5.

РОБОТА В КОМАНДНОМУ РЕЖИМІ ОС LINUX

Робота в командному режимі операційної системи Linux подібна роботи в середовищі ОС Windows. Багато внутрішніх команд схожі або ідентичні командам Windows.

Для виконання запропонованого завдання необхідно спочатку завантажити операційну систему Linux.

Завдання 5.1. Створити дерево заданої структури (рис. 5.1).

Рис. 5.1. Структура до завдання 5.1

Порядок роботи:

✓ Робота з командним рядком буде проводитися в кореневому каталозі asplinux@localhost (*Домашня папка користувача asplinux на робочому столі. На вашому комп'ютері може бути й інша папка*).

1. Створення каталогу ПОРТФЕЛЬ

$ mkdir ПОРТФЕЛЬ

✓ Потрібно набрати команду й натиснути клавішу Enter. Знак $ тут і далі набирати не потрібно.

2. Перегляд змісту кореневого каталогу

$ dir

✓ Буде показаний список видимих елементів каталогу в рядку.

✓ Для одержання більш повної інформації про файли потрібно виконати таку команду:

$ dir -1

3. Створення каталогу КОМНАТА

$ mkdir КОМНАТА

4. Відкриття каталогу КОМНАТА

$ cd КОМНАТА

5. Перегляд змісту каталогу КОМНАТА
$ dir
✓ Так як каталог порожній, дана команда не дасть ніякого результату
6. Створення файлу БАМБУК.txt
$ touch БАМБУК.txt
7. Введення тексту в створений файл БАМБУК.txt

$ echo Бамбук – із сімейства мятликові, або злаки, більше відомий як рослина, що дає будівельні матеріали, але деякі його види цінуються як овочеві рослини. > БАМБУК.txt
8. Перегляд змісту каталогу КОМНАТА
$ dir
9. Створення каталогу ПОЛКА
$ mkdir ПОЛКА
10. Перегляд змісту каталогу КОМНАТА
$ dir -1
11. Відкриття каталогу ПОЛКА
$ cd ПОЛКА
12. Перегляд змісту каталогу ПОЛКА
$ dir
✓ Тому що каталог порожній, дана команда не дасть ніякого результату
13. Створення файлу ЛОТОС.txt
$ touch ЛОТОС.txt
14. Уведення тексту в створений файл ЛОТОС.txt

$ echo Лотос – із сімейства кувшинкові. Водяна рослина, у якої використовують в їжу кореневище й плоди (горішки). > ЛОТОС.txt
15. Перегляд змісту каталогу ПОЛКА
$ dir
16. Створення файлу ЯМС.txt
$ touch ЯМС.txt
17. Уведення тексту в створений файл ЯМС.txt

$ echo Ямс – із сімейства діоскорейні – клубнєносна тропічна рослина. Його високопоживні крохмалисті бульби досягають величезних розмірів (до 1 м) і маси до 50 кг. > ЯМС.txt
18. Перегляд змісту каталогу ПОЛКА
$ dir -1
✓ Створення структури завершене!

Завдання 5.2. Скопіювати файл БАМБУК.txt у каталог ПОЛКА під тим же ім’ям.

Порядок роботи:
1. Закриття каталогу ПОЛКА й перехід у батьківський для нього каталог КОМНАТА

$ cd ..
2. Копіювання файлу БАМБУК.txt
$ cp БАМБУК.txt ПОЛКА
3. Перегляд результатів копіювання
$ dir -1
4. Перехід у каталог ПОЛКА
$ cd ПОЛКА

Завдання 5.3. Скопіювати файл ЯМС.txt у каталог КОМНАТА з іменем YAMS.txt.

Порядок роботи:
1. Копіювання файлу ЯМС.txt
$ cp ЯМС.txt ..
2. Перехід у каталог КОМНАТА
$ cd ..
3. Перегляд результатів копіювання
$ dir -1
4. Перейменування файлу ЯМС.txt
$ mv ЯМС.txt YAMS.txt
5. Перегляд результатів перейменування
$ dir -1

Завдання 5.4. Перемістити файл БАМБУК.txt у каталог ПОРТФЕЛЬ із тим же іменем.

Порядок роботи:
1. Переміщення файлу БАМБУК.txt
$ mv БАМБУК.txt ../ПОРТФЕЛЬ /
✓ При переміщенні файлів символ «/» наприкінці рядка обов'язковий!
2. Перегляд результатів переміщення
$ dir -1
3. Перехід у каталог ПОРТФЕЛЬ
$ cd ../ПОРТФЕЛЬ
$ dir

Завдання 5.5. Перемістити файл YAMS.txt із каталогу КОМНАТА в каталог ПОЛКА з іменем DIOSCOREA.txt.

Порядок роботи:
1. Перехід у каталог ПОЛКА
$ cd ../КОМНАТА/ПОЛКА
2. Перегляд змісту каталогу ПОЛКА
$ dir -1
3. Переміщення файлу YAMS.txt
$ mv ../YAMS.txt DIOSCOREA.txt
4. Перегляд каталогу ПОЛКА
$ dir-1

5. Перехід у каталог КОМНАТА
$ cd ..
6. Перегляд каталогу КОМНАТА
$ dir-1

Завдання 5.6. Об'єднати файли БАМБУК.txt, ЛОТОС.txt, ЯМС.txt у каталозі ПОЛКА. Результат помістити в каталог ПОРТФЕЛЬ із іменем ОВОЧІ.txt.

Порядок роботи:
1. Перехід у каталог ПОЛКА
$ cd ПОЛКА
2. Об'єднання зазначених файлів
$ cat БАМБУК.txt ЛОТОС.txt..ЯМС.txt> ../../ПОРТФЕЛЬ /ОВОЧІ.txt
3. Перегляд результатів об'єднання
- перехід у каталог ПОРТФЕЛЬ

$ cd ../../ПОРТФЕЛЬ
- перевірка наявності файлу ОВОЧІ.txt

$ dir -1

Завдання 5.7. Переглянути вміст результуючого файлу ОВОЧІ.txt.

Порядок роботи:
$ cat ОВОЧІ.txt

Завдання 5.8. Скопіювати всі файли із каталогу ПОЛКА в каталог ПОРТФЕЛЬ.

Порядок роботи:
1. Перехід у каталог ПОЛКА
$ cd ../КОМНАТА/ПОЛКА
2. Копіювання всіх зазначених файлів у каталог ПОРТФЕЛЬ
$ cp БАМБУК.txt ЛОТОС.txt ЯМС.txt DIOSCOREA.txt ../../ПОРТФЕЛЬ /
3. Перехід у каталог ПОРТФЕЛЬ
$ cd ../../ПОРТФЕЛЬ
4. Перегляд змісту каталогу
$ dir -1

✓ Створення структури завершене.

5. Перед виконанням наступного пункту показати отриманий результат викладачеві, але перед цим необхідно перейти в кореневий каталог asplinux@localhost. Виконати самостійно.
6. Усю створену структуру можна відобразити командою.
$ ls -R

Завдання 5.9. Видалити отриману структуру

✓ Слід пам'ятати про те, що не можна видалити каталог доти, поки в ньому знаходяться файли!

Порядок роботи:

1. Видалення вмісту каталогу ПОРТФЕЛЬ.

$ rm БАМБУК.txt ЛОТОС.txt ЯМС.txt DIOSCOREA.txt

✓ Для простоти можна скористатися шаблоном для об'єднання всіх текстових файлів

$ rm *.txt

2. Перегляд результату видалення

$ dir

✓ Так як каталог порожній, дана команда не дасть ніякого результату!

3. Вихід з каталогу в кореневий каталог

$ cd ..

4. Видалення каталогу ПОРТФЕЛЬ

$ rmdir ПОРТФЕЛЬ

5. Перегляд результату видалення

$ dir -1

6. Перехід у каталог КОМНАТА

$ cd КОМНАТА

7. Перехід у каталог ПОЛКА

$ cd ПОЛКА

8. Видалення вмісту каталогу (скористаємося спрощеним записом)

$ rm *.txt

9. Перегляд результату видалення

$ dir

10. Видалення каталогу ПОЛКА

$ cd ..

$ rmdir ПОЛКА

✓ Після видалення каталогу ПОЛКА, ви автоматично перейдете в каталог КОМНАТА

11. Видалення каталогу КОМНАТА

$ cd ..

$ rmdir КОМНАТА

12. Перегляд результату видалення

$ dir

Практичні завдання для самостійної роботи

1. Виконайте таку послідовність дій:

⇒ створіть каталог DIR1 в кореневому каталозі;

⇒ створіть підкаталог DIR2 у каталозі DIR1;

⇒ створіть файли AF1.TXT, 1.DOC і BF2.FFF у DIR1;

⇒ скопіюйте файли AF1.TXT, BF2.FFF одночасно в підкаталог DIR2, заодно давши їм розширення .DOC;

⇒ перейдіть у підкаталог;

⇒ покажіть його вміст.

2. Виконайте таку послідовність дій:

⇒створіть каталог DIR3 в кореневому каталозі;

⇒створіть підкаталог DIR4 у каталозі DIR3;

⇒створіть файл AF1.TXT у DIR4;

⇒перейменуйте його в файл BF2.DOC;

⇒перейдіть у кореневий каталог;

⇒знаходячись у кореневому каталозі, покажіть вміст файла BF2.DOC.

3. Виконайте таку послідовність дій:

⇒ створіть каталог DIR1 в кореневому каталозі;

⇒ створіть підкаталог DIR2 у каталозі DIR1;

⇒ створіть файли AF1.TXT і BF2.FFF у DIR2;

⇒ об'єднайте вміст файлів AF1.TXT і BF2.FFF в AAA.DOC;

⇒ перейменуйте файл AAA.DOC у файл AFA.DOC;

⇒ перемістіть файли AF1.TXT і AFA.DOC одночасно в надкаталог DIR1;

⇒ скопіюйте файли AF1.TXT і AFA.DOC одночасно із каталогу DIR1 у каталог DIR2, заодно давши їм розширення .DDC;

⇒ перейдіть у надкаталог;

⇒ покажіть його вміст.

4. Виконайте таку послідовність дій:

⇒ створіть каталог DIR5 в кореневому каталозі;

⇒ створіть підкаталоги DIR6 і DIR7 у каталозі DIR5;

⇒ створіть файли AF1.TXT, 2.DOC і BF2.FOF у DIR5;

⇒ скопіюйте файли 2.DOC і BF2.FOF одночасно із каталогу DIR5 у каталог DIR7, заодно давши їм розширення .DDC;

⇒ створіть текстовий файл 1.TXT у каталозі DIR7;

⇒ перемістіть файли 1.TXT і 2.DDC одночасно в каталог DIR6;

⇒ видаліть каталог DIR5.

5. Виконайте таку послідовність дій:

⇒ створіть каталог DIR1 в кореневому каталозі;

⇒ створіть підкаталог DIR2 у каталозі DIR1;

⇒ створіть файл AF1.TXT у DIR2;

⇒ перейменуйте каталог DIR2 у DIR3;

⇒ перейдіть в надкаталог;

⇒ знаходячись в ньому, виведіть вміст файла AF1.TXT на екран.

Що зміниться в останній команді при виведенні вмісту файла AF1.TXT на принтер?

6. Виконайте таку послідовність дій:

⇒ створіть каталог DIR1 в кореневому каталозі;

⇒ створіть підкаталоги DIR2 і DIR3 у каталозі DIR1;

⇒ створіть файли AF1.TXT, 1.FXF і BF2.FFF у DIR2;

⇒ перейменуйте файли AF1.TXT і BF2.FFF одночасно у файли з розширенням DOC;

⇒ перемістіть їх одночасно в каталог DIR3;
⇒ покажіть його вміст.

7. Покажіть файли з розширенням EXE в активному каталозі. Що потрібно зробити, щоб показати файли з розширенням COM у будь-якому з Ваших каталогів ?

8. Виконайте таку послідовність дій:
⇒ створіть каталог DIR1 в кореневому каталозі;
⇒ створіть підкаталоги DIR2 і DIR3 у каталозі DIR1;
⇒ створіть файли AF1.TXT, 1.FXF і BF2.FFF у DIR2;
⇒ перейдіть у надкаталог;
⇒ скопіюйте файли AF1.TXT, 1.FXF одночасно у каталог DIR3;
⇒ покажіть вміст каталогу DIR3;
⇒ перейменуйте його у каталог DIR4;
⇒ перемістіть файли даного каталогу у каталог DIR1.

9. Як буде виглядати команда перейменування файлів активного каталогу в імені яких друга буква P, а четверта Z з розширенням .TXT в файли з розширенням .DOC?

10. Виконайте таку послідовність дій:
⇒ очистіть екран монітора;
⇒ створіть каталог DIR1 в кореневому каталозі;
⇒ створіть підкаталоги DIR2 і DIR3 у каталозі DIR1;
⇒ перейдіть у підкаталог в DIR3
⇒ створіть у ньому файли AF1.TXT, BF2.FFF і BF2.FXT;
⇒ знаходячись в DIR3, скопіюйте файли AF1.TXT, BF2.FXF одночасно в каталог DIR2, заодно давши їм розширення .DOC;
⇒ перейдіть у кореневий каталог;
⇒ знаходячись у ньому, покажіть вміст каталогу DIR2.

11. Виконайте таку послідовність дій:
⇒ створіть каталог DIR4 в кореневому каталозі;
⇒ створіть підкаталог DIR5 у ньому;
⇒ створіть файли AF1.TXT, BF2.FFF і ATF.DOC у DIR4;
⇒ об’єднайте вміст файлів AF1.TXT і BF2.FFF у першому із них;
⇒ скопіюйте їх одночасно в надкаталог DIR5, заодно давши їм розширення .DOC;
⇒ покажіть його вміст, спочатку зробивши його активним;
⇒ перемістіть підкаталог DIR5 у кореневий каталог.

12. Як скопіювати всі файли з розширенням .DOC активного каталогу у файли з розширенням .TXT активного каталогу?

13. Як скопіювати файли з розширенням .EXE із кореневого каталогу в активний каталог?

14. Ваші дії при зависанні комп’ютера або при неправильній роботі програм.

15. Як буде виглядати команда копіювання файлів із активного каталогу, які починаються з ABC і мають в своїх іменах не більше 5 символів з будь-яким розширенням у кореневий каталог? Що зміниться в даній команді, коли нам потрібно скопіювати ці файли в активний каталог?

Лабораторная работа № 10.
ОСНОВЫ РАБОТЫ В ОС LINUX ИЗ КОМАНДНОЙ СТРОКИ

Цель работы: изучить основы работы в ОС Linux из командной строки.

Продолжительность работы – 2 часа.

Общие сведения
Вход в систему после инсталляции

При входе в систему необходимо ввести имя пользователя и пароль. Если это первый вход в систему после ее установки, то для возможностей проведения последующих настроек входить надо под именем «root». Это единственный пользователь, обязательно определяемый во время инсталляции. Этот пользователь является полным хозяином системы, т.е. имеет неограниченный доступ к ее ресурсам, может заводить и удалять других пользователей, останавливать систему и т.д. Неосторожное поведение пользователя с такими правами легко может привести к печальным последствиям, вплоть до полного краха системы. Поэтому обычно под этим именем входят в систему только для выполнения административных задач. Но у нас сейчас как раз такой случай, так что входим под именем «root».

Консоль и виртуальные терминалы

ОС UNIX создавалась для больших компьютеров (мейнфреймов), и пользователи работали через множество последовательных интерфейсов для подключения удаленных терминалов. Терминал – это устройство, которое предназначено для взаимодействия пользователя с компьютером и состоит из монитора и клавиатуры. К вашему ПК наверняка не подключены удаленные терминалы, но есть клавиатура и монитор, выполняющие роль терминала пользователя – только в его состав добавилась мышь.

У мейнфреймов для системного администратора был особый терминал – консоль, подсоединяемый к компьютеру через специальный разъем.

Поскольку в UNIX-системах соблюдаются традиции, клавиатура и монитор ПК ведут себя так же, как раньше консоль.

Linux позволяет подключать к компьютеру удаленные терминалы и работать с виртуальными терминалами с одной консоли.

1. Нажмите комбинацию клавиш <Ctrl>+<Alt>+<F1> – вы увидите черный экран Терминала (как в MS DOS) и приглашение войти в

систему – login:. Набрав имя пользователя и введя <Enter>, получим запрос на ввод пароля:

Password (или на русском языке – Пароль):

После ввода пароля вы увидите следующую информацию (рис. 1):

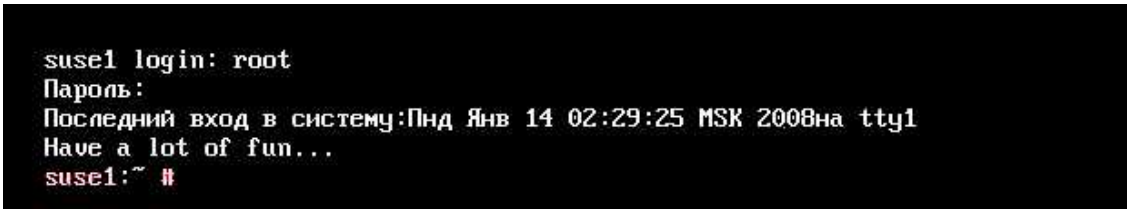

Рис. 1

Здесь tty1 – идентификатор первого терминала системы – мы вошли в него по <Ctrl>+<Alt>+<F1>, где F1 обозначает первый терминал; по <Ctrl>+<Alt>+<F2> войдем во второй терминал (tty2), ...по <Ctrl>+<Alt>+<F6> – в шестой (tty6), по <Ctrl>+<Alt>+<F7> – вернемся в графический режим.

Последняя строка называется приглашением. Появление приглашения означает, что система готова воспринять и выполнить вашу команду. Это говорит о том, что вы успешно вошли в систему. Вы видите черный экран и приглашение системы к вводу команды – то, что в MS-DOS или Windows принято называть режимом командной строки. Это – текстовый режим (в отличие от графического режима, предоставляемого системой X Window).

В приведенном примере приглашение включает в себя указание имени ПК (suse1), текущего каталога (~) и признака пользователя root – #. Вид приглашения тоже можно изменить. Обратите внимание, что приглашение для пользователя root выделено красным цветом.

Следует иметь в виду, что в любой UNIX-системе учитывается регистр символов, т.е. различаются строчные и прописные буквы. Поэтому вводить все команды и их параметры следует, учитывая регистр.

2. Первая изучаемая команда – useradd – добавляет нового пользователя. После имени команды введите пробел и имя нового пользователя, например:

Suse1:~# useradd nik

После этого система будет знать о существовании пользователя nik. Однако войти в систему под этим именем еще невозможно, т.к. каждому пользователю надо задать пароль. Для этого вводим команду

Suse1:~# passwd nik

Появится строка

Новый пароль:

Введите пароль (не менее 6 символов) и запомните его, т.к. в UNIX-системах пароль шифруется, и, если он забыт, войти в систему будет невозможно. После того как вы завершите ввод пароля нажатием клавиши <Enter>, система попросит ввести его повторно:

Повторите Новый пароль: Если вы не ошиблись при вводе (пароль приходится вводить «вслепую», поскольку он не отображается на экране), появится сообщение:

Пароль измен.

Если вы выбрали слишком короткий или простой пароль, вам будет выдано предупреждение, но система все равно примет пароль и позволит новому пользователю входить с ним в систему.

3. Следующая команда, о которой нужно знать каждому пользователю любой UNIX-системы, – это команда man. Команда man – это система встроенной помощи Linux. Вводить ее надо с параметром – именем другой команды или ключевым словом. Введите, например,

Suse1:~# man passwd

В ответ вы получите описание команды passwd или информацию по теме, обозначенной ключевым словом. Информация обычно не помещается на одном экране и при просмотре можно пользоваться клавишами <PageUp> и <PageDown>, а также клавишей пробела. Нажатие клавиши <Q> в любой момент приводит к выходу из режима просмотра и возврату в режим ввода команд. Просмотрите информацию по рассмотренным командам login и passwd. Точно так же можно получить информацию по самой команде *man*. Введите

Suse1:~# man man

К сожалению, во многих случаях информация выдается по-английски.

4. В табл. 1 перечислены некоторые команды и их параметры, которые вам следует изучить и освоить практически самостоятельно.

Таблица 1

Команда	Краткое описание
whoami	Сообщает имя, с которым вы вошли в систему в данном сеансе работы
w или who	Сообщает, какие пользователи работают в данный момент в системе
pwd	Сообщает имя текущего каталога
ls -l	Выдает список файлов и подкаталогов текущего каталога
cd <имя_каталога>	Осуществляет смену текущего каталога
ps ax	Выдает список выполняющихся процессов

Просмотрите описания этих команд с помощью команды man. Проверьте их на практике.

С остальными командами познакомимся дальше.

5. Кроме консоли, Linux позволяет подключать к ПК удаленные терминалы и, более того, обеспечивает возможность работы с несколькими виртуальными терминалами с одной консоли. Нажмите комбинацию клавиш <Ctrl>+<Alt>+<F2>. Вы снова увидите приглашение login:. Однако это не возврат к началу работы с системой – вы просто переключились в другой виртуальный терминал. Здесь вы можете зарегистрироваться под другим именем. Войдите в систему под именем только что заведенного пользователя. После этого нажмите комбинацию клавиш <Ctrl>+<Alt>+<Fl>. Вы вернетесь к первому экрану. По умолчанию Linux открывает при запуске 6 параллельных сеансов работы (виртуальных терминалов), и этим очень удобно пользоваться.

Для переключения между виртуальными терминалами используются комбинации клавиш <Ctrl>+<Alt>+<Fl> – <Ctrl>+<Alt>+<F6> (при работе в текстовом режиме тот же результат можно получить, используя комбинации клавиш <Alt>+<Fl> – <Alt>+<F6>, однако в графическом режиме без клавиши <Ctrl> не обойтись, так что лучше сразу привыкать к комбинациям из 3-х клавиш). Кстати, если в процессе работы вы забыли, в каком терминале находитесь в данный момент, воспользуйтесь командой tty, которая выводит имя терминала в следующем формате: /dev/tty2.

6. Если вы хотите завершить сеанс работы с системой в одном из терминалов, вы можете сделать это нажатием комбинации клавиш <Ctrl>+<D>. Это не приведет ни к остановке работы компьютера, ни к перезагрузке системы. Ведь Linux – многозадачная и многопользовательская система. Завершение работы одного пользователя не означает, что надо выключать компьютер. Просто завершается сеанс работы одного из пользователей, и система снова выводит в данном терминале приглашение. Можно завершить сеанс работы и введя одну из команд logout или exit.

7. Если вы хотите вернуться в графический режим, введите <Ctrl>+<Alt>+<F7>.

8. Зная теперь, как открыть и закрыть сеанс работы в системе, выполните приведенные выше рекомендации, т.е. заведите себя как рядового пользователя (без суперпользовательских прав), завершите все сеансы работы, открытые от имени root, и снова войдите в систему под своим новым именем.

9. Кратко рассмотрим командную оболочку. Оболочка, или просто shell, – это программа, которая осуществляет все общение с пользова-

телем. Именно оболочка воспринимает все команды, вводимые пользователем с клавиатуры, и организует исполнение этих команд. Поэтому оболочку можно назвать еще командным процессором. Приглашение выводит именно оболочка, ожидая ввода пользователем очередной команды. Каждый раз, когда очередной пользователь входит в систему, команда login запускает для него командный процессор – оболочку. Если вы легировались со второго терминала под именем пользователя nik, то обратите теперь внимание на различие в приглашениях у пользователей root и nik. У пользователя root приглашение оканчивается символом #, а у остальных пользователей имеет вид **nik@suse1:~>** (в других дистрибутивах для этого часто используется символ $).

Оболочку может запускать не только команда login. Вы можете просто ввести команду bash (именно так называется программа-оболочка в системе Linux) и тем самым запустить новый экземпляр оболочки. Выходя из него (по команде exit или по комбинации клавиш <Ctrl>+<D>), вы вернетесь к предыдущему экземпляру оболочки.

Оболочка bash является не только командным процессором, но и мощным языком программирования. В ней имеется целый ряд встроенных команд и операторов, и, кроме того, в качестве команды может использоваться любая программа, хранящаяся в виде файла на диске. Список встроенных команд можно получить по команде help. Попробуйте! Детальную информацию по конкретной встроенной команде выдает та же команда help с указанием в качестве параметра имени встроенной команды, например: help cd.

Для UNIX-подобных ОС разработано несколько альтернативных bash оболочек. Их можно использовать в Linux, а по умолчанию запускается bash.

10. Входить в систему под именем root без особой надобности не рекомендуется, поскольку любое неосторожное действие суперпользователя может привести к нежелательным последствиям. Входя под именем простого пользователя, вы, по крайней мере, не можете по неосторожности удалить или испортить системные файлы. В то же время, имеется ряд действий (например, монтирование файловых систем (ФС), выполнить которые может только суперпользователь. Не перезагружать же каждый раз компьютер! Именно в таких ситуациях выручает команда su. Достаточно ввести команду su, и текущая оболочка запустит для вас новый экземпляр оболочки, в который вы попадете уже с правами пользователя root. Естественно, что для этого вам придется ввести пароль суперпользователя. Закончив выполнять администраторские действия, выйдите из оболочки, и вы снова станете непривилегированным пользователем с отведенными ему полномочиями.

Если вы вошли в систему под именем root, то вы можете аналогичным образом запустить новый экземпляр оболочки от имени любого пользователя, пароль которого вы знаете. Но для этого надо указать имя этого пользователя в командной строке, например:

nik@<имя ПК>:~> su jim

Естественно, что пользователь jim должен быть зарегистрирован в системе.

Когда мы вводим su без указания имени, по умолчанию подставляется имя суперпользователя root. Но в ОС Linux есть еще одна возможность временно переключаться в root для выполнения административных функций. Ведь Linux – многопользовательская система, в ней одновременно могут работать несколько пользователей. Поэтому в первом виртуальном терминале можно войти под именем простого пользователя, а в любом другом терминале – root. Основную работу вы можете выполнять как простой пользователь, а когда потребуется выполнить административные функции, вы переключаетесь на терминал системного администратора.

Редактирование командной строки и история команд

1. Если в процессе набора команд возникли ошибки, их легко исправить. Однако существуют команды, позволяющие редактировать командную строку, а также изменять режимы работы оболочки с помощью клавиатуры (табл. 2).

Таблица 2

Клавиши	Описание реакции системы
1	2
<?> или <Ctrl>+<F>	Перемещение вправо по командной строке в пределах уже набранной цепочки символов плюс один символ справа (место для ввода следующего символа)
<?> или <Ctrl>+<B>	Перемещение на один символ влево
<Esc+<F>	Перемещение на одно слово вправо
<Esc+<B>	Перемещение на одно слово влево
<Home> или <Ctrl>+<A>	Перемещение в начало набранной цепочки символов
<End> или <Ctrl>+<E>	Перемещение в начало/конец набранной цепочки символов

Продолжение табл. 2

1	2
<Del> или <Ctrl>+<D>	Удаление символа, на который показывает курсор
<Backspase>	Удаление символа в позиции, предшествующей курсору
<Ctrl>+<K>	Удалить правую часть строки, начиная с символа, на который указывает курсор
<Ctrl>+<U>	Удалить левую часть строки, включая символ, который находится слева от курсора
<Enter> или <Ctrl>+<M>	Запуск на выполнение команды, определяемой набранной цепочкой символов
<Ctrl>+<L>	Очистить экран и поместить текущую команду в верхней строке
<Ctrl>+<T>	Поменять местами два символа: символ, на который показывает курсор, и символ слева от курсора, затем курсор переместить на один символ вправо
<Esc>+<T>	Поменять местами два слова: слово, на которое указывает курсор и слово слева от первого
<Ctrl>+<K>	Вырезать часть строки от текущей позиции курсора до конца строки (вырезанная часть строки сохраняется в буфере, ее можно вставить в другое место строки)
<Esc+<D>	Вырезать часть строки от текущей позиции курсора до конца текущего слова (если курсор указывает на пробел между словами, то вырезается все слово справа от курсора)
<Esc+<Del>	Вырезать часть строки от текущей позиции курсора до начала текущего слова (если курсор указывает на пробел между словами, то вырезается все слово слева от курсора)
<Ctrl>+<W>	Вырезать часть строки от текущей позиции курсора до предыдущего пробела
<Ctrl>+<Y>	Вставить последний вырезанный текст в позицию курсора
<Esc+<C>	Символ, на который указывает курсор, заменить на тот же, но заглавный, а курсор переместить на первый пробел справа от текущего слова

Окончание табл.2

1	2
<Esc+<U>	Сделать символы слова заглавными, начиная с символа, на который указывает курсор, а курсор установить на пробел справа от слова
<Esc+<L>	Превратить символы, начиная с символа, на который указывает курсор, до конца данного слова в прописные (маленькие) буквы, а курсор установить на пробел справа от слова
<Shift>+<PgUp> <Shift>+<PgDown>	Позволяют просмотреть несколько страниц экранного вывода (количество зависит от размера видеопамяти); полезны в тех случаях, когда та или иная команда выводит на экран очень много информации, быстро пробегающей по экрану и как бы исчезающей для пользователя; как видите, эта информация не пропадает
<Ctrl>+<C>	Прервать выполнение запущенной команды
<Ctrl>+<D>	Выход из оболочки bash

Список клавиатурных команд не ограничивается приведенными в таблице.

Для получения дополнительной информации введите команду info bash.

Проверьте все эти команды практически!

2. Оболочка bash имеет встроенную подпрограмму, предназначенную для облегчения ввода команд в командной строке. Эта подпрограмма вызывается нажатием клавиши <Tab> после того, как вы уже ввели некоторое число символов. Если эти символы являются началом названия одной из стандартных команд, которые известны оболочке, то возможны два варианта реакции оболочки на нажатие клавиши <Tab>. Если по введенным символам команда определяется однозначно, оболочка просто добавляет окончание команды в командную строку. Если однозначно восстановить имя команды по введенным символам невозможно, то выдается список возможных вариантов продолжения для того, чтобы пользователь мог ввести еще несколько символов, позволяющих однозначно завершить ввод команды нажатием клавиши <Tab>.

Если вы попробуете ввести символ табуляции в пустой командной строке, то после первого ввода вы получите только звуковой сигнал, а после второго – примерно следующее сообщение: «There are 2442 possibilities. Do you really wish to see them all? (y or n)» («Возможны 2442 варианта завершения. Вы действительно хотите увидеть их все?»).

Если ввести символ табуляции после того, как введена одна из команд и пробел, оболочка предположит, что вы ищете имя файла, который должен вводиться как параметр команды, и выдаст в качестве подсказки список файлов текущего каталога. Если же достаточная часть имени файла введена, то заканчивается ввод этого имени в командную строку. Аналогичным образом можно пытаться угадывать окончания переменных окружения, если вместо клавиши <Tab> воспользоваться комбинацией <Esc>+<$>.

3. Для практической работы с оболочкой полезно знать, что она запоминает некоторое число введенных команд (по умолчанию 1000) и позволяет вызывать их путем выбора из списка – так называемой истории команд. Историю команд можно просмотреть, введя в командной строке history (здесь вы сможете воспользоваться комбинациями клавиш <Shift>+<PgUp> и <Shift>+<PgDown>, чтобы просмотреть то, что выдаст эта команда). История команд сохраняется в файле, определяемом переменной HISTFILE. Для работы с историей команд в оболочке bash используются следующие комбинации клавиш (табл. 3). Проверьте практически!

Таблица 3

Клавиши	Описание реакции системы
<?> или <Ctrl>+<P>	Переход к предыдущей команде в списке (движение назад по списку)
<?> или <Ctrl>+<N>	Переход к следующей команде в списке (движение вперед по списку)
<PgUp>	Переход к (вызов в командную строку) самой первой команде, сохраненной в истории команд
<!>, <N>	Выполняется (без <Enter>) n-я команда из списка истории команд
<!>, <->, <N>	Выполняется n-я от конца списка команда
<!>, строка символов	Выполняется команда, имя которой начинается на строку символов (поиск нужной команды осуществляется движением в обратном порядке от конца файла истории и выполняется первая попавшаяся команда, которая начинается на строку символов)
<Ctrl>+<O>	То же, что нажатие клавиши <Enter>, затем отображается очередная команда из файла истории

Завершение работы ОС Linux

1. Хотя ПК, работающий под управлением ОС Linux, при выполнении некоторых условий можно оставлять работающим круглосуточно (прежде всего это серверы), большинство пользователей ПК привыкли выключать их после завершения работы. Если вы работаете с ОС Linux, нельзя выключать компьютер простым отключением питания. Дело в том, что в любой момент времени в системе запущено несколько процессов. Это вы можете посмотреть, выполнив команду Suse1:~# ps ax

2. Главная причина состоит в том, что некоторые из этих процессов могут работать с файлами, причем система не записывает все изменения файлов на диск сразу после внесения этих изменений пользователем или процессом, а сохраняет их временно в оперативной памяти (кэширует). Если просто выключить питание, эти изменения не будут сохранены и пропадут, что может привести даже к невозможности последующей загрузки системы. Так что надо уметь правильно завершить работу системы перед выключением компьютера. Это делается командой shutdown.

Команда shutdown может быть выполнена только пользователем root, т.е. вы либо должны войти в систему под этим именем, либо предварительно выполнить команду su, чтобы приобрести соответствующие права.

Команда shutdown имеет следующий синтаксис:

Suse1:~ # shutdown <options> <time> <warning-massage>

Существует некоторая вероятность того, что запустив команду, вы получите ответ «command not found». Это значит, что оболочка не знает, где находится файл программы. В таком случае вам необходимо ввести команду с указанием полного пути, в данном случае в виде /sbin/shutdown -h, поскольку для команды shutdown файл программы лежит в каталоге /sbin.

Из опций программы shutdown наиболее часто используются две:

- -h – полная остановка системы (компьютер будет выключен);
- -r – перезагрузить систему.

Параметр time указывает время, когда должна быть выполнена команда (необязательно выполнять ее немедленно). Время можно указать в форме задержки от текущего момента. Например, если вы хотите, чтобы система остановилась через 5 минут, вводите команду

suse1:~# shutdown -r +5,

что будет означать: «остановить систему через 5 минут и перезагрузиться после того, как работа будет корректно завершена». Для вас пока наиболее актуальной формой этой команды будет, скорее всего,

suse1:~# shutdown -h 0, когда вы захотите просто выключить компьютер. Эквивалентом команды является команда halt.

При нажатии известной комбинации клавиш <Ctrl>+<Alt>+<Del> в Linux выполняются действия, аналогичные команде shutdown -r 0.

Помощь в работе с Linux

Лучше всего искать помощь в самой системе. Дистрибутив содержит тексты документации, представленной в электронном виде. Существует несколько независимых источников, которые содержат информацию почти по любому аспекту работы в системе Linux:

- страницы интерактивного руководства man;
- гипертекстовое руководство info;
- документация, прилагаемая к пакетам ПО;
- текстовые файлы HOWTO и FAQ проекта Linux Documentation Project;
- команда locate.

Необходимо сразу сказать, что большую часть информации из этих источников вы будете получать на английском языке. Только для русифицированных дистрибутивов часть страниц интерактивного руководства man выдается на русском языке. Можно дополнительно скачать из Интернета имеющиеся там страницы руководства man, переведенные на русский язык, и разместить их в соответствующих каталогах. Но тем не менее на русский язык переведено далеко не все. Учитывая это замечание, рассмотрим каждый из перечисленных выше источников информации подробнее.

Интерактивное руководство *man*

1. С помощью команды man пользователь всегда может в затруднительной ситуации получить подсказку почти по любой команде системы, по форматам файлов и системным вызовам. Это основной способ получения подсказки во всех UNIX-системах.

Руководство man в Linux делится на секции (табл. 4).

Таблица 4

Секция	Содержание
1	2
1	Команды пользователя
8	Системные команды
2	Системные вызовы

Окончание табл. 4

1	2
3	Библиотечные вызовы
4	Устройства
5	Форматы файлов
6	Игры
7	Разное
9	Ядро (kernel internals)
л	Tcl/Tk commands

Файлы с информацией расположены в подкаталогах каталога /usr/man, и команда man ищет нужную информацию, просматривая эти подкаталоги именно в том порядке, который приведен в табл. 4. По команде man swapon получим справку о команде swapon из секции 8. Поэтому если вы хотите получить справку по системному вызову swapon, надо дать команду man 2 swapon, указывая номер секции, в которой надо искать информацию.

2. Страницы man просматриваются с помощью команды less (это дает возможность просматривать информацию поэкранно и перемещаться по этим экранам вперед и назад), так что для управления процессом вывода информации можно использовать клавиши, используемые в программе less. Наиболее употребительные приведены в следующей табл. 5.

Таблица 5

Клавиша	Назначение
<Q>	Выход из программы
<Enter>	Просмотр *строка за строкой*
<Space>	Вывод следующего экрана информации
<B>	Вернуться к предыдущему экрану
</>, за которой следует строка символов и <Enter>	Поиск введенной строки символов
<N>	Повторение предыдущего поиска

3. Для того чтобы получить необходимую информацию, нужно еще знать, что искать. В таком случае могут помочь команды whatis и apropos. Команда whatis производит контекстный поиск заданного ключевого слова (шаблона) в базе данных, содержащей перечень системных команд с кратким описанием команды. Выводятся только точные совпадения с ключевым словом. Команда apropos производит по-

иск по фрагментам слов. Аналогично команде apropos работает команда man с параметром -k. Попробуйте, например, man -k net.

Однако для того чтобы команды man -k, whatis и apropos работали, следует вначале создать базу данных о системных командах, для чего надо запустить команду makewhatis. Иначе вы можете на любой запрос получить сообщение «nothing appropriate». Правом запустить команду makewhatis обладает только root. Если вы не выключаете компьютер на ночь, то лучше всего запускать эту команду как задание для процесса **cron**.

Страницы руководства man создавались не для первоначального изучения системы, а для опытных пользователей, которым в процессе работы нужно иметь под рукой справку по формату, опциям и синтаксису команд, чтобы не приходилось держать весь этот громоздкий материал в голове.

Команда *info*

Команда info – альтернатива команде man. Для получения информации по отдельной команде надо задать в командной строке info с параметром, являющимся именем интересующей вас команды, например, info man.

Информация, которую вы увидите, в большинстве случаев несколько отличается от той, которую дает команда man, причем в лучшую сторону. Но самое существенное отличие заключается в том, что выдаваемая info информация представлена в гипертекстовом формате. В силу этого вы получаете возможность просматривать различные разделы помощи, не выходя из оболочки, предоставляемой командой info. Работая в текстовом режиме, вы можете запустить info в одной из альтернативных консолей (помните:

<Ctrl>+<Alt>+<F2>, <Ctrl>+<Alt>+<F3> и т.д.) и переключаться за помощью в случае необходимости. В тех случаях, когда вы не знаете, где именно найти нужную информацию, может оказаться полезным побродить по разным разделам текста с помощью гипертекстовых ссылок, предоставляемых командой info. Эти ссылки обозначены символом звездочки (*), что несколько отличается от способа обозначения гипертекстовых ссылок в широко распространенных браузерах типа Internet Explorer или Netscape Navigator, но от этого не становится менее удобным. Перемещаться по ссылкам можно также с помощью клавиши <Tab>. Достигнув названия нужной темы, нажмите клавишу <Enter>. Нажатие клавиши <P> возвращает вас к предыдущей странице, <N> вызывает переход на следующую страницу, а <U> переводит на один уровень вверх по иерархической структуре страниц документации.

Кроме того, можно вызвать переход по ссылке другим способом, аналогичным системе меню. Для этого надо нажать клавишу <M> и набрать в появившейся в низу экрана строке ввода некоторое число начальных символов названия нужного вам раздела помощи (из числа названий, представленных на отображаемой в данный момент на экране странице, причем даже если не вся страница помещается на экране). Число символов должно быть достаточным для однозначного определения раздела помощи (если недостаточно, программа попросит дополнить название). Выход из программы – по клавише <Q>.

Команда *help*

Введите в командной строке help без параметров, и вы получите список всех встроенных команд оболочки. Если ввести команду help name, где name – имя одной из этих команд, то вы получите очень краткую справку о применении этой команды.

Документация, поставляемая с дистрибутивом и пакетами ПО

Если в процессе установки системы вы не отказались от установки документации, то после завершения инсталляции в каталоге /usr/doc/ или /usr/share/doc вы найдете подкаталоги, содержащие обширнейшую документацию по системе Linux в целом и отдельных аспектах ее применения. Большая часть этой документации представляет собой обычные текстовые ASCII-файлы, которые можно просматривать по командам more filename или less filename, а также с помощью встроенной программы просмотра, включенной в оболочку Midnight Commander.

Просмотр этих файлов может быть для вас основным источником получения информации при освоении Linux.

К сожалению, большая часть документации написана на английском языке.

Большинство пакетов программного обеспечения поставляется разработчиками с обширной документацией по установке и использованию этих пакетов.

Если пакет представлен в формате RPM, то эта документация будет развернута в соответствующих подкаталогах каталога /usr/ share/doc. Имена этих подкаталогов соответствуют названию пакета и версии ПО. Например, для графической оболочки KDE создается подкаталог KDE.

Иногда в поиске нужного файла документации может помочь команда locate, в некотором смысле аналогичная командам whatis и apropos. По этой команде производится поиск всех файлов, имена которых содержат

заданный шаблон. Например, по команде locate net будет найдена масса имен файлов, в названиях которых встречается подстрока "net".

В шаблоне могут применяться метасимволы *, ? , [ ]. Однако команда locate производит поиск не по каталогам файловой системы, а в специально созданной базе имен файлов, которую надо вначале создать (и иногда обновлять) командой updatedb.

В некоторых дистрибутивах (например, в ALTLinux) вместо locate имеется команда slocate, которая сама создает для себя базу имен файлов (после запуска с соответствующим параметром).

Порядок выполнения работы

1. Изучите назначение и общую характеристику текстового интерфейса ОС Linux.
2. Изучите и практически освойте основные возможности консоли и виртуальных терминалов.
3. Изучите и практически освойте основные команды регистрации пользователей.
4. Изучите и практически освойте основные возможности команд man, whoami, w или who, pwd, ls –1, cd <имя_каталога>, ps ax, tty, exit, logout, su.
5. Изучите и практически освойте основные возможности средств редактирования командной строки и истории команд.
6. Изучите и практически освойте команды завершения работы в ОС Linux.
7. Изучите и практически освойте основные возможности средств справки и помощи в ОС Linux.
8. Создайте и распечатайте файл отчета с ответами на контрольные вопросы, подробным описанием и иллюстрациями этапов работы.
9. Защитите и сдайте отчет преподавателю.

Контрольные вопросы

1. Дайте общую характеристику текстового интерфейса ОС Linux.
2. Каковы возможности консоли и виртуальных терминалов?
3. Какие основные команды регистрации пользователей вы знаете?
4. Каковы возможности команд man, whoami, w или who, pwd, ls –1, cd <имя_каталога>, ps ax, tty, exit, logout, su?
5. Каковы возможности средств редактирования командной строки и истории команд?
6. Как завершить работу в ОС Linux?
7. Каковы возможности средств справки и помощи в ОС Linux?

Лабораторная работа № 11.
ИЗУЧЕНИЕ ФАЙЛОВОЙ СИСТЕМЫ ОС LINUX

Цель работы: изучить основные особенности файловой системы ОС Linux и основ работы в ней.

Продолжительность работы – 2 часа.

Общие сведения
Характеристика файловой системы ОС Linux

Файловая система (ФС) – это структура, с помощью которой ядро ОС предоставляет пользователям и процессам ресурсы долговременной памяти системы на магнитных дисках, дисках CD и DVD и т.п.

ФС обращена к пользователю (точнее – к приложениям) одной стороной. С этой стороны ФС выглядит как логическая структура каталогов и файлов.

Но у нее есть и другая сторона, образующая внутреннее устройство ФС. Эта невидимая сторона ФС устроена не просто, т.к. она реализует механизмы записи файлов на носители, алгоритмы доступа и многое другое.

ОС Linux поддерживает ряд типов ФС, в том числе MS DOS и Windows – Fat16, Fat32, NTFS, а также сетевую ФС NFS. Но есть основной на данный момент тип ФС для ОС Linux – это ext2fs и ext3fs, имеющие небольшие различия между собой. Рассмотрим именно эти файловые системы.

Большинство ФС Unix-подобных ОС сходны между собой. ФС Linux – **ext2 (ext3)**. К основным понятиям Unix подобных ФС относятся:

1. Блок загрузки (boot block).
2. Суперблок (superblock).
3. Индексный (информационный) узел (inode).
4. Блок данных (data block).
5. Блок каталога (directory block).
6. Косвенный блок (indirection block).

Блок загрузки содержит программу для первоначального запуска Unix.

В суперблоке содержится общая информация о ФС, а также информация о количестве свободных блоков и информационных узлов. Для повышения устойчивости создается несколько копий суперблока, но при монтировании используется только одна. Если первая копия суперблока повреждена, то используется его резервная копия.

В индексном (информационном) узле хранится вся информация о файле, кроме его имени. Имя файла и его дескриптор (номер информационного узла – inode) хранятся в блоке каталога. В информационном узле есть место только для хранения нескольких блоков данных. Если же нужно обеспечить большее количество, то в этом случае динамически выделяется необходимое пространство для указателей на новые блоки данных. Такие блоки называются **косвенными.** Для того чтобы найти блок данных, нужно найти его номер в косвенном блоке.

Файловая система ext2 имеет следующую структуру (рис. 1):

1. Суперблок.
2. Описатель группы.
3. Карта блоков.
4. Карта информационных узлов.
5. Таблица информационных узлов.
6. Блоки данных.

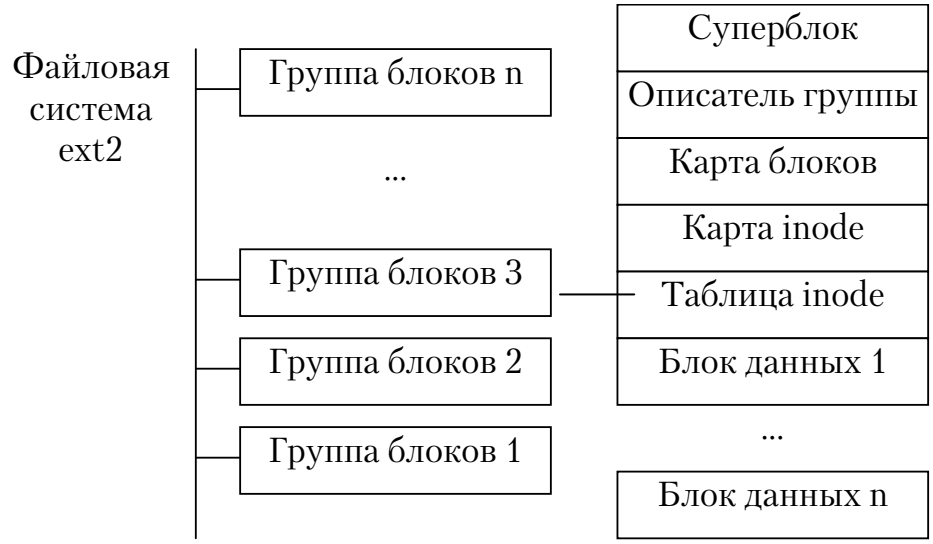

Рис. 1.

Суперблок содержит информацию обо всей ФС. Он имеется в каждой группе блоков, но это всего лишь копия суперблока первой группы блоков: так достигается избыточность ФС.

Описатель (дескриптор) *группы* содержит информацию о группе блоков. Каждая группа имеет свой дескриптор группы.

Карты блоков и *информационных узлов* – это массивы битов, которые указывают на блоки или информационные узлы соответственно.

Таблица информационных узлов содержит информацию о выделенных для данной группы блоков в информационных узлах.

Блоки данных – это блоки, содержащие реальные данные.

В ОС Linux существует четыре типа файлов:

1. Обычные файлы.
2. Ссылки (жесткие и символические).
3. Каталоги.
4. Файлы устройств.

Обычные файлы, в свою очередь, делятся на *нормальные* (текстовые) и *двоичные*.

Ссылки позволяют хранить один и тот же файл под разными именами.

Каталоги – это специальные файлы, содержащие информацию о других файлах (файлах устройств (/dev), обычных файлов и ссылок). Конечно, это довольно грубое определение, скорее интуитивное, чем точное.

Файлы устройств представляют собой устройства вашего компьютера и находятся в каталоге /dev. Например, /dev/ttyS0 – первый последовательный порт (COM1).

Свойства файловой системы ext2:

Максимальный размер файловой системы	4 Тбайт.
Максимальный размер файла	2 Гбайт.
Максимальная длина имени файла	255 символов.
Минимальный размер блока	1024 байт.
Количество выделяемых индексных дескрипторов	1 на 4 Кбайт раздела.

Файлы и их имена

С точки зрения ОС файл представляет собой непрерывный поток (или последовательность) байтов определенной длины. Внутренний формат файла ОС не интересует. Но ОС должна дать файлу имя, с помощью которого пользователь, а точнее, программы-приложения, будут обращаться к файлу. Как организовать это обращение – дело файловой системы, пользователя это чаще всего не интересует. Поэтому с точки зрения пользователя файловая система выглядит как логическая структура каталогов и файлов.

1. Имена файлов в Linux могут иметь длину до 255 символов и состоять из любых символов, кроме символа с кодом 0 и символа / (слэша). Однако имеется ряд символов, имеющих в оболочке shell специальное значение и которые поэтому не рекомендуется включать в имена. Это символы: **! @ | $ & ~ % * ( ) [ ] { } ' " \ : ; > <** – **Пробел.**

Можно также заключить имя файла или каталога с такими символами в двойные кавычки. Например, для создания каталога с именем

My old files следует использовать команду: mkdir "My old files", т.к. команда mkdir My old files создаст каталог с именем My.

Здесь и далее для упрощения команды приводятся без приглашения ОС.

Аналогичным образом можно поступать и с другими символами, перечисленными выше, т.е. их можно включать в имена файлов, если имя файла взять в двойные кавычки. Но все же предпочтительнее не использовать эти символы, включая пробел, в именах файлов и каталогов, т.к. могут возникнуть проблемы при обращении к таким файлам из некоторых приложений, а также при переносе таких файлов в другие файловые системы.

2. К точке сказанное не относится, и в Linux часто ставят более одной точки в именах файлов.

При этом теряет смысл такое понятие (принятое в DOS), как расширение имени файла, хотя все же часто последние части имени, отделенные точками, используют для обозначения файлов каких-то особых типов (например, .tar.gz используется для обозначения сжатых архивов). Но исполняемые и неисполняемые файлы в Linux распознаются не по расширениям имен файлов. Для этого существуют другие признаки, о которых мы скажем чуть позже. Точка имеет особое значение в именах файлов. Если она является первым символом имени, то данный файл считается скрытым для некоторых команд, например, он не показывается при выполнении команды ls.

3. В Linux различаются символы верхнего и нижнего регистра в именах файлов. Поэтому FILENAME.tar.gz и filename.tar.gz вполне могут существовать одновременно и являться именами разных файлов.

4. Мы привыкли считать, что файл полностью определяется его именем. Однако с точки зрения ОС и файловой системы это не так.

Каждому файлу в Linux соответствует «индексный дескриптор» файла, или «inode». Именно индексный дескриптор содержит всю необходимую файловой системе информацию о файле, включая информацию о расположении частей файла на носителе, типе файла и многое другое. Индексные дескрипторы файлов содержатся в специальной таблице (inode table), которая формируется при создании файловой системы на носителе. Каждый логический и физический диск имеет собственную таблицу индексных дескрипторов.

Дескрипторы в этой таблице пронумерованы последовательно, и именно номер дескриптора файла является его истинным именем в системе (этот номер называют индексом файла). Получить информа-

цию обо всех индексах в текущем каталоге можно с помощью команды ls -i. Однако для человека такая система имен неудобна. Поэтому файлам даются смысловые имена, и, помимо этого, файлы группируются в каталоги.

Жесткие ссылки

Приведенная выше информация нужна только для того чтобы сказать, что имя любого файла в Linux является ничем иным, как ссылкой на индексный дескриптор файла. Поэтому каждый файл может иметь сколько угодно разных имен. Эти имена называют «жесткими» ссылками. Когда вы удаляете файл, имеющий несколько разных имен – жестких ссылок, то фактически удаляется только ссылка, указанная в команде удаления файла. Даже когда вы удаляете последнюю ссылку, это еще может не означать удаления содержимого файла – если файл еще используется системой или каким-то приложением, то он сохраняется до тех пор, пока не «освободится».

Чтобы дать файлу (или каталогу) дополнительное имя (создать жесткую ссылку), используется команда ln в следующем формате:

ln <имя существующего файла> <новое_имя файла >

Примеры:

ln /home/howto/font-HOWTO-ru/Font-HOWTO.html ~/fonts.html (специальный символ ~ здесь и вообще в системе означает домашний каталог пользователя). Теперь можно вместо длинного имени /home/howto/font-HOWTO-ru/Font-HOWTO.html использовать просто ~/fonts.html.

Допустим, у нас есть файл text. Просмотрим его индекс командой ls:

ls -i text

25617 text

Теперь создадим жесткую ссылку на файл text командой ln:

ln text words

Заметьте: ссылка words на файл text имеет тот же индекс, что и файл text:

ls -i words

25617 words

Следовательно, жесткие ссылки привязываются к индексу файла. В рамках одной файловой системы вы можете организовывать только жесткие ссылки.

Командой ln можно создать множество ссылок на один файл и все они будут иметь один и тот же индекс.

Изменяя файл words, вы автоматически измените файл text. Удаляя файл words, вы можете удалить и файл text, но только в том случае, когда на него нет больше ссылок. В противном случае удалению подлежит только ссылка.

Количество ссылок отображается по команде ln -1. Число, стоящее слева от имени владельца, и есть количество ссылок. При этом доступны две ссылки на каталоги: "." – ссылка на текущий каталог, а ".." – на родительский.

Подробнее о команде ln читайте справку по команде man ln.

Символические ссылки

Выше уже говорилось о том, что файл в Linux может иметь несколько имен или «жестких» ссылок. Жесткая ссылка является просто еще одним именем для исходного файла. Она прописывается в индексном дескрипторе исходного файла. После создания жесткой ссылки невозможно различить, где исходное имя файла, а где ссылка. Если вы удаляете один из этих файлов (точнее одно из этих имен), то файл еще сохраняется на диске (пока у него есть хоть одно имя-ссылка).

Очень трудно различить первоначальное имя файла и позже созданные жесткие ссылки на него. Поэтому жесткие ссылки применяются там, где отслеживать различия и не требуется. Одно из применений жестких ссылок состоит в том, чтобы предотвратить возможность случайного удаления файла.

Особенностью жестких ссылок является то, что они прямо указывают на номер индексного дескриптора, а следовательно, такие имена могут указывать только на файлы внутри той же самой файловой системы (то есть на том же носителе, на котором находится каталог, содержащий это имя).

Но в Linux имеется другой тип ссылок – символические ссылки. Эти ссылки тоже могут рассматриваться как дополнительные имена файлов, но в то же время они представляются отдельными файлами – файлами типа символических ссылок. В отличие от жестких ссылок символические ссылки могут указывать на файлы, расположенные в другой файловой системе, например, на монтируемом носителе или, даже, на другом компьютере. Если исходный файл удален, символическая ссылка не удаляется, но становится бесполезной. Используют символические ссылки тогда, когда хотите избежать путаницы, связанной с применением жестких ссылок.

Создание любой ссылки внешне подобно копированию файла, но фактически как исходное имя файла, так и ссылка указывают на один и

тот же реальный файл на диске. Поэтому, например, если вы внесли изменения в файл, обратившись к нему под одним именем, вы обнаружите эти изменения и тогда, когда обратитесь к файлу по имени-ссылке. Чтобы создать символическую ссылку, используется уже упоминавшаяся команда ln с дополнительной опцией -s:

ln -s <имя_файла_или_каталога> <имя_ссылки>

Пример:

ln -s /home/kos/ve/HOWTO/font-HOWTO-ru/ ~/FONTS

После выполнения такой команды в домашнем каталоге пользователя появится подкаталог FONTS. Если теперь мы просмотрим список файлов в каталоге /home/kos с помощью команды ls -l, то среди прочих увидим строку:

lrwxrwxrwx 1 kos kos 31 Dec 13 21:13 FONTS -> /home/kos/ve/HOWTO/font-HOWTO-ru/

Обратите внимание на самый первый символ l в этой строке: он показывает, что данная запись соответствует символической ссылке (эти обозначения рассматриваются ниже). Впрочем, это видно и в поле имени, где после нового имени и стрелки указано исходное имя файла (в данном случае – каталога).

Если вы создали в каталоге katl символическую ссылку, которая указывает на какой-то другой каталог, то вы можете переместить каталог katl куда угодно, символическая ссылка при этом будет оставаться корректной. Точно так же можно перемещать сами символические ссылки. Но остерегайтесь использовать ".." (то есть ссылку на родительский каталог) в полных именах файлов, включающих символические ссылки, поскольку по символической ссылке нельзя проследовать в обратном направлении, а ".." всегда означает истинный родительский каталог данного каталога.

Каталоги

1. Файлы группируются в каталоги, которые, в свою очередь, могут быть включены в другие каталоги. В результате получается иерархическая структура каталогов, начинающаяся с корневого каталога. Каждый (под)каталог может содержать как отдельные файлы, так и подкаталоги.

Иерархическую структуру каталогов обычно иллюстрируют рисунком «дерева каталогов», в котором каждый каталог изображается узлом «дерева», а файлы – «листьями». В MS Windows или DOS каталоговая структура строится отдельно для каждого физического носителя (то есть имеем не отдельное «дерево», а целый «лес») и корневой каталог

каждой каталоговой структуры обозначается какой-нибудь буквой латинского алфавита (отсюда уже возникает некоторое ограничение). В Linux (и UNIX вообще) строится единая каталоговая структура для всех носителей, и единственный корневой каталог этой структуры обозначается символом «/». В эту единую каталоговую структуру можно подключить любое число каталогов, физически расположенных на разных носителях (как говорят, «смонтировать файловую систему» или «смонтировать носитель»).

2. Имена каталогов строятся по тем же правилам, что и имена файлов. Каталоги ничем, кроме своей внутренней структуры (до которой ОС уже есть дело), не отличаются от «обычных» файлов, например, текстовых.

Полным именем файла (или путем к файлу) называется список имен вложенных друг в друга подкаталогов, начинающийся с корневого каталога и оканчивающийся собственно именем файла. При этом имена подкаталогов в этом списке разделяются тем же символом /, который служит для обозначения корневого каталога. Например, /home/kos/ve/book/filesysteml.htm является полным именем файла.

3. В каждый момент времени пользователь работает с одним экземпляром командной оболочки shell, и эта оболочка хранит значение так называемого «текущего» каталога, т.е. того каталога, в котором пользователь сейчас работает. Имеется специальная команда, которая сообщает вам значение текущего каталога – pwd.

4. Кроме текущего каталога для каждого пользователя определен еще его «домашний каталог» – каталог, в котором пользователь имеет все права: может создавать и удалять файлы, менять права доступа к ним и т.д. В каталоговой структуре Linux домашние каталоги пользователей обычно размещаются в каталоге /home и имеют имена, совпадающие с именем пользователя. Например, /home/jim. Каждый пользователь может обратиться к своему домашнему каталогу с помощью значка ~, т.е. например, пользователь jim может обратиться к каталогу /home/jim/doc как к ~/doc. Когда пользователь входит в систему, текущим каталогом становится его домашний каталог.

5. Для изменения текущего каталога служит команда cd. В качестве параметра этой команде надо указать полный или относительный путь к тому каталогу, который вы хотите сделать текущим. Понятие полного пути уже было пояснено, а понятие относительного пути требует дополнительного пояснения.

Относительным путем называется перечисление тех каталогов, которые нужно пройти в «дереве каталогов», чтобы перейти от текущего каталога к какому-то другому каталогу (назовем его целевым). Если целевой каталог, т.е. каталог, который вы хотите сделать текущим, расположен ниже текущего в структуре каталогов, то сделать это просто: вы указываете сначала подкаталог текущего каталога, затем подкаталог того каталога и т.д., вплоть до имени целевого каталога. Если же целевой каталог расположен выше в каталоговой структуре или вообще на другой «ветви» дерева, то ситуация несколько сложнее. Конечно, можно было бы пользоваться полным путем, но тогда придется записывать очень длинные маршруты.

Чтобы решить этот вопрос, для каждого каталога (кроме корневого) в дереве каталогов однозначно определен «родительский каталог». В каждом каталоге имеются две особые записи. Одна из них обозначается просто точкой и является указанием на этот самый каталог, а вторая запись, обозначаемая двумя точками, – указатель на родительский каталог.

Эти имена из двух точек и используются для записи относительных путей. Чтобы сделать текущим родительский каталог, достаточно дать команду cd..

А чтобы перейти по дереву каталогов на два «этажа» вверх, откуда спуститься в подкаталог katl/kat2, надо дать команду cd . . / . ./katl/kat2.

6. Команда ls служит для вывода на экран списка имен файлов и подкаталогов текущего каталога. Нужно отметить, что фактически команда ls просто выводит содержимое файла, который описывает данный каталог, и не происходит никаких обращений к самим файлам. Любой каталог, как уже говорилось, – это обычный файл, в котором перечислены все файлы и подкаталоги этого каталога.

Если дать команду ls без параметров, то выводятся только имена файлов текущего каталога. Если нужно просмотреть содержимое не текущего, а какого-то другого каталога, надо указать команде ls полный или относительный путь к этому каталогу.

Назначение основных системных каталогов

В Linux типовая структура каталогов выдерживается более строго, чем в Windows. Более того, существует стандарт на структуру каталогов для UNIX-подобных ОС, так называемый Filesystem Hierarchy Standart (FHS), полный текст которого можно найти по адресу **http://www.pathname.com/fhs/.**

В табл. 1 дан краткий перечень основных стандартно создаваемых каталогов той файловой структуры, которая формируется при установке большинства дистрибутивов с ОС Linux.

Таблица 1

Каталог	Назначение
1	2
/bin	Этот каталог содержит в основном готовые к исполнению программы, большинство из которых необходимы во время старта системы (или в однопользовательском системном режиме для отладки). Здесь хранится значительное количество общеупотребительных команд Linux
/boot	Содержит основные постоянные файлы для загрузки системы, в частности загружаемое ядро
/dev	Каталог специальных файлов или файлов устройств – дисков, принтеров и т.п.
/etc	Содержат большинство данных, необходимых для начальной загрузки системы, и основные конфигурационные файлы. В /etc находятся, например, файл inittab, определяющий загружаемую конфигурацию и файл паролей пользователей passwd. Часть конфигурационных файлов может находиться и в /usr/etc. Каталог /etc не должен содержать двоичных файлов (их следует перенести в /bin или /sbin)
/home	Содержит каталоги пользователей
/lib	Содержит разделяемые библиотеки функций, необходимых компилятору языка С, и модули (драйверы устройств). Они загружаются в память по мере необходимости выполнения каких-то функций, что позволяет уменьшить объем кода программ – в противном случае один и тот же код многократно повторялся бы в различных программах
/lost+found	Используется при восстановлении файловой системы командой fsck. Если fsck обнаруживает файл, родительский каталог которого определить невозможно, она помещает его в каталог /lost+found. Поскольку. родительский каталог потерян, таким файлам присваиваются имена, совпадающие с номерами их индексных дескрипторов

Продолжение табл. 1

1	2
/mnt	Точка монтирования для временно монтируемых файловых систем. Если на компьютере запускается поочередно Linux и MS-DOS, то этот каталог обычно используется, чтобы монтировать файловую систему MS-DOS. Если вы имеете привычку монтировать несколько дополнительных носителей, например, дискеты, CD-ROM, дополнительный жесткий диск и т.д., то можно создать в нем соответственно дополнительные подкаталоги для каждого носителя
/proc	Точка монтирования для файловой системы proc, которая обеспечивает информацию о выполняющихся процессах, ядре, оборудовании вычислительной установки и т.д. Это псевдофайловая система, подробности о которой можно узнать по команде man 5 proc. Специальные файлы из этого каталога используются для получения и передачи данных ядру
/root	Домашний каталог суперпользователя. Он расположен не там, где располагаются личные каталоги остальных пользователей (в /home)
/sbin	Подобно каталогу /bin, содержит исполняемые файлы – программы и утилиты ОС, используемые в процессе загрузки и запускаемые системным администратором. В этот каталог надо помещать те исполняемые файлы, которые используются после успешного подключения файловой системы /usr. Минимальное содержимое этого каталога включает программы clock, getty, init, update, mkswap, swapon, swapoff, halt, reboot, shutdown, fdisk, fsck.*, mkfs.*, lilo, arp, ifconfig, route
/tmp	Каталог для временных файлов. В любой момент суперпользователь может удалить файлы из этого каталога без большого ущерба для остальных пользователей. Однако не стоит удалять файлы из этого каталога, если вам не стало ясно, что конкретный файл или группа файлов мешают продолжению продуктивной работы на машине. Система сама периодически очищает этот каталог, поэтому не следует хранить тут файлы, которые вам могут понадобиться в дальнейшем

Окончание табл. 1

1	2
/usr	Этот каталог огромен и его структура в основном повторяет структуру корневого каталога. В его подкаталогах находятся все основные приложения. В соответствии со стандартом FHS рекомендуется выделять для этого каталога отдельный раздел диска или вообще располагать его на сетевом диске, общем для всех компьютеров в сети. Такой раздел или диск монтируют только для чтения и располагают в нем общие конфигурационные и исполняемые файлы, документацию, системные утилиты и библиотеки, а также включаемые файлы (файлы типа include)
/var	Содержит файлы, в которых сохраняются различные переменные данные, определяющие конфигурацию некоторых программ при следующем запуске или временно сохраняемую информацию, которая будет использоваться позже в ходе текущего сеанса. Объем данных в этом каталоге может сильно изменяться, поскольку он содержит, например, файлы протоколов (логи), файлы спулинга и блокировки (locking), временные файлы и т.д.

Файлы физических устройств

В ОС Linux все подключаемые к компьютеру устройства (жесткие и съемные диски, терминал, принтер, модем и т.д.) представляются файлами. Если, например, надо вывести на экран какую-то информацию, то система как бы производит запись в файл /dev/tty0l.

Физические устройства бывают двух типов: символьными (или байт-ориентированными) и блочными (или блок-ориентированными). Различие между ними состоит в том, как производятся считывание и запись информации в эти устройства. Взаимодействие с символьными устройствами производится посимвольно, в режиме потока байтов. К таким устройствам относятся, например, терминалы. На блок-ориентированных устройствах информация записывается (и соответственно считывается) блоками. Примером устройств этого типа являются жесткие диски. На диск невозможно записать или считать с него один байт: обмен с диском производится только блоками.

Взаимодействием с физическими устройствами в Linux управляют драйверы устройств, которые либо встроены в ядро, либо подключаются к нему как отдельные модули. Для взаимодействия с остальными частями ОС каждый драйвер образует коммуникационный интерфейс,

который выглядит как файл. Большинство таких файлов для различных устройств как бы «заготовлены заранее» и располагаются в каталоге /dev.

Загляните в каталог /dev. Вы увидите там огромное количество файлов физических устройств. («Заглянуть в каталог» означает выполнить последовательно две команды: cd и ls.) В табл. 2 приведена небольшая справка по именам наиболее часто используемых файлов физических устройств.

Таблица 2

Имя	Значение
1	2
/dev/console	Системная консоль, т.е. монитор и клавиатура, физически подключенные к компьютеру
/dev/hd	Жесткие диски с IDE-интерфейсом. Устройство /dev/hda1 соответствует первому разделу на первом жестком диске (/dev/hda), т.е. на диске, подключенном как Primary Master
/dev/sd	Жесткие диски с SCSI-интерфейсом
/dev/fd	Файлы дисководов для гибких дисков. Первому дисководу соответствует/dev/fd0, второму/dev/fd1
/dev/tty	Файлы поддержки пользовательских консолей. Название сохранилось с тех пор, когда к системе UNIX подключались телетайпы в качестве терминалов. В Linux эти файлы устройств обеспечивают работу виртуальных консолей (переключаться между которыми можно с помощью комбинаций клавиш <Alt>+<F1>–<Alt>+<F6>)
/dev/pty	Файлы поддержки псевдотерминалов. Применяются для удаленных рабочих сессий с использованием telnet
/dev/ttS	Файлы, обеспечивающие работу с последовательными портами. /dev/ttS0 соответствует COM1 в MS-DOS, /dev/ttS1 – COM2. Если ваша мышь подключается через последовательный порт, то /dev/mouse является символической ссылкой на соответствующий /dev/ttSN
/dev/null	Это устройство – просто черная дыра. Все, что записывается в /dev/null, навсегда потеряно. На это устройство можно перенаправить вывод ненужных сообщений. Если /dev/null используется как устройство ввода, то оно ведет себя как файл нулевой длины

Окончание табл. 2

1	2
/dev/mouse	Мышь
/dev/audio	Звуковая карта
/dev/modem	Модем. Обычно /dev/modem – ссылка на один из файлов /dev/ttyS0
/dev/lpN	Параллельный порт
/dev/ttySN	Последовательный порт. /dev/ttyS0 аналогичен файлу COM4 в DOS
/dev/mdN	Массив RAID
/dev/ethN	Сетевая плата

На устройствах hdxN и sdxN необходимо остановиться подробнее. Известно, что к (E)IDE (ATA) контроллеру можно подключить четыре IDE-устройства: Primary Master, Primary Slave, Secondary Master, Secondary Slave.

Этим устройствам соответствуют символы: a, b, c, d. Например, /dev/hda – Primary Master, a /dev/hdd – Secondary Slave. Номер N в обозначении устройства обозначает номер раздела на жестком диске. Первичный раздел DOS на первом жестком диске обозначается так: /dev/hdal.

Права доступа к файлам и каталогам

1. Поскольку Linux – система многопользовательская, организация разграничения доступа к файлам и каталогам является одним из существенных вопросов, которые должна решать ОС. Механизмы разграничения доступа, разработанные для системы UNIX, очень просты, но они оказались настолько эффективными, что просуществовали уже более 30 лет и по сей день успешно выполняют стоящие перед ними задачи.

2. В основе механизмов разграничения доступа лежат имена пользователей и имена групп пользователей. В Linux каждый пользователь имеет уникальное имя, под которым он входит в систему. Кроме того, в системе создается некоторое число групп пользователей, причем каждый пользователь может быть включен в одну или несколько групп.

Создает и удаляет группы суперпользователь, он же может изменять состав участников той или иной группы. Члены разных групп могут иметь разные права по доступу к файлам, например, группа администраторов может иметь больше прав, чем группа программистов.

3. В индексном дескрипторе каждого файла записаны имя так называемого владельца файла и группы, которая имеет права на этот файл. При создании файла его владельцем объявляется тот пользователь, который этот файл создал. Точнее – тот пользователь, от чьего имени запущен процесс, создающий файл. Группа тоже назначается при создании файла – по идентификатору группы процесса, создающего файл. Владельца и группу файла можно поменять в ходе дальнейшей работы с помощью команд chown и chgrp.

4. Выполним команду ls -l и зададим ей в качестве дополнительного параметра имя конкретного файла, например, файла, задающего саму команду ls. (Обратите внимание на эту возможность команды ls -l – получить информацию о конкретном файле, а не о всех файлах каталога сразу).

ls -1 /bin/ls

-rwxr-xr-x 1 root root 49940 Sep 12 1999 /bin/ls

Вы видите, что в данном случае владельцем файла являются пользователь root и группа root. Но нас сейчас больше интересует первое поле, определяющее тип файла и права доступа к файлу. Это поле в приведенном примере представлено цепочкой символов -rwxr-xr-x. Эти символы можно условно разделить на 4 группы.

Первая группа, состоящая из единственного символа, определяет тип файла. Этот символ в соответствии с возможными типами файлов может принимать такие значения:

- – обычный файл;
d – каталог;
b – файл блочного устройства;
c – файл символьного устройства;
s – доменное гнездо (socket);
p – именованный канал (pipe);
l – символическая ссылка (link).

Далее следуют три группы по три символа, которые и определяют права доступа к файлу соответственно для владельца файла, для группы пользователей, которая сопоставлена с данным файлом, и для всех остальных пользователей системы. В нашем примере права доступа для владельца определены как rwx, что означает – владелец (root) имеет право читать файл (r), производить запись в этот файл (w) и запускать файл на выполнение (x).

Замена любого из этих символов прочерком будет означать, что пользователь лишается соответствующего права. В том же примере мы видим, что все остальные пользователи (включая и тех, которые вошли

в группу root) лишены права записи в этот файл, т.е. не могут файл редактировать и вообще как-то изменять.

Вообще говоря, права доступа и информация о типе файла в UNIX-системах хранятся в индексных дескрипторах в отдельной структуре, состоящей из двух байтов, т.е. из 16 бит (это естественно, ведь компьютер оперирует битами, а не символами r, w, x).

4 бит из этих 16 отведены для кодированной записи о типе файла.

Следующие 3 бит задают особые свойства исполняемых файлов, о которых мы скажем чуть позже.

И, наконец, оставшиеся 9 бит определяют права доступа к файлу. Эти 9 бит разделяются на 3 группы по 3 бит. Первые 3 бит задают права пользователя, следующие 3 бит – права группы, последние 3 бит определяют права всех остальных пользователей (то есть всех пользователей, за исключением владельца файла и группы файла).

При этом если соответствующий бит = 1, право предоставляется, а если он = 0, то нет. В символьной форме записи прав единица заменяется соответствующим символом(r, w или x), а 0 представляется прочерком.

Право на чтение (r) файла означает, что пользователь может просматривать содержимое файла с помощью различных команд просмотра, например, командой more или с помощью любого текстового редактора. Но, подредактировав содержимое файла в текстовом редакторе, вы не сможете сохранить изменения в файле на диске, если не имеете права на запись (w) в этот файл. Право на выполнение (x) означает, что вы можете загрузить файл в память и попытаться запустить его на выполнение как исполняемую программу. Конечно, если в действительности файл не является программой (или скриптом shell), то запустить этот файл на выполнение не удастся, но, даже если файл действительно является программой, но право на выполнение для него не установлено, то он тоже не запустится.

Вот мы и узнали, какие файлы в Linux являются исполняемыми! Как видите, расширение имени файла тут ни при чем, все определяется установкой атрибута «исполняемый», причем право на исполнение может быть предоставлено не всем!

5. Если выполнить ту же команду ls -l, но в качестве последнего аргумента ей указать не имя файла, а имя каталога, мы увидим, что для каталогов тоже определены права доступа, причем они задаются теми же самыми символами rwx. Например, выполнив команду ls -l /, мы увидим, что каталогу bin соответствует строка:

drwxr-xr-x 2 root root 2048 Jun 21 21:11 bin

По отношению к каталогам трактовка понятий «право на чтение», «право на запись» и «право на выполнение» несколько изменяется.

Право на чтение по отношению к каталогам легко понять, если вспомнить, что каталог – это просто файл, содержащий список файлов в данном каталоге. Следовательно, если вы имеете право на чтение каталога, то вы можете просматривать его содержимое (этот самый список файлов в каталоге). Право на запись тоже понятно – имея такое право, вы сможете создавать и удалять файлы в этом каталоге, т.е. просто добавлять в каталог или удалять из него запись, содержащую имя какого-то файла и соответствующие ссылки.

Право на выполнение интуитивно менее понятно. Оно в данном случае означает право переходить в этот каталог. Если вы, как владелец, хотите дать доступ другим пользователям на просмотр какого-то файла в своем каталоге, вы должны дать им право доступа в каталог, т.е. дать им «право на выполнение каталога». Более того, надо дать пользователю право на выполнение для всех каталогов, стоящих в дереве выше данного каталога. Поэтому в принципе для всех каталогов по умолчанию устанавливается право на выполнение как для владельца и группы, так и для всех остальных пользователей. И если вы хотите закрыть доступ в каталог, то лишите всех пользователей (включая группу) права входить в этот каталог. Только не лишайте и себя такого права, а то придется обращаться к суперпользователю!

После прочтения предыдущего абзаца может показаться, что право на чтение каталога не дает ничего нового по сравнению с правом на выполнение. Однако разница в этих правах все же есть. Если задать только право на выполнение, вы сможете войти в каталог, но не увидите там ни одного файла (этот эффект особенно наглядно проявляется в том случае, если вы пользуетесь каким-то файловым менеджером, например, программой Midnight Commander). Если вы имеете право доступа в каком-то из подкаталогов этого каталога, то вы можете перейти в него (командой cd), но, как говорится «вслепую», по памяти, потому что списка файлов и подкаталогов текущего каталога вы не увидите.

6. Алгоритм проверки прав пользователя при обращении к файлу можно описать следующим образом. Система вначале проверяет, совпадает ли имя пользователя с именем владельца файла. Если эти имена совпадают (то есть владелец обращается к своему файлу), то проверяется, имеет ли владелец соответствующее право доступа: на чтение, на запись или на выполнение (суперпользователь может лишить некоторых прав и владельца файла). Если право такое есть, то соответствующая операция разрешается. Если же нужного права владелец не имеет, то проверка прав, предоставляемых через группу или через группу атрибутов доступа для остальных пользователей, уже даже не осуществ-

ляется, а пользователю выдается сообщение о невозможности выполнения затребованного действия (что-то вроде «Permission denied»).

Если имя пользователя, обращающегося к файлу, не совпадает с именем владельца, то система проверяет принадлежность владельца к группе, сопоставленной с данным файлом (ее называют группой файла). Если принадлежит, то для определения возможности доступа к файлу используются атрибуты, относящиеся к группе, а на атрибуты для владельца и всех остальных пользователей внимания не обращается. Если пользователь не является владельцем файла и не входит в группу файла, то его права определяются атрибутами для остальных пользователей. Следовательно, группа атрибутов, определяющих права доступа к файлу, относится ко всем пользователям, кроме владельца файла и пользователей, входящих в группу файла.

7. Для изменения прав доступа к файлу используется команда chmod. Ее можно использовать в двух вариантах. В первом варианте вы должны явно указать, кому какое право даете или кого этого права лишаете:

chmod wXp имя-файла,

где вместо символа **w** подставляется:

- либо символ **u** (то есть пользователь, который является владельцем);
- либо **g** (группа);
- либо **o** (все пользователи, не входящие в группу, которой принадлежит данный файл);
- либо **a** (все пользователи системы, т.е. и владелец, и группа, и все остальные).

Вместо X ставится:

- либо **+** (предоставляем право);
- либо **-** (лишаем соответствующего права);
- либо **=** (установить указанные права вместо имеющихся).

Вместо **p** – символ, обозначающий соответствующее право:

- **r** (чтение);
- **w** (запись);
- **x** (выполнение).

Вот несколько примеров использования команды chmod:

chmod a+x file_name – предоставляет всем пользователям системы право на выполнение данного файла.

chmod go-rw file_name – удаляет право на чтение и запись для всех, кроме владельца файла.

chmod ugo+rwx file_name – дает всем права на чтение, запись и выполнение.

Если опустить указание на то, кому предоставляется данное право, то подразумевается, что речь идет вообще обо всех пользователях, т.е. вместо chmod a+x file_name можно записать просто chmod +x file_name

Второй вариант задания команды chmod (он используется чаще) основан на цифровом представлении прав. Для этого мы кодируем символ r цифрой 4, символ w – цифрой 2, а символ x – цифрой 1. Для того чтобы предоставить пользователям какой-то набор прав, надо сложить соответствующие цифры.

Получив, таким образом, нужные цифровые значения для владельца файла, для группы файла и для всех остальных пользователей, задаем эти три цифры в качестве аргумента команды chmod (ставим эти цифры после имени команды перед вторым аргументом, который задает имя файла). Например, если надо дать все права владельцу (4+2+1=7), право на чтение и запись группе (4+2=6) и не давать никаких прав остальным, то следует дать такую команду:

chmod 760 file_name

Если вы знакомы с двоичным кодированием восьмеричных цифр, то поймете, что цифры после имени команды в этой форме ее представления есть не что иное, как восьмеричная запись тех самых 9 бит, которые задают права для владельца файла, группы файла и для всех пользователей.

Выполнять смену прав доступа к файлу с помощью команды chmod может только сам владелец файла или суперпользователь. Для того чтобы иметь возможность изменить права группы, владелец должен дополнительно быть членом той группы, которой он хочет дать права на данный файл.

8. Команда chmod может устанавливать еще три возможных атрибута файла, которые в его индексном дескрипторе в двухбайтовой структуре прав на файл занимают позиции 5–7, сразу после кода типа файла.

Первый из этих атрибутов – так называемый бит смены идентификатора пользователя. Смысл этого бита состоит в следующем. Обычно, когда пользователь запускает некоторую программу на выполнение, эта программа получает те же права доступа к файлам и каталогам, которые имеет пользователь, запустивший программу. Если же установлен бит смены идентификатора пользователя, то программа получит права доступа к файлам и каталогам, которые имеет владелец файла программы (таким образом, рассматриваемый атрибут лучше называть битом смены идентификатора владельца). Это позволяет решать некоторые задачи, которые иначе было бы трудно выполнить. Самый характерный пример – команда смены пароля passwd. Все пароли пользователей хранятся в файле /etc/passwd, владельцем которого является суперпользователь root. Поэтому программы, запущенные обычными

пользователями, в том числе команда passwd, не могут производить запись в этот файл, а значит, пользователь как бы не может менять свой собственный пароль. Но для файла /usr/bin/passwd установлен бит смены идентификатора владельца, каковым является пользователь root. Следовательно, программа смены пароля passwd запускается с правами root и получает право записи в файл /etc/passwd (уже средствами самой программы обеспечивается то, что пользователь может изменить только одну строку в этом файле).

Установить бит смены идентификатора владельца может суперпользователь с помощью команды chmod +s file_name

Аналогичным образом работает бит смены идентификатора группы.

Еще один возможный атрибут исполняемого файла – это бит сохранения задачи, или sticky bit (дословно – «бит прилипчивости»). Этот бит указывает системе, что после завершения программы надо сохранить ее в оперативной памяти. Удобно включить этот бит для задач, которые часто вызываются на выполнение, т.к. в этом случае экономится время на загрузку программы при каждом новом запуске. Этот атрибут был необходим на старых компьютерах. На современных быстродействующих системах он используется редко.

Если используется цифровой вариант задания атрибутов в команде chmod, то цифровое значение этих атрибутов должно предшествовать цифрам, задающим права пользователя: chmod 4775 file_name

При этом веса этих битов для получения нужного суммарного результата задаются следующим образом:

- 4 – бит смены идентификатора пользователя;
- 2 – бит смены идентификатора группы;
- 1 – бит сохранения задачи (sticky bit).

Если какие-то из этих 3 бит установлены в 1, то несколько изменяется вывод команды ls -l в части отображения установленных атрибутов прав доступа. Если установлен в 1 бит смены идентификатора пользователя, то символ x в группе, определяющей права владельца файла, заменяется символом s. Причем, если владелец имеет право на выполнение файла, то символ x заменяется на маленькое s, а если владелец не имеет права на выполнение файла (например, файл вообще неисполняемый), то вместо x ставится S.

Аналогичные замены имеют место при задании бита смены идентификатора группы, но заменяется символ x в группе атрибутов, задающих права группы.

Если бит сохранения задачи (sticky bit) равен 1, то заменяется символ x в группе атрибутов, определяющей права для всех остальных пользователей, причем x заменяется символом t, если все пользователи могут запускать файл на выполнение, и символом T, если они такого права не имеют.

Таким образом, хотя в выводе команды ls -l не предусмотрено отдельных позиций для отображения значений битов смены идентификаторов и бита сохранения задачи, соответствующая информация выводится. Вот небольшой пример того, как это будет выглядеть:

```
ls -l priml
-rwSrwsrwT 1 kos root 12 Dec 18 23:17 priml
```

Порядок выполнения работы

1. Изучите назначение и характеристику файловой системы ОС Linux.
2. Изучите структуру файловой системы ОС Linux.
3. Изучите основные типы файлов ОС Linux.
4. Изучите и практически освойте правила определения имен файлов в Linux.
5. Изучите и практически освойте назначение и правила определения жестких ссылок.
6. Изучите и практически освойте назначение и правила определения символических ссылок.
7. Изучите и практически освойте типовую иерархическую структуру каталогов ОС Linux.
8. Изучите и практически освойте назначение и общую характеристику файлов физических устройств.
9. Изучите и практически освойте систему и методы определения прав доступа к файлам в ОС Linux.
10. Создайте и распечатайте файл отчета с ответами на контрольные вопросы, подробным описанием и иллюстрациями этапов работы.
11. Защитите и сдайте отчет преподавателю.

Контрольные вопросы

1. Дайте общую характеристику файловой системы ОС Linux.
2. Сравните файловые системы ОС Linux и Windows.
3. Какие типы файлов ОС Linux вы знаете?
4. Каковы правила определения имен файлов в Linux?
5. Что такое жесткие ссылки и как они реализуются?
6. Что такое символические ссылки и как они реализуются?
7. Охарактеризуйте типовую иерархическую структуру каталогов ОС Linux.
8. Дайте характеристику файлов физических устройств.
9. Как определяются права доступа к файлам в ОС Linux?

Лабораторная работа № 12.
ИЗУЧЕНИЕ КОМАНД ДЛЯ РАБОТЫ С ФАЙЛАМИ ИЗ КОМАНДНОЙ СТРОКИ В ОС LINUX

Цель работы: изучить основные команды для работы с файлами из командной строки в ОС Linux.

Продолжительность работы – 2 часа.

Общие сведения
Команды для работы с файлами

В табл. 1 приведены основные команды ОС Linux, предназначенные для работы с файлами.

Таблица 1

Команда	Назначение
touch <имя файла>	Создает пустой файл
cat <имя файла>	Просмотр текстового файла
tac <имя файла>	Вывод содержимого текстового файла в обратном порядке, т.е. сначала выводится последняя строка, потом предпоследняя и т.д.
cp <файл1> <файл2>	Копирует файл <файл1> в файл <файл2>. Если <файл2> существует, программа попросит разрешения на его перезапись
mv <файл1> <файл2>	Перемещает файл <файл1> в файл <файл2>. Эта же команда пригодна и для переименования файла
rm <файл>	Удаляет файл
locate <файл>	Выполняет быстрый поиск файла
which <программа>	Выводит каталог, в котором находится программа, если она вообще установлена. Поиск производится в каталогах, указанных в переменной окружения PATH (это путь поиска программ)
less <файл>	Служит для удобного просмотра файла с возможностью скроллинга (постраничной прокрутки)
chmod <опции> <файл >	Используется для изменения прав доступа к файлу

Рассмотрим несколько часто используемых команд для работы с файлами.

Команды *chown* и *chgrp*

Эти команды служат для смены владельца файла и группы файла. Выполнять смену владельца может только суперпользователь, смену группы может выполнить сам владелец файла или суперпользователь. Чтобы иметь право сменить группу, владелец должен дополнительно быть членом той группы, которой он хочет дать права на данный файл. Формат этих двух команд аналогичен:

chown vasja имя-файла

chgrp usersgrp имя-файла

Команда *mkdir*

Команда mkdir позволяет создать подкаталог в текущем каталоге. В качестве аргумента этой команде надо дать имя создаваемого каталога. Во вновь созданном каталоге автоматически создаются две записи: **.** (ссылка на этот самый каталог) и **..** (ссылка на родительский каталог). Чтобы создать подкаталог, вы должны иметь в текущем каталоге право записи. Можно создать подкаталог не в текущем, а в каком-то другом каталоге, но тогда необходимо указать путь к создаваемому каталогу:

mkdir /home/kos/book/glava5/partl

Команда mkdir может использоваться со следующими опциями:

-m mode – задает режим доступа для нового каталога (например, -m 755);

-p – создать указанные промежуточные каталоги (если они не существуют).

Команда *cat*

Команда cat часто используется для создания файлов (хотя можно воспользоваться и командой touch). По команде cat на стандартный вывод (то есть на экран) выводится содержимое указанного файла (или нескольких файлов, если их имена последовательно задать в качестве аргументов команды).

Если вывод команды cat перенаправить в файл, то можно получить копию какого-то файла:

cat filel > file2

Первоначальное предназначение команды cat как раз и предполагало перенаправление вывода, т.к. эта команда создана для конкатенации, т.е. объединения нескольких файлов в один:

cat filel file2 ... fileN > new-file

Именно возможности перенаправления ввода и вывода этой команды и используются для создания новых файлов. Для этого на вход ко-

манды cat направляют данные со стандартного ввода (то есть с клавиатуры), а вывод команды – в новый файл:

cat > newfile

Когда вы напечатаете все, что хотите, нажмите комбинацию клавиш <Ctrl>+<D> или <Ctrl>+<C>, и все, что вы ввели, будет записано в newfiie. Конечно, таким образом создаются в основном короткие текстовые файлы.

Команда *cp*

Хотя для копирования файлов иногда пользуются командой cat, в Linux для этого есть специальная команда cp. Ее можно применять в одной из двух форм:

cp [options] source destination

cp [options] source_directory new_directory

В первом случае файл или каталог source копируется соответственно в файл или каталог destination, а во втором случае файлы, содержащиеся в каталоге source_directory, копируются в каталог new_directory. Для копирования надо иметь права на чтение файлов, которые копируются, и права на запись в каталог, в который производится копирование.

Если в качестве целевого указывается существующий файл, его содержимое будет затерто, поэтому при копировании надо соблюдать осторожность.

Впрочем, можно использовать команду cp с опцией -i, тогда перед перезаписью существующего файла будет запрашиваться подтверждение.

У команды cp имеется еще несколько полезных опций (табл. 2).

Таблица 2

Опция	Значение
-p	Сохраняет время модификации файла и максимально возможные полномочия. Без этой опции для нового файла задаются полномочия, соответствующие полномочиям запустившего команду пользователя
-R или -r	Если source – каталог, то копируется как он, так и все входящие в него подкаталоги, т.е. сохраняется исходная форма дерева каталогов
-d	Если задать эту опцию, то символические ссылки будут оставаться ссылками (а иначе вместо ссылки копируется файл, на который дается ссылка)
-f	Перезаписывать файлы при копировании (если такие уже есть) без дополнительных предупреждений

Команда *mv*

Если вам необходимо не скопировать, а переместить файл из одного каталога в другой, вы можете воспользоваться командой mv. Синтаксис этой команды аналогичен синтаксису команды cp. Более того, она сначала копирует файл (или каталог), а только потом удаляет исходный файл (каталог). И опции у нее такие же, как у cp.

Команда mv может использоваться не только для перемещения, но и для переименования файлов и каталогов (то есть перемещения их внутри одного каталога). Для этого надо просто задать в качестве аргументов старое и новое имя файла:

mv oldname newname

Учтите, что команда mv не позволяет переименовать сразу несколько файлов (используя шаблон имени), т.е. команда mv *.xxx *.yyy не будет работать.

При использовании команды mv, так же как и при использовании cp, не забывайте применять опцию -i для того, чтобы получить предупреждение, когда файл будет перезаписываться.

Команды *rm* и *rmdir*

Для удаления ненужных файлов и каталогов в Linux служат команды rm (удаляет файлы) и rmdir (удаляет пустой каталог). Для того чтобы воспользоваться этими командами, вы должны иметь право записи в каталоге, в котором расположены удаляемые файлы или каталоги. При этом полномочия на изменение самих файлов необязательны. Если хотите перед удалением файла получить дополнительный запрос на подтверждение операции, используйте опцию -i.

Если вы попытаетесь использовать команду rm (без всяких опций) для удаления каталога, то будет выдано сообщение, что это каталог, и удаления не произойдет. Для удаления каталога надо удалить в нем все файлы, после чего удалить сам каталог с помощью команды rmdir. Однако можно удалить и непустой каталог со всеми входящими в него подкаталогами и файлами, если использовать команду rm с опцией -r.

Если вы дадите команду rm *, то удалите все файлы в текущем каталоге. Подкаталоги при этом не удалятся. Для удаления как файлов, так и подкаталогов текущего каталога надо тоже воспользоваться опцией -r. Однако всегда помните, что в Linux нет команды восстановления файлов после их удаления (даже если вы спохватились сразу же после ошибочного удаления файла или каталога)!

Так что дважды подумайте до удаления чего-либо и не пренебрегайте опцией -i.

Команды *more* и *less*

Команда cat позволяет выдать на стандартный вывод (на экран) содержимое любого файла, однако она используется для этих целей очень редко, разве что для вывода очень небольших по объему файлов. Дело в том, что содержимое большого файла мгновенно проскакивает на экране, и пользователь видит только последние строки файла. Поэтому cat используется в основном по ее прямому назначению – для конкатенации файлов, а для просмотра содержимого файлов (конечно, текстовых) используются команды more и less (или текстовые редакторы).

Команда-фильтр more выводит содержимое файла на экран отдельными страницами размером как раз в целый экран. Для того чтобы увидеть следующую страницу, надо нажать на клавишу пробела. Нажатие на клавишу <Enter> приводит к смещению на одну строку. Кроме клавиш пробела и <Enter> в режиме паузы еще некоторые клавиши действуют как управляющие (например, клавиша <B> возвращает вас на один экран назад), но мы здесь не будем приводить полного их перечня, как и перечня опций команды. Вам для начала надо еще только запомнить, что выйти из режима просмотра можно с помощью клавиши <Q>, т.к. если вы этого не знаете, то вам придется долго и нудно нажимать пробел, пока вы не доберетесь до конца длинного файла. Обо всех опциях команды more вы можете прочитать в интерактивных руководствах man или info.

Утилита less, разработанная в рамках проекта GNU, содержит все функции и команды управления выводом, имеющиеся в программе more, и некоторые дополнительные, например, позволяет использовать клавиши управления курсором (<i>, <t>, <PgUp>, <PgDown>) для перемещения по тексту.

Команды more и less позволяют производить поиск подстроки в просматриваемом файле, причем команда less позволяет производить поиск как в прямом, так и в обратном направлении. Для организации поиска строки символов string надо набрать в командной строке программы в нижней части экрана (там, где двоеточие) /string. Если искомая строка найдётся, будет отображен соответствующий фрагмент текста, причем найденная строка будет находиться в самом верху экрана.

Команда *find* и символы шаблонов для имен файлов

Одной из часто используемых команд для работы с файлами в Linux является команда поиска нужного файла find. Команда find может искать файлы по имени, размеру, дате создания или модификации и некоторым другим критериям.

Общий синтаксис команды find имеет следующий вид:

find [список_каталогов] критерий_поиска

Параметр *список_каталогов* определяет, где искать нужный файл. Проще всего задать в качестве начального каталога поиска корневой каталог /, однако, в таком случае поиск может затянуться очень надолго, т.к. будет просматриваться вся структура каталогов, включая смонтированные файловые системы (в том числе сетевые, если таковые есть). Можно сократить объем поиска, если задать вместо одного корневого каталога список из нескольких каталогов (естественно, тех, в которых может находиться искомый файл):

find /usr/share/doc /usr/doc /usr/locale/doc -name instr.txt

Началом "критерия_поиска", определяющего, что именно должна искать программа find, считается первый аргумент, начинающийся на один из символов: – ( ) , !. Все аргументы, предшествующие "критерию_поиска", трактуются как имена каталогов, в которых надо производить поиск. Если не указано ни одного пути, поиск производится только в текущем каталоге и его подкаталогах.

Чаще всего поиск проводится по именам файлов, как это показано в предыдущем примере, т.е. "критерий_поиска" задается как -name имя_файла. Вместо опции -name можно использовать опцию -path, тогда команда будет искать совпадения в полном имени файла с указанием пути. Например, команда find . -path './sr*sc' найдет в текущем каталоге подкаталог ./src/misc. Вместо полного имени файла или каталога в этом примере использован так называемый «шаблон имени». Поскольку шаблоны имен файлов могут использоваться не только с командой find, но и со многими другими командами (включая chmod, chown, chgrp, cp, rm, cat, mv), правилам составления шаблонов стоит уделить некоторое внимание.

Чаще всего шаблоны имен файлов строятся с помощью специальных символов * и ?. Символ * используется для замены произвольной строки символов в Linux:

* – соответствует всем файлам, за исключением скрытых;

. * – соответствует всем скрытым файлам (но также текущему каталогу **.** и каталогу уровнем выше **..**: не забывайте об этом!);

*. * – соответствует только тем файлам и каталогам, которые имеют точку в середине имени или оканчиваются на точку;

p*r – соответствует и peter, и piper;

c – соответствует И picked, И peck.

Значок ? заменяет один произвольный символ, поэтому index7.htm будет соответствовать именам index0.htm, index5.htm И indexa.htm.

Кроме * и ? в Linux при задании шаблонов имен можно использовать квадратные скобки [ ], в которых дается либо список возможных символов, либо интервал, в который должны попадать возможные символы. Например, [abc]* соответствует всем именам файлов, начинающимся с символов a, b, c;

* [i-Ni-З] соответствует файлам, имена которых оканчиваются на i, j, к, L, M, N, 1, 2, 3.

А теперь вернемся к команде find и расскажем подробнее о том, какие критерии поиска возможны. Несколько примеров простых критериев поиска перечислены в табл. 3.

Таблица 3

Опция	Значение
-name шаблон	Ищет файлы, имена которых соответствуют шаблону
-group имя	Ищет файлы, принадлежащие указанной группе
-size число [c]	Ищет файлы, размером в число 512-байтных блоков. Если после числа стоит символ c, значит размер указан в байтах (символах)
-mtime число	Ищет файлы, которые в последний раз изменялись указанное число дней назад
-newer образец	Ищет файлы, которые изменялись после изменения файла, указанного в образце
-type тип_файла	Ищет файлы указанного типа. Тип задается одним из символов b (блок-ориентированные устройства), c (байт-ориентированные устройства), d (файл каталога), f (обычный файл), p (именованный канал) либо 1 (символическая ссылка)

Команды архивирования файлов

При работе в ОС Linux вам придется обязательно работать с командами архивирования и разархивирования, хотя бы потому, что вы будете часто встречать архивированные файлы в Интернете.

Основным средством архивирования в UNIX и в Linux является комплекс из двух программ – tar и gzip. Хотя никто не запрещает пользоваться arj, pkzip, lha, rar и т.д. – версии этих программ для Linux общедоступны. Просто уж исторически сложилось, что пользователи UNIX чаще применяют именно tar и gzip, и именно в таком формате распространяется большая часть программного обеспечения для UNIX. Поэтому овладеть работой с tar и gzip – дело чести любого пользователя Linux.