Шамшин Ю.В.

Лабораторный практикум N 8

**Изучение файловой системы** ОC **UNIX**

по курсу

«Операционные системы»

Цель работы:

1. познакомиться с содержимым дерева каталогов ФС UNIX и разобраться с назначением основных подкаталогов;
2. приобрести навыки по чтению, установке и использованию прав доступа на объекты файловой системы UNIX;
3. освение команд chmod, chown, chgrp, umask, chattr, lsattr.

8.1. Описание формата вывода команды ls

Команда ***ls –l*** выводит список файлов с их атрибутами в следующем формате:

1 9 3 8 8 12 3 2 5 25 и более

TPPPPPPPPP llL Uuuuuuuu Gggggggg sssssssssssS MMM dD yYYYY Nnnnnnnnnnnnnnnnnnnnnnnn

-rwxr-xr-- 1 vasja students 123456789 Feb 3 2000 hello

lrwxrwxrwx 1 vasja students 4 Nov 23 22:05 privet -> hello

Все поля кроме первых двух отделены пробелами. Все поля кроме последнего имеют фиксированную ширину. Над каждым полем показана его ширина. Строчные буквы в формате поля обозначают необязательное заполнение этой позиции, если содержимое поля меньше отведенного, то недостающие позиции заполняются пробелами.

Далее приведено краткое описание полей.

**T (Type) - тип файла**

- обычный файл.

d directory – каталог (фактически, каталог - это специальный файл, в котором перечислены файлы этого каталога).

l link – символическая связь (файл, содержащий имя другого файла - является сылкой на этот файл; подробности ниже в команде *ln*).

c character – символьное устройство (устройство, с которым операции чтения-записи производятся побайтно).

b block – блочное устройство (операции чтения-записи буферизуются). Некоторые устройства являются только символьными (например, последовательный порт), а некоторые представлены в обоих вариантах (например, диск).

p pipe – конвейер (программный канал, организованный в виде очереди, для передачи данных между двумя процессами).

s socket – сокет (логический канал, образованный между двумя сетевыми приложениями, используются для взаимодействия между процессами).

**P (Permission) – права на использование файла**

Следующие после типа файла 9 символов представляют собой права на использование файла: первая тройка символов - для пользователя-владельца файла, вторая - для группы-владельца, третья - для всех остальных. Наличие буквы в каждой тройке означает наличие соответствующего права.

Для обычных файлов:

r - право на чтение содержимого файла

w - право на изменение содержимого файла

x - право на исполнение (запуск) файла

Для каталогов:

r - право на получение списка файлов каталога

w - право на создание и удаление файлов в каталоге

x - право на "поиск" в каталоге

Для символических ссылок права определяются целевым объектом (директорией, файлом, устройством), а в поле всегда содержится rwxrwxrwx.

Набор прав --x для каталога позволяет войти в него и прочитать файл, имя которого заранее известно (при условии наличия прав чтения этого файла). Можно также выполнить команды ls и ls -l для этого файла. При этом получить список файлов (ls (-l) для каталога), а также создать или удалить файл нельзя.

Право r-- для каталога позволяет произвести чтение каталога, то есть получить список имен его файлов. При этом ни войти в каталог, ни узнать атрибуты файлов (ls -l), ни прочесть файл (даже при наличии прав чтения этого файла) нельзя. Создать или удалить файл, естественно, нельзя.

*Что означают права -wx для каталога? Где это может применяться?*

Модификация файла требует прав на запись в файл, так как это операция над файлом. Удаление файла требует прав на запись в каталог, в котором находится этот файл - это операция над каталогом. Наличие права записи в файл не обязательно означает возможность его удалить (хотя можно размер файла уменьшить до нуля).

**L (Links) – число жестких связей**

Число жестких связей (имён) файла - подробнее см. ниже команду [ln](http://athena.vvsu.ru/unix/labs/lab1.html%22%20%5Cl%20%22ln).

**U (User****owner) – пользователь-владелец файла**

Файлы в Linux имеют двух владельцев: *пользователя (user owner)* и *группу (group owner)*, причем владелец файла не обязательно должен быть членом группы, владеющей файлом.

Каждый пользователь может быть членом сразу нескольких групп, одна из которых называется *первичной (primary)*, а все остальные — *дополнительными (supplementary)*. Это дает большую гибкость в организации доступа к определенному файлу.

**G (Group owner) – группа-владелец файла**

Группа - это именованное множество пользователей; группы составляются администратором. Каждый пользователь является членом как минимум одной группы.

Совместное пользование некоторым ресурсом организовать очень просто — достаточно создать новую группу и включить в нее всех, кому это действительно необходимо. Если же человек, предположим, перешел в другой отдел, и уже нет необходимости в использовании данного файла, необходимо просто исключить его из состава данной группы.

Пользователь-владелец файла (далее - просто владелец) не обязательно является членом группы-владельца, хотя и может быть таковым.

**S (Size) – размер файла в байтах**

Для директории показывается суммарный размер в байтах занятых директорией кластеров, показывается размер не менее одного кластера, т.к. даже в пустой директории существует две записи: ссылка на себя и на родительскую. Для HDD размер директории кратен 512\*n байтов, где n – размер кластера в блоках. Для символической связи показывается длина имени ссылки на целевой файл.

**M, D, Y (Month, Day, Year) – дата последней модификации файла**

Выводится месяц и день модификации файла в формате MMM dD, а также время или год. Если файл был изменен  менее 6 месяцев назад, то выводится время модификации файла в формате HH:MM, иначе выводится год в формате YYYY.

Для специальных файлов-устройств (символьные и блочные устройства) через запятую показываются старший и младший номера устройств.

**N (Name) – имя файла**

Типичная поддерживаемая длина имени файла составляет 1024 байта, могут использоваться любые символы ASCII-8 (даже непечатаемые). Для символических ссылок отображаются имя ссылки и имя цели.

8.2. Изучение содержимого файлов

Самостоятельно освойте приведенные ниже команды.

**Команда** **file** определяет тип файла путем анализа его содержимого.

file имя\_файла [имя\_файла ...]

Определите тип всех файлов вашего домашнего каталога и каталога **/etc**.

Расширение файла (часть имени после точки) в Unix не имеет никакого специального значения. Например, для запуска файла требуется не расширение типа .exe, а право на исполнение. Анализ типа файла базируется на специальном файле **magic**.

**Команда** **cat** выводит содержимое файла на терминал.

cat имя\_файла

**Команда more** выводит файл постранично, с остановом.

**more имя\_файла**

Фактически, команда *man* формирует текст справки и пропускает его через *more*. Поэтому команды управления прокруткой текста см. в описании *man*.

**Команда tail** выводит «хвост» файла.

**tail [-|+число] имя\_файла**

Без ключа выводит последние 10 строк файла; +n – начиная с n и до конца; -n – указанное количество строк с конца.

**Команда head** выводит первые 10 строк файла или сколько указано.

**head [-|+число] имя\_файла**

**Команда od** выводит файл в восьмеричном виде.

**od имя\_файла**

Ключи:

**-Nчисло** - вывести только указанное число первых байт,

**-jчисло** - пропустить указанное число первых байт,

**-j0xчисло** - пропустить указанное число первых байт (число является 16ричным),

**-tx** - вывести в шестнадцатеричном виде,

**-с** - вывести каждый байт в виде символа (непечатные символы отображаются как спецсимволы языка Си: \t,\n,\b,\f,\r,\0, либо как трехзначное 8-ричное число).

$ od -N10 tx b.txt

**Команда du** размер файлов в указанном каталоге.

**du -k каталог**

Выводит информацию о суммарном размере файлов в указанном каталоге и - рекурсивно - его подкаталогах (плюс размер собственно каталога); ключ -k означает, что данные выводятся в килобайтах, иначе - в 512-байтных блоках. Если каталог не указан, подразумевается текущий.

Ключ -s подавляет рекурсивный вывод информации о подкаталогах - команда сообщает только суммарный объем.

$ du –k; du -ks

Как получить сводку размеров домашних каталогов пользователей (без детализации по подкаталогам)?

$ du -ks /home; du -k /home; du -ks /home/\*

8.3. Символические и жесткие связи

**Символические связи**

Символическая связь - особый тип файла, содержащий имя другого файла (в листинге ls -l такие файлы обозначаются буквой l в первой колонке). Чтение-запись в файл-связь на самом деле приводят к чтению-записи в файл, на который он ссылается.

Создание символической связи:

**ln -s сущетсвующий\_файл файл\_связь**

$ ln -s b.txt c.txt

$ ls -l

(внимание на файл c.txt и его размер: 5 - это число символов в имени "b.txt")

$ more c.txt

**Жесткие связи**

В файловой системе UNIX имя файла является указателем на индексный дексриптор, который содержит атрибуты файла и массив адресов дисковых блоков, в которых находятся данные файла. Однако, индексный дексриптор не содержит ссылки на имя файла, следовательно файл может иметь несколько имен. Дексриптор содержит только счетчик числа этих имен, значение которого показывается во второй колонке листинга ls -l.

Создание жесткой связи - это создание еще одного имени, ссылающегося на тот же самый индексный дескриптор:

**ln сущетсвующее\_имя\_файла новое\_имя\_файла**

$ ln b.txt d.txt

$ ls -l

(внимание на значение второй колонки для b.txt и d.txt и размер этих файлов, заметьте также время последней модификации)

Файлы b.txt и d.txt абсолютно равноправны, т.к. это два разных имени одного и того же физического файла.

$ touch d.txt

$ ls -l b.txt d.txt

(внимание на время последней модификации у обоих файлов)

$ ls -i b.txt d.txt

(выводит номера индексных дескрипторов - видим, что это один и тот же файл)

8.4. Что представляют собой команды UNIX?

Большинство команд представляют собой программы, находящиеся в каталоге **/usr/bin**. Чтобы выяснить, что понимает шелл под той или иной командой, используйте команду ***type***. Шелл либо укажет путь к исполняемому файлу, либо сообщит, что это - его встроенная команда, либо скажет, что это псевдоним.

$ type ls rm kill type man

Поиск исполняемого файла производится в каталогах, установленных переменной окружения PATH. Чтобы определить в каком каталоге находится программа, используйте команду **which** или **whereis:**

$ which ls; which kill

$ which qwerty #можете увидеть, что содержится у вас в PATH

$ whereis man

Оказывается, существует программа **/bin/kill**, но шелл интерпретирует **kill** как свою встроенную команду. Дело в том, что все вышесказанное относится к случаю, когда вы вводите команду просто по имени. Если же вы вводите путь к исполняемому файлу (полный или относительный), то шелл никак не интерпретирует имя программы, а запускает конкретный указанный вами файл. То есть

$ /bin/kill

запустит именно этот файл, а не встроенную команду шелла. Впрочем, делают они одно и то же.

8.5. Основные каталоги файловой системы Unix

***Цель:*** получить представление об общем устройстве дерева фаловой системы Unix и об основных каталогах и их предназначении. Изучение производится на примере ОС Solaris. В других Unix-системах могут быть некоторые непринципиальные отличия. Изучите список каталогов верхнего уровня.

8.5.2. Изучение каталога /etc/

В каталоге /etc/ содержатся системные конфигурационные файлы. Все эти файлы - текстовые.

Практикум

Сообщение, выводимое при входе в систему:

$ more motd

Имя компьютера:

$ more HOSTNAME

/etc/rc.d/

В каталоге /etc/rc.d содержатся файлы, исполняемые при загрузке операционной системы. По умолчанию загружается только ядро и его системные процессы. Все остальные сервисы, включая поддержку сети, обслуживание сеансов работы пользователей и проч., запускаются из файлов этого каталога. Содержание и назначение этих файлов изучаются в "Администрировании Unix".

8.5.2. Изучение каталога /usr/

В каталоге /usr содержатся команды, утилиты, библиотеки, документы справочника man и т.п. То есть в этом каталоге находится вся "исполняемая" часть Unix, за исключеним ядра. Основные подкаталоги каталога /usr:

/usr/bin/

(также /bin) - команды Unix для обычного пользователя.

/usr/sbin/

(также /sbin) - команды Unix для выполнения задач администрирования.

/usr/lib/

(также /lib) - библиотеки: статические и разделяемые.

/usr/include/

- заголовочные файлы функций языка С.

/usr/ man/

(также /usr/local/man или /usr/share/man) - документы справочника man.

/usr/local/

- поддерево установки дополнительных программ.

В /usr/local устанавливается ПО общесистемного назначения: компиляторы, дополнительные утилиты, библиотеки. Большие пакеты прикладного характера ставятся обычно в /opt.

Каталог /usr/local по своей структуре копирует основные подкаталоги /usr: bin, lib, man, include.

Практикум

$ ls /bin

$ ls /sbin

$ ls /usr/bin

$ ls /usr/sbin

Просмотрите список команд в этих каталогах, прочтите справку по команде с заитересовавшим вас именем.

$ ls lib

Обратите внимание на расширения файлов: .a - статические библиотеки (архивы объектных файлов), .so - динамические библиотеки. Файл libc - стандартная библиотека языка Си.

$ file libc.so

$ file libc.a

$ ar -t libc.a     #распечатывает содержание статической библиотеки

$ ls include

$ ls -l man

$ ls -l share/man

Каждый подкаталог каталога man содержит документы соответствующего раздела. Документы являются текстовыми файлами с символами форматирования. Команда **man** перед показом пропускает документ через форматирующую программу, а после этого - через **more**.

$ ls /usr/share/man/man1

$ more /usr/share/man/man1/cp.1

$ man cp

Сравните результаты двух последних команд.

$ ls local

$ ls local/bin

Обратите внимание, какие программы имеются в /usr/local/bin. попробуйте получить справку по этим программам (например, perl). Если man не находит требуемого документа, то, вероятно он не производит поиск в /usr/local/man (это регулируется переменной окружения MANPATH); попробуйте такой вариант:

$ man -M /usr/local/man perl

а теперь установим переменную MANPATH:

$ MANPATH=/usr/man:/usr/local/man

$ export MANPATH

$ man perl

$ man -l perl

8.5.3. Изучение каталогов /dev/ и /devices/

В этих каталогах находятся специальные файлы типов **b** и **c**, представляющие собой устройства компьютера. Обычно в Unix спецфайлы лежат в каталоге **/dev**, однако в Solaris в **/dev** находятся только ссылки на спецфайлы, которые на самом деле размещены в каталоге **/devices**.

Спецфайлы в каталоге **/devices** организованы по принципу размещения устройств на системной шине компьютера.

Практикум

$ cd /devices

$ ls

Устройства делятся на настоящие устройства, расположенные на шине PCI, и псевдо-устройства. Например, оперативная память mm или псевдотерминалы pts, к которым подключается программа telnet.

$ ls pseudo

$ cd pci@1f,0/pci@1,1/ide@3

Путь дан для платформы SUN Ultra-5/10. На других платформах путь выглядит по другому.

$ ls

В листинге представлены IDE-устройства компьютера: диск (диски) и CDROM

$ ls -l

В длинном листинге (ls –l) для каждого файла устройства вместо размера указывается два числа: major number и minor number. Major number идентифицирует для ядра драйвер устройства; аналогичные устройства имеют одинаковые major numbers. Minor number идентифицирует конкретный экземпляр устройства.

Конкретно в каталоге **pci@1f,0/pci@1,1/ide@3** каждый диск идентифицируется своим major number, а каждый размел диска (всего 8 разделов) имеет свой minor number. Каждый раздел диска представлен в виде спецфайлов двух типов: символьного и блочного. Симольный доступ к диску считается низкоуровневым (raw). При работе с файловыми системами используется буферизированный ввод-вывод и, соответственно, блочные устройства.

Со спецфайлами во многих отношениях работать также как с обычными файлами. Например, для посекторного копирования одного раздела диска в другой (естественно, при условии их одинаковых размеров) можно скопировать соответствующий символьный спецфайл.

Имена устройств из каталога **/devices/** не используются в обычной практике по причине своей неудобозапоминаемости и, что главное, - из-за зависимости от аппаратной конфигурации компьютера. Вместо этого, как и во всех Unix-системах, в Solaris употребляются файлы из каталога **/dev**, которые, однако, в этой системе являются символическими связями, указывающими на соответствующий файлы в **/devices**. При этом достигается независимость от аппаратной части: например, **/dev/dsk/c0t0d0s0** - это всегда первый раздел первого жесткого диска.

Конфигурирование каталогов **/dev** и **/devices** (если необходимо) происходит при загрузке операционной системы.

$ cd /dev

$ ls

$ ls -l term/a

$ ls -l term/b

Это последовательные порты - они же терминалы.

$ ls -l pts

Это псевдотерминалы, используемые программой telnet - они являются псевдоустройствами.

$ ls dsk

$ ls -l dsk

Разделы дисков с блочным доступом; обратите внимание куда указывают связи.

$ ls rdsk

$ ls -l rdsk

Разделы дисков с символьным доступом; обратите внимание куда указывают связи.

$ ls -l diskette

Флоппи-диск.

$ ls -l mem

Оперативная память – пседоустройство.

$ ls -l null

Весь вывод, направленный в файл **/dev/null**, уничтожается, а ввод из него имеет нулевую длину; обратите внимание, что это псевдоустройство того же класса, что и **/dev/mem**.

8.5.4. Изучение каталога /var/

В подкаталогах каталога **/var** хранятся временные или служебные файлы различных программ. Основные подкаталоги каталога **/var**:

**/var/log** – журналы регистрации событий в системе.

**/var/adm** – журналы регистрации событий административного контекста.

**/var/mail** – новая (непрочитанная) поста пользователей.

**/var/spool** – различные очереди (на печать, на отправку почты, на выполнение заданий по расписанию).

**/var/tmp** – различные временные файлы.

Практикум

$ cd /var; ls

$ ls -l log

$ tail log/syslog

$ ls -l mail

$ ls -ld tmp

Обратите внимание: запись в этот каталог разрешена всем. Аналогичную функцию выполняет каталог **/tmp**, но природа его файловой системы другая, что, в общем-то, для пользователя не имеет значения.

$ ls -ld /tmp

8.5.5. Изучение каталогов home, opt, sbin, kernel

В каталоге **/home/** размещены домашние каталоги пользователей.

В каталоге **/opt/** располагаются большие пакеты прикладного ПО (аналог каталога Program Files).

В каталоге **/sbin/** располагаются самые необходимые команды и утилиты системного админитратора (в том числе и его шелл **/sbin/sh**), поскольку в случае сбоя системы каталог **/usr/sbin/** может быть недоступен, так как мог располагаться на другой файловой системе.

В каталоге **/kernel/** расположено ядро операционной системы (файл **genunix**), драйверы устройств и другие служебные файлы.

$ ls -l /home

$ ls -l /opt

$ ls -l /sbin

$ ls -l /kernel

8.6. Исследование смонтированных файловых систем

Команда **mount** выводит информацию о смонтированных файловых системах и параметрах монтирования.

Команда **df** выводит данные о размерах файловых систем, занятом и свободном пространстве; ключ -k определяет формат вывода и означает, что данные выводятся в килобайтах.

$ mount

$ df -k

Нарисуйте дерево основных каталогов с указанием того, какие файловые системы (разделы дисков) куда смонтированы, каков их размер и процент занятости.

Вставьте диск в привод CDROM и через несколько секунд повторите команды **mount** и **df -k**. Что изменилось? Обратите внимание на параметры монтирования CD. Выньте диск после подачи команды:

$ eject cdrom

Обычно монтирование файловых систем производится автоматически при загрузке ОС.

Файл **/etc/vfstab** содержит информацию о файловых системах, точках и параметрах монтирования, указания для программы проверки fsck.

$ more /etc/vfstab

Администратор должен редактировать **/etc/vfstab** крайне аккуратно. Некорректные данные в этом файле или изменение его формата (например автоматической сворачивание длинных строк текстовым редактором) приведут к тому, что система не будет загружаться.

Научитесь работать с флоппи-диском.

$ mount /dev/fd0

$ umount /mnt/floppy

8.7. Изменение прав доступа к файлу

Для изменения прав доступа используется команда

**chmod [-R] права файл\_или\_каталог [файл2 ...]**

Необязательный ключ -R распространяет действие команды рекурсивно на содержимое каталогов, если таковые обнаружатся в списке файлов, переданном в командной строке.

Права указываются в одной из двух нотаций: числовой и символьной.

**Числовая нотация команды chmod**

Набор прав разбивается на 4 тройки:

sst rwx rwx rwx

и рассматривается в виде битового поля: бит установлен, если соответствующее право имеется. Каждая тройка бит записывается десятичным числом. Лидирующий ноль можно опускать.

Примеры

|  |  |
| --- | --- |
| --- rwx rwx rwx | 0777 |
| --- rwx r-x r-x | 0755 |
| --- rw- r-- --- | 0640 |
| ss- rwx r-x r-x | 6755 |
| --t rwx r-x r-- | 1754 |

**Символьная нотация команды chmod**

В отличие от числовой нотации символьная нотация указывает не права, а изменения прав. Нотация состоит из 3 элементов, указанных в следующей последовательности: чьи права изменять, каким образом, и какие именно права.

|  |  |  |
| --- | --- | --- |
| **Чьи права изменять** | **Каким образом** | **Какие именно права** |
|  u (владельца)g (группы)o (всех остальных)a (всех трех категорий) | + (добавить)- (убрать)= (установить и сбросить остальные) | rwx |
| = (сделать такими же) | u (как у владельца)g (как у группы)o (как у всех остальных) |
| ug | +- | s (SUID или SGID) |
| u | +- | t (Sticky bit) |

Примеры:

Добавить группе право на запись:

$ chmod g+w file

Убрать у прочих права на запись и исполнение:

$ chmod o-wx file

Добавить владельцу и группе права на чтение и запись:

$ chmod ug+rw file

Установить права прочих такими же, как у группы

$ chmod o=g file

Установить права прочих и группы такими же, как у владельца

$ chmod og=u file

Несколько изменений можно перечислять через запятую:

Добавить вдалеьцу право на исполнение, а у группы и прочих убрать право на запись:

$ chmod u+x,go-w file

Задание.

·         Поэкспериментируйте с командой chmod на файлах своего каталога. Используйте сначала числовую, а потом символьную нотацию. Попробуйте все приведенные выше варианты. Результаты контролируйте командой ls -l.

·         Попробуйте назначать "экзотические" сочетания (наличие бита SUID, SGID или sticky bit без права исполнения).

·         Устанавливается ли sticky bit для файлов?

·         Работают ли SUID, SGID биты установленные для скриптов?

·         Попробуйте различные варианты прав для каталога и изучите, что вы можете сделать при этом с файлами каталога: проверяйте работу команд ls и ls -l, возможность перейти в каталог, создать или удалить файл, просмотреть содержимое файла.

·         Устанавливаются ли SUID, SGID биты для директорий?

·         Попробуйте рекурсивную форму команды chmod.

Не забудьте восстановить разумные права для своих файлов и каталогов в конце работы.