## Министерство общего и профессионального образования РФ

Волжский университет им.Татищева

Факультет «Информатика и телекоммуникации»

Кафедра «Информатика и системы управления»

УТВЕРЖДАЮ

Проректор по УР

**Е.В.** Никифорова

2000 г.

**МЕТОДИЧЕСКОЕ ПОСОБИЕ**

**дисциплина «Операционные системы» ОС UNIX**

**Работа с файлами и каталогами**

для студентов специальностей

Вычислительные машины, системы, комплексы и сети

Информационные системы в технике и технологии

г.Тольятти

2001

Методическое пособие предназначено для студентов дневного и заочного отделений специальностей «Вычислительные машины, системы, комплексы и сети», «Информационные системы в технике и технологии» для изучения основных правил проектирования операционных систем. Разработано на основании государственного стандарта.

Составитель: ст. преподаватель Трубачева С. И.

### Файловая система

#### Структура файловой системы

Файловая система построена из обыкновенных файлов, символических связей и справочников. Эти компоненты обеспечивают способ организации, поиска и управления информацией. Прежде чем подробно описывать компоненты, дадим короткий обзор.

**Обыкновенные файлы** - это набор символов, хранящихся на диске. Они могут хранить тексты или программы или коды программ.

**Специальные файлы** соответствуют физическим устройствам (например, терминал или диск).

**Символическая связь** - это файл, который указывает на другой файл.

**Справочник** содержит имена других справочников (называемых подсправочниками) или файлов. Файлы вы можете группировать в справочники по своему выбору.

Всякий раз, когда вы взаимодействуете с системой UNIX, то вы находитесь в определенном месте структуры файловой системы. Система UNIX автоматически помещает вас в определенную точку в файловой системе каждый раз, когда вы регистрируетесь. Из этой точки вы можете передвинуться по иерархии, чтобы работать в любом вашем справочнике над любым файлом, к которому вы имеете доступ.

##### Ваш собственный справочник

После того как вы успешно завершили процедуру регистрации, система UNIX помещает вас в определенное место в структуре файловой системы, называемое вашим собственным справочником. Регистрационное имя, назначенное вам системой UNIX, обычно является именем этого справочника. Каждый пользователь с полномочным регистрационным именем имеет уникальный собственный справочник в файловой системе.

Система UNIX способна держать связь со всеми собственными справочниками, поддерживая один или более системных справочников для их организации. Например, собственные справочники с регистрационными именами startship, mary2 и jmrs содержатся в системном каталоге, называемом home.

Внутри вашего собственного справочника вы можете создавать файлы и подсправочники. Вы можете перемещать и удалять ваши файлы и справочники и управлять доступом к ним. Вы несете ответственность за все, что создаете в своем собственном справочнике.

**Текущий справочник**

Все время, что вы будете работать в вашем собственном справочнике, предполагается, что он ваш текущий рабочий справочник. Если вы перейдете в другой справочник, то он становится вашим текущим справочником.

Команда pwd печатает имя справочника, в котором вы сейчас работаете. Например, если ваше регистрационное имя startship и вы задаете команду pwd в ответ на первую после регистрации подсказку, то система UNIX ответит следующим образом:

$ pwd<CR> /home/startship

$

В ответ система назовет вам имя справочника, в котором вы работаете (startship) и местоположение этого справочника в файловой системе. Имя пути /home/startship говорит, что справочник root (обозначен косой чертой в строке) содержит справочник home, который в свою очередь содержит справочник startship. Все другие косые черты в имени пути используются для разделения имен справочников и файлов, и показывают положение каждого справочника по

отношению к root.. Имя справочника, которое показывает местоположение справочника в этом пути, называется **полным именем справочника** или **именем пути**.

**Имя пути**

Каждый файл и справочник системы UNIX идентифицируется уникальным именем пути. Имя пути показывает местоположение файла или справочника и обеспечивает направление поиска его. Существует 2 типа имени пути: полное и родственное.

###### Полное имя пути

Полное имя пути (иногда называемое абсолютным именем пути) дает направление, которое начинается в справочнике root и показывает путь далее по уникальной последовательности справочников к конкретному справочнику или файлу. Вы можете использовать полное имя пути для поиска любого файла или справочника в системе UNIX.

Так как полное имя пути всегда начинается со справочника root, то это имя начинается с символа косая черта (/). Последнее имя в полном имени пути может быть либо именем файла либо именем справочника. Все другие имена пути должны быть справочниками.

Чтобы понять как полное имя пути строится, обратитесь к следующему примеру. Предположим, что вы работаете в справочнике startship, расположенном в /home. Вы задаете команду pwd и система печатает полное имя пути вашего рабочего справочника:

/home/startship

/ - это корневой справочник (root);

home - системный справочник, расположенный на один уровень ниже справочника root;

/ - разделяет имена справочников root и startship;

startship - текущий рабочий справочник.

Родственное имя пути

Родственное имя пути дает направления, которые начинаются в вашем текущем рабочем справочнике и ведут вас вверх или вниз через серию справочников к конкретному файлу или справочнику. Двигаясь вниз из текущего справочника, вы можете получить доступ к своему файлу или справочнику. Двигаясь вверх из текущего каталога, вы пройдете через родительские справочники к родителю всех системных справочников, т.е. к root.

Родственное имя пути начинается с имени справочника или файла.

Одна точка (.) означает текущий справочник, две точки (..) - справочник, непосредственно находящийся выше текущего справочника в иерархии файловой системы. Справочник, представленный двумя точками, называется родительским для справочника, обозначенного одной точкой (вашего текущего справочника).

Например, вы находитесь в справочнике startship, который содержит справочники draft, letters и bin и файл mbox. Родственным именем пути для каждого из них является просто его имя.

Справочник draft, принадлежащий справочнику startship, содержит файлы outline и table. Родственное имя пути от startship к файлу outline:

draft/outline

Обратите внимание, что косая черта в этом имени пути разделяет имя справочника draft и имя файла outline. Здесь наклонная черта показывает, что outline является подчиненным справочнику draft, т.е. draft является родителем outline.

Чтобы подняться к родителю вашего текущего справочника, вы можете ввести две точки (..). Это означает, что если вы находитесь в справочнике draft, то .. является именем пути к startship, и ../.. является именем пути к родительскому справочнику для startship, т.е. к home.

Из справочника draft вы также можете проследить путь к файлу sanders, указав имя пути ../letters/sanders. Две точки (..) приводит вас к starship. Затем имена letters и sanders ведут вас вниз через справочник letters к файлу sanders.

Правила именования справочников и файлов

Вы можете давать справочникам или файлам любые имена в соответствии со следующими правилами:

допустимы все символы, за исключением /;

некоторые имена лучше не использовать, такие как пробел, табуляция и следующие: ? " #$^ ();<>[]|\\*@'~&. Если вы воспользуетесь символами пробел или табуляция в имени файла или справочника, то вы должны заключить имя в двойные кавычки в командной строке;

избегайте использования знаков + - или **.** в качестве первого символа в имени файла;

система UNIX различает большие и маленькие буквы. Например, система предполагает, что справочник (или файл) названный draft отличается от DRAFT.

**Организация справочника**

В этом разделе описываются 4 системные команды, позволяющие вам организовывать и использовать структуру справочника:

**mkdir** - создать новый справочник или подсправочник в вашем текущем справочнике;

**ls** - напечатать список имен всех подсправочников и файлов в справочнике;

**cd** - изменить ваше местоположение в файловой системе, т.е. перейти из одного справочника в другой;

**rmdir** -удалить пустой справочник. Эти команды могут использовать либо полное имя пути, либо родственное. Две команды могут применяться без имени пути.

**Команда mkdir - создать справочник**

Рекомендуется создавать подсправочники в вашем собственном справочнике в соответствии с логической схемой, которая будет облегчать восстановление информации ваших файлов. Если вы поместите все файлы, относящиеся к одной теме, в один справочник, то вы будете знать, где их затем найти. Для создания справочника воспользуйтесь командой mkdir.

**Синтаксис команды:**

**mkdir** **имя справочника(ов)**

Например, подсправочник draft создается при помощи следующей команды, выдаваемой из справочника home (/home/startship): $ mkdir draft

Пользователь может создать сразу несколько подсправочников одновременно, перечислив их всех в одной командной строке:

$ mkdir draft letters bin

Вы можете перейти к подсправочнику и в нем построить дополнительные подсправочники.

**Команда ls - напечатать содержимое справочника**

Все справочники в файловой системе имеют информацию о содержащихся в них файлах и справочниках, такую как: имя, размер и дата последней модификации. Вы можете получить эту информацию о вашем текущем справочнике и других системных справочниках, задав команду ls.

Команда ls перечисляет имена всех файлов и подсправочников в указанном справочнике. Если вы не укажете справочник, то команда ls напечатает информацию о файлах и справочниках в вашем текущем справочнике.

Обратимся к примеру. Предположим, что вы зарегистрировались в системе UNIX под именем startship и задали команду pwd. Система напечатает имя пути /home/startship. Чтобы отобразить имена файлов и справочников в этом текущем справочнике, введите команду ls и нажмите клавишу <RETURN>. После того, как вы выполните последовательность этих команд, экран будет выглядеть следующим образом:

$ pwd<CR>

/home/ startship

$ ls <CR>

bin

draft

letters

list

mbox

$

Обратите внимание, что система выдает перечень справочников в алфавитном порядке. Если первым символом имени файла или справочника будет цифра или большая буква, то оно будет напечатано первым.

Чтобы напечатать имена файлов и подсправочников в справочнике, отличном от текущего без перехода из текущего справочника, вы должны указать имя справочника.

**Синтаксис команды:**

**ls имя пути**

Имя пути может быть либо полным именем пути требуемого справочника, либо родственным. Например, вы можете получить содержание справочника draft, когда вы работаете в справочнике startship, если введете команду ls draft. Экран терминала будет выглядеть следующим образом:

$ls draft<CR>

Outline

table

$

В этом примере draft является родственным именем пути от родителя (starship) к подчиненному справочнику (draft).

Вы также можете использовать родственное имя пути для печати содержимого родительского справочника в то время, когда вы находитесь в подчиненном справочнике. Две точки (..) являются самым простым способом выполнения этой функции. Например, следующая командная строка указывает родственное имя пути от справочника startship к home:

$ Is ..<CR> jmrs

mary2 startship

$

Вы получите тот же результат, если зададите полное имя пути от root к home:

ls /home<CR>

Вы можете распечатать содержимое любого системного справочника, к которому имеете право доступа, выполнив команду ls с полным или родственным именем пути.

Команда ls особенно полезна, если вы имеете много файлов и пытаетесь определить, существует ли определенный файл в вашем текущем справочнике. Например, вы находитесь в справочнике draft и хотите определить, есть ли в нем файлы outline и notes. Задайте команду Is следующим образом:

$ ls outline notes<CR>

outline

notes: No such file or directory

$

Система сообщит, что файл outline существует в этом справочнике, а вот файл notes не найден.

Команда Is не распечатывает содержимое файла. Для этих целей воспользуйтесь командами cat, pg, pr.

**Часто используемые ключи команды ls**

Команда ls может содержать ключи, которые перечисляют специфичные атрибуты файла или подсправочника. Чаще всего используются ключи -а и -l.

1 .Перечислить все файлы в справочнике.

Имена некоторых файлов в вашем собственном справочнике начинаются с точки (например, .profile). Когда имя файла начинается с точки, он не включается в список, распечатываемый командой Is. Чтобы распечатать такие файлы, введите команду Is с ключом -а..

Например, чтобы распечатать информацию о всех файлах в вашем текущем справочнике (starship), включая и файлы, имя которых начинается с точки, введите команду Is -а и нажмите клавишу <RETURN>.

2.Распечатать содержание в укороченном формате.

Ключи -С и -F команды ls используются достаточно часто. Вместе эти ключи распечатывают подсправочники и файлы справочников и помечают исполняемые файлы символом \*, справочники - символом /, символическую связь (каналы) - символом @.

Например, вы можете распечатать информацию о всех файлах вашего рабочего справочника startship, выполнив команду:

3.Распечатать содержание в расширенном формате.

Возможно, более информативным ключом команды ls является ключ -1, который отображает содержание справочника в расширенном формате. Этот формат включает в себя: режим, число связей, владелец, группа, размер в байтах и время последней модификации каждого файла. Предположим, вы задали команду Is -1, находясь в справочнике startship:

Первая выводная строка (total 30) показывает объем дискового пространства в байтах. Последующие строки дают представление о справочниках и файлах в справочнике startship. Первый символ в каждой строке (d, -, l, b или с) говорит о типе файла:

d = справочник

- = обыкновенный файл

1 = символическая связь (канал)

b = специальный блочный файл

с = специальный символьный файл

Таким образом на экране вы можете видеть, что справочник startship содержит три справочника и два обычных файла.

Следующие несколько символов, которые являются либо буквами, либо дефисами, идентифицируют право на чтение и использование файла или справочника.

Далее следует цифра - счетчик связей. Для справочника этот счетчик показывает число справочников, расположенных под ним, плюс два (для самого справочника и справочника-родителя).

Следующим является регистрационное имя владельца файла (в данном случае startship), и за ним - групповое имя файла или справочника (project).

Следующее число показывает длину файла или справочника в байтах. Месяц, день и время последней модификации файла - в предпоследней колонке. В последней колонке представлено имя справочника или файла.

**Команда cd - изменить текущий справочник**

Сразу же после регистрации в системе UNIX, вы попадаете в ваш собственный справочник. Пока вы будете работать в нем, он будет вашим текущим справочником. С помощью команды cd вы сможете работать в другом справочнике.

**Синтаксис команды:**

**cd имя-пути-нового-справочника<СR>**

Любое допустимое имя пути (полное или родственное) может использоваться в качестве аргумента команды cd.. Если вы не укажете имя пути, то команда переместит вас в ваш собственный справочник. Справочник, в который вы перешли, становится вашим текущим справочником.

Например, чтобы перейти из справочника startship в подчиненный draft, введите команду cd draft и нажмите клавишу RETURN. После получения подсказки вы можете проверить ваше новое местоположение, введя команду pwd.

Теперь, находясь в справочнике draft, вы можете создать подсправочники в нем, задав команду mkdir, и новые файлы, используя редактор ed или vi.

Нет необходимости находиться в справочнике draft, чтобы получить доступ к файлам, расположенных в нем. Вы можете получить доступ к ним из любого справочника, указывая полное или родственное имя пути для этого.

Вы также можете использовать полное имя пути в команде cd. Например, чтобы перейти из справочника draft в справочник letters, введите:

cd /home/startship/letters<CR>

Так как letters и draft являются подчиненными справочнику srartship, то вы можете использовать родственное имя пути ../letters в команде cd. Две точки (..) перемещают вас в справочник startship и /letters перемещает вас в letters.

Итак, команда cd изменяет ваш рабочий справочник. Аргументом команды cd является новое имя справочника. Если аргумент отсутствует, команда cd помещает вас в ваш собственный справочник.

Когда shell помещает вас в указанный справочник, то возвращается подсказка $. Чтобы получить доступ к справочнику, который не является вашим рабочим справочником, вы должны указать полное или родственное имя пути в командной строке.

**Команда rmdir - удалить справочник**

Если вам больше не нужен справочник, вы можете удалить его с помощью команды rmdir.

**Синтаксис команды:**

**rmdir имя (имена) справочника(ов)**

Вы можете указать более одного имени справочников в командной строке.

Командой rmdir вы не можете удалить справочник, если вы не являетесь его владельцем или он не пустой. Если вы хотите удалить файл из справочника другого пользователя, то владелец должен дать вам право на запись для родительского справочника этого файла.

Если вы попытаетесь удалить справочник, в котором содержатся подсправочники и файлы, то команда rmdir напечатает сообщение:

имя-справочника not empty

Например, предположим, что вы имеете справочник memos, который содержит один подсправочник: tech и два файла june.30 и july.31. Если вы попытаетесь удалить справочник memos, то получите сообщение:

$ rmdir memos<CR> rmdir: memos not empty

$

Чтобы удалить справочник memos, вы должны сначала удалить его содержимое: подсправочник tech и файлы june.30 и july.31. Удалив содержимое справочника memos, можете удалить и его. Однако сначала переместитесь в родительский справочник (ваш собственный). Команда rmdir не отработает, если вы будете находиться в том же справочнике, который удаляете. Из своего собственного справочника задайте команду:

rmdir memos<CR>

Если справочник memos пуст, то эта команда его удалит и возвратит вам подсказку.

**Доступ к файлу и работа с ним**

В этом подразделе описываются несколько команд системы UNIX, которые осуществляют доступ к файлу и производят обработку его в структуре файловой системы. Команды разделены на две группы: основные и advanced.

Основные команды являются фундаментальными в использовании файловой системы; advanced команды предполагают сложную технику обработки информации при работе с файлами.

##### Основные команды

В этом подразделе описываются команды системы UNIX, необходимые для доступа к файлам и для их использования.

cat - Печатает содержимое указанного файла на терминал

pg - Печатает содержимое указанного файла на терминал порциями или страницами

рг - Печатает частично отформатированную версию указанного файла на терминал

ср - Создает копию существующего файла

mv - Перемещает или удаляет файлы

rm - Удаляет файл

wc - Распечатывает информацию о числе строк, слов и символов в файле

chmod - Изменяет режим доступа к файлу (или справочнику)

**Команды cat, pg, рг - распечатать содержимое файла**

В системе UNIX существует три команды для распечатки содержимого файлов: cat, pg, рг. Команда cat выводит содержимое файла на экран терминала или, если вы укажете, в другой файл или новую команду.

Команда pg особенно полезна, если вы хотите прочитать содержимое большого файла, т.к. она отображает текст файла постранично.

Команда рг форматирует указанные файлы и отображает на терминал или направляет вывод на печать.

**Команда cat**

Команда cat отображает содержимое файла или файлов. Например, предположим, вы находитесь в справочнике letters и вы хотите отобразить содержимое файла Johnson. Введите команду cat Johnson и на экране появится следующая информация:

$ cat johnson<CR> March 5, 1986

Mr. Ron Johnson Layton Printing 52 Hudson Street New York, N.Y.

Dear Mr. Johnson:

I enjoyed with you this morning

about your company's plans to

your business.

Enclosed please find

the material you requested

about AB&C's line of computers

and office automation software.

If I can be of furtther assistance to you, please don't besitate to call.

Yours truly,

John Home

$

Чтобы отобразить содержимое двух или более файлов, перечислите имена этих файлов в командной строке. Например, чтобы распечатать содержимое файлов Johnson и sanders, введите команду:

$ cat Johnson sanders<CR>

Команда cat прочитает файлы Johnson и sanders и распечатает их содержимое на терминале.

**Синтаксис команды:**

**cat ключи имя-файла(ов)<СR>**

Если указанный файл не существует или его нельзя прочитать, на экране появляется следующее сообщение:

cannot open имя-файла

**Команда pg**

Команда pg позволяет вам распечатывать содержимое файла(ов) на терминал. После того, как pg отобразит страницу текста, она напечатает подсказку "двоеточие" (:), которая служит сигналом ввода вашей инструкции. Возможной инструкцией может быть запрос вывода следующей страницы содержимого файла, либо запрос на поиск указанного символа по образцу.

Команда Функция

h - Помощь. - Распечатывает список допустимых инструкций команды pg

q или Q - Прекращает режим команды pg

<CR> - Отображает следующую страницу текста

l - Отображает следующую строку текста

d или ^d - Отображает дополнительную половину страницы текста

или ^l - Вновь отображает текущую страницу текста

f - Пропускает одну страницу текста и распечатывает следующую за ней

n - Начинает распечатывать следующий указанный в командной строке файл

р - Отображает предыдущий указанный в командной строке файл

s - Отображает последнюю страницу текста текущего файла

/pattern - Осуществляет поиск вперед указанного шаблона символов

?pattern - Осуществляет поиск назад указанного шаблона символов

Примечание. Некоторые команды могут быть введены с цифрой впереди. Например:

+1<CR> - отображает следующую страницу;

1<CR> - отображает предыдущую страницу;

1<CR> - отображает первую страницу текста.

Команда pg особенно полезна, если вы читаете большой файл или серию файлов, т.к. пауза после каждой страницы дает возможность для анализа информации. Размер отображаемой страницы зависит от типа терминала. Например, терминал способен отображать 24 строки; значит страница текста определяется 23 строки плюс одна строка для двоеточия.

Если файл имеет менее 23 строки, то длина страницы будет равна числу строк в файле плюс одна строка (для двоеточия).

**Синтаксис команды:**

pg имя-файла(ов)<СК>

Например, чтобы отобразить содержимое файла outline из справочника draft, введите команду:

pg outline<CR>

Первая страница текста появится на экране. Так как в файле больше строк, чем поместилось на одной странице, то внизу экрана появится двоеточие. Когда вы будете готовы читать текст дальше, нажмите клавишу <RETURN> и pg напечатает следующую страницу текста.

Когда весь файл будет прочитан, в последней строке появится:

(EOF):

Подсказка: (двоеточие) предлагает вам ввести новую команду.

Если вы в ответ на подсказку: нажмете клавишу <RETURN>, то вновь появится подсказка. Вы можете воспользоваться одной из ранее перечисленных допустимых команд.

Успешное выполнение команды pg зависит от типа терминала, который вы используете. Это связано с тем, что программа pg достаточно гибкая и может быть запущена на различных терминалах; то, как она запускается, зависит от конкретного типа терминала. Указав тип терминала, вы говорите команде:

сколько строк печатать;

сколько колонок печатать;

как очистить экран;

как высветить подсказку или другие слова;

как стереть текущую строку.

Чтобы определить тип терминала, присвойте код вашего терминала переменной TERM в файле .profile.

В команде pg могут применяться все допустимые ключи.

**Команда рг**

Команда рг используется для форматирования и печати содержимого файла. Она форматирует заголовки, количество страниц и печатает файл на экране терминала.

Вы можете с помощью ключа задать распечатку файла на построчно-печатающее устройство или направить вывод в другой файл.

Если вы не выберете ни один из допустимых ключей, то команда рг сформирует вывод в одну колонку, страница будет содержать 66 строк и тексту будет предшествовать короткий заголовок. Заголовок состоит из 5 строк: две пустые строки, строка, содержащая дату, время, имя файла и номер страницы и далее две пустые строки.

Команда рг часто используется с командой lр для получения копии текста на бумаге в том виде, в каком он был введен в файл.

Например, чтобы проанализировать содержимое файла johnson, введите команду:

$pr johnson<CR>

Пустые строки после последней строки символов в файле рг добавляет к выводу, так что каждая страница содержит в общем 66 строк. Если вы работаете с видеотерминалом, который имеет 24 строки, то все 66 строк отформатированного текста без пауз распечатаются на экране. Это означает, что первые 42 строки невозможно будет прочитать, т.к. вы не можете вернуть экран или два назад.

В этом случае воспользуйтесь:

^s - для прерывания потока выводимого на терминал

и

^q - для продолжения вывода на экран.

**Команда ср - сделать копию файла**

При работе с системой UNIX у вас может появиться необходимость сделать копию файла. Команда ср полностью копирует содержимое одного файла в другой. Также она позволяет вам скопировать один или более файлов из одного справочника в другой, оставив оригинал файла на прежнем месте.

Чтобы скопировать файл outline в файл new.outline в справочнике draft, введите команду: ср outline new.outline. Когда система скопирует файл, то вернет подсказку. Чтобы проверить существование нового файла, введите команду:

ls<CR>

Эта команда перечислит имена всех файлов и справочников в текущем справочнике (в данном случае в справочнике draft). Экран будет выглядеть следующим образом:

$ ср outline new.outline<CR>

$ls<CR>

new.outline

outline

table

$

Система UNIX не позволяет иметь в одном справочнике два файла с одним именем. Если бы файл new.outline уже существовал в этом справочнике, то он был бы заменен на копию файла outline, предыдущая версия файла new.outline будет удалена.

Если вы попытаетесь скопировать файл outline в другой файл с тем же именем в том же справочнике, то система сообщит вам, что имена файлов идентичны и возвратит вам подсказку.

Если вы затем проверите содержание справочника, чтобы определить сколько копий файла outline существует, то вы получите следующий экран:

$ ср outline new.outline<CR>

cp: outline and outline are identical

$ls<CR>

outline

table

$

Система UNIX разрешает иметь файлы с одинаковыми именами в разных справочниках. Например, вы можете скопировать файл outline из справочника draft в файл outline в справочнике letters. Если вы находитесь в справочнике draft, то сможете воспользоваться одной из перечисленных ниже 4-х командных строк. В первых двух строках вы указываете имя нового файла при выполнении копирования.

ср outline /home/startship/letters/outline<CR>

(указано полное имя пути)

ср outline . ./letters/outline<CR>

(указано родственное имя пути)

Однако система UNIX не требует, чтобы указывали новое имя файла. Если вы не укажите имя нового файла в командной строке, то ср даст новому файлу то же имя, которое имеет оригинальный файл. В этом случае можете задать команду в одном из следующих видов:

ср outline /home/startship/letters<CR>

(указано полное имя пути)

ср outline ../letters<CR>

(указано родственное имя пути)

Если вы хотите дать другое имя новому файлу, то обязательно укажите это имя.

Например, чтобы скопировать файл outline (справочник draft) в файл с именем outlin.vers2 в справочник letters, нужно задать одну из следующих команд:

ср outline home/startship/letters/outlin.vers2<CR>

(указано полное имя пути)

ср outline ../letters/outlin.vers2<CR>

(указано родственное имя пути)

**Синтаксис команды:**

ср ключи имя-файла1 имя-фaйлa2<CR>

ср ключи имя-файла(ов) cnpaвочник<CR>

**Команда mv - переименование файла**

Команда mv позволяет вам переименовать файл в том же справочнике или переместить его из одного справочника в другой. Когда перемещаете файл в другой справочник, то можете переименовать его или оставить прежнее имя.

**Синтаксис команды:**

**mv имя-файла1 имя-файла2<СК>**

Команда mv изменяет имя-файла 1 на имя-файла2 и удаляет файл 1. Имя-файла 1 и имя-файла 2 могут быть любыми допустимыми именами, включая имя пути. Имя-файла2 может быть справочником.

Например, если вы находитесь в справочнике draft и хотите переименовать файл table на new.table, введите команду:

mv table new.table<CR>

Если команда выполнится успешно, то на экране появится подсказка. Проверьте, что файл new.table существует. Экран будет выглядеть следующим образом:

$ mv table new.table<CR>

$ls<CR>

outline

new.table

$

Чтобы переписать файл в другой справочник без изменения имени, введите команду:

mv имя-файла(ов) cnpaвочник<CR>

Например, вы хотите переписать файл table из текущего справочника draft (который имеет полное имя пути /home/startship/draft) в файл с тем же именем в справочник letters (который имеет родственное имя пути из draft ../letters и полное имя пути /home/startship/letters). Для этого можете воспользоваться одной из следующих командных строк:

mv table /home/startship/letters<CR>

mv table /home/startship/letters/table<CR>

mv table ../letters<CR>

mv table .. /letters/table<CR>

mv /home/startship/draft/ table /home/startship/letters<CR>

Теперь, предположим, вы хотите переименовать файл table в table2 и переписать в справочник letters. Для этого воспользуйтесь одной из следующих командных строк:

mv table /home/startship/letters/table2<CR>

mv table ../letters/table2<CR>

Когда вы переписываете файл с именем имя-файла1 в имя-файла2 и файл с именем имя-файла2 уже существует, команда mv перезапишет новую версию файла. Старая версия файла удаляется.

**Команда rm - удалить файл**

Чтобы удалить файл, введите следующую команду:

rm фaйл(ы)<CR>

Вы можете удалить несколько файлов, указав их имена в командной строке. В этом случае команда будет выглядеть следующим образом:

rm файл1 файл2 файл3 ... **<CR>**

Чтобы проверить, успешно ли выполнилась команда, выполните команду ls. Например, предположим, что в вашем справочнике находятся файлы outline и table. Вы хотите удалить оба файла. Если команда rm выполнится успешно, то справочник будет пустой. Проверьте, задав команду ls. Экран будет выглядеть следующим образом:

$ rm outline table<CR>

$ls

$

Подсказка говорит о том, что файлы outline и table удалены.

**Команда wc - подсчет строк, слов и символов**

С помощью команды wc вы можете подсчитать число строк, слов и символов в указанном файле. Если указано более одного файла в командной строке, то программа we осуществляет подсчет строк, слов и символов в каждом файле и затем выдает общее число. Вы можете с помощью ключей указать либо подсчет только строк, или только слов, или символов.

**Синтаксис команды:**

wc имя-фaйлa<CR>

Система отвечает строкой в следующем формате:

1 w с файл

где 1 - число строк в файле;

w - число слов в файле;

с - число символов в файле.

Например, чтобы подсчитать число строк, слов и символов в файле johnson, находящегося в текущем справочнике, введите команду:

$ we johnson<CR> 24 66 406 johnson

$

Система отвечает, что в файле Johnson 24 строки, 66 слов и 406 символов.

Чтобы подсчитать число строк, слов и символов в нескольких файлах, используйте следующий формат:

we файл1 файл2<СR>

Система отвечает следующим образом:

1 w с файл1

1 w с файл2

1 w с total

Число строк, слов и символов для файл1 и файл2 отображается на отдельных строках. На последней строке отображается общее число строк, слов и символов в двух файлах.

Например, подсчитаем число строк, слов и символов в файлах johnson и sanders в текущем справочнике. Экран будет выглядеть следующим образом:

$ wc johnson sanders<CR> 24 66 406 Johnson

1. 92 559 sanders
2. 52 158 965 total

$

Последняя строка показывает, что файлы Johnson и sanders вместе имеют 52 строки, 158 слов и 965 символов.

Чтобы получить только число строк, или число слов, или число символов, выберите один из соответствующих форматов командной строки:

wc -1 файл<СR> (число строк)

wc -w файл<СR> (число слов)

wc -с файл<СК> (число символов)

Например, если вы используете ключ -1, то система напечатает только число строк в файле sanders:

$ wc –l sanders<CR>

28 sanders

$

**Команда chmod - изменить статус файла**

Команда chmod позволяет вам устанавливать разрешение на чтение, запись и использование вашего файла. Так как операционная система UNIX является многопользовательской, то обычно вы не работаете в одиночку в файловой системе. Пользователи системы могут любым путем обращаться к различным справочникам и читать файлы, принадлежащие другим пользователям до тех пор, пока есть разрешение на это.

Если вы являетесь владельцем файла, то можете решить, кто имеет право на чтение файла, запись в него и, если это программа, то выполнить ее. Вы также можете ограничить права доступа к справочнику. Когда вы предоставляете право доступа к справочнику, это значит, что позволяете указывать пользователю команду cd и распечатывать содержимое справочника с помощью команды ls. Чтобы указать право доступа, используются следующие символы:

r - разрешает пользователям системы читать файл или копировать его содержимое;

w -разрешает пользователям системы вносить изменения в файл или копировать его содержимое;

х -позволяет пользователям системы запускать файл на выполнение.

Чтобы указать, кто из пользователей имеет эти права на доступ, используются следующие символы:

u - владелец файла и справочника; g - член вашей группы; о -все пользователи системы.

Когда вы создаете файл или справочник, система автоматически предоставляет или не предоставляет право доступа вам, члену вашей группы или всем пользователям системы. Вы можете изменить это автоматическое действие по предоставлению права. Кроме того, вне зависимости от того, какие права вам предоставляются при создании файла, вы как владелец файла или справочника можете воспользоваться ключом внесения изменений.

**Как определить права**

Вы можете определить права доступа к файлу или справочнику, воспользовавшись командой ls -1. Например, если вы находитесь в справочнике startship/bin, и введете команду ls -1, то получите следующую информацию:

$ ls -1<CR>

total 35

-rwxr-xr-x 1 startship project 9346 Nov 1 08:06 display

-rw-r-- r-- 1 startship project 6428 Dec 2 10:26 list drwxr-x--x 2 startship project 32 Nov 8 15:32 tools $

В левой части экрана отображены права доступа для файлов display и list и справочника tools:

-rwxr-xr-x для файла display

-rw-r- -r -- для файла list

drwxr-x--x для справочника tools

Первый символ описывает тип файла (например, символ "-" говорит, что это обыкновенный файл, символ "d" - справочник). Дальнейшие девять символов описывают права доступа. Первые три символа из девяти устанавливают права доступа для владельца, вторые три - для группы, третьи - для всех пользователей. Каждый набор символов r,w,x показывает текущие установленные права доступа для каждой категории пользователей. Если вместо символа стоит "-", то соответствующее право отсутствует.

Существует два ограничения. Случайно могут появиться буквы 1 или s в строке вместо г, w или х. Буква s представляет специальное право на выполнение файла. Она появляется на том месте, где обычно стоит буква r в наборах для владельца или для группы пользователей и это означает специальное право на выполнение файла. Она имеет значение только для программистов и системных программистов. Буква l указывает, что блокировка будет появляться при обращении к файлу. Это не означает, что файл защищен.

**Как изменить существующие права**

Изменить существующие права можно с помощью команды chmod. **Синтаксис команды:**

chmod кто+права файл(ы)<СR>

или

chmod кто-права файл(ы)<СR>

где

chmod -имя программы; кто -одна их трех групп пользователей: u - владелец; g -группа; о - все пользователи; "+" или "-" -наличие или отсутствие права; право -одно из трех прав: г -право на чтение; w -право на запись; х - право на выполнение; файл(ы) - имя файла(ов) (справочника); предполагается, что файл находится в текущем справочнике или вы должны указать полное имя пути.

Примечание. Команда chmod не будет выполняться, если вы поставите пробел между "кто", символами "+","-" и "право".

В следующих примерах приведены несколько способов использования команды chmod. Как владелец display вы можете читать файл, писать в него и запускать на выполнение файл. Вы можете защитить файл от случайного изменения. Чтобы сделать это, введите следующую командную строку:

chmod u-w display<CR>

После того как получите подсказку, введите команду:

ls -l<CR>

чтобы проверить, как изменились права. Экран будет выглядеть следующим образом:

$ chmod u-w display<CR> $ ls -l<CR>

total 35

-r-xr-xr-x 1 startship project 9346 Nov 1 08:06 display

-rw-r --r-- 1 startship project 6428 Dec 2 10:26 list

drwxr-x--x 2 startship project 32 Nov 8 15:32tools

$

Как видите, изменилось право на внесение изменений в файл. Вы не сможете изменить этот файл, пока существует это право записи.

Теперь обратимся к другому примеру. Право на запись в файл display не разрешено вашей группе и всем пользователям системы. Однако им предоставлено право на чтение. Это означает, что они могут копировать файл в свой собственный справочник и затем вносить в него изменения. Чтобы предотвратить эти действия, вы должны отменить право на чтение. Для этого введите команду:

chmod go-r display<CR>

где g и о обозначают права для группы и для всех пользователей системы;

-г - запрещает читать или копировать файл. -

Проверьте результат и введите команду ls -1. Экран будет выглядеть следующим образом:

$ chmod go-r display<CR> $ ls -1<CR>

total 35

-rwx--x--x 1 startship project 9346 Nov 1 08:06 display

-rw-r--r--1 startship project 6428 Dec 2 10:26 list

drwxr-x--x 2 startship project 32 Nov 8 15:32 tools

$

Вы можете использовать команду chmod для предоставления или отрицания права для справочников, как и для файлов.

Например, вы предоставили право на чтение справочника себе (и), членам вашей группы (g) и всем пользователям системы. Каждый пользователь, имеющий доступ к системе, будет иметь право читать имена файлов, содержащихся в этом справочнике, задав команду ls -1. Аналогично, предоставление права на запись позволяет пользователям создавать новые файлы в справочнике и удалять в нем существующие файлы. Предоставленное право на выполнение в справочнике позволяет обозначенным пользователям перемещаться в этот справочник (и делать его своим текущим справочником) с помощью команды cd.

**Альтернативный метод**

Существует два метода, с помощью которых может быть выполнена команда chmod. Способ, описанный выше, в котором символы r, w и х используются для указания права, называется символическим методом.

Альтернативным методом является восьмеричный метод. Его формат требует от вас указать права с использованием трех восьмеричных цифр (от 0 до 7).

**Дополнительные команды**

В этом пункте описаны три команды, которые могут вам пригодиться при работе с файлами: chown, id, groups. Если вы владелец файла, ваше регистрационное имя размещается в категории owner. Команда chown позволяет владельцу файла изменить собственный ID на любой другой. Например, если вы введете команду:

ls -l display<CR>

на экране появится информация:

-r-xr-xr-x l owner group 9346 Nov 1 08:06 display

Чтобы изменить ваш ID на чей-либо другой (например, sara) вы должны ввести:

chown sara display<CR>

Если вы введете:

ls -l display<CR>

на экране появится сообщение:

-r-xr-xr-x l sara group 9346 Nov 1 08:06 display

Если вы введете команду chown и на экране появится сообщение об ошибке, это будет указывать на то, что системный программист не предусмотрел эту возможность при установке системы. Если вы введете:

id<CR>

то система отобразит ID пользователей и ваш действительный групповой ID (gid). В зависимости от начальной установки системы вы можете принадлежать более чем к одной группе. Чтобы определить, членом какой группы вы являетесь, введите:

groups<CR>

На экране появится список тех групп, членом которой вы являетесь. Таким образом, вы имеете доступ к файлам тех пользователей, чьи групповые ID совпадают с вашими групповыми ID.

Так как вы хорошо знакомы с этими командами, вам необходимо получить более углубленные знания по технике обработки информации. В этом пункте представим три команды:

diff- найти различия между двумя файлами;

grep - поиск в файле по шаблону;

sort - сортировка и слияние файлов.

**Команда diff**

Команда diff обнаруживает и сообщает обо всех различиях между двумя файлами и говорит вам как изменить первый файл, чтобы он был дубликатом второго.

**Синтаксис команды:**

diff файл\_1 фaйл\_2<CR>

Если файл\_1 и файл\_2 идентичны, то система возвращает подсказку. Если есть различия между ними, то команда diff инструктирует вас, как изменить первый файл (с помощью редактора ed), чтобы он соответствовал второму файлу. Система UNIX помечает строки в файл\_1 (которые будут изменены) символом < (меньше), и строки в файл\_2 (шаблон текста) символом > (больше).

Например, вы хотите найти различия между файлами johnson и mcdonough. Файл mcdonough содержит тот же текст письма, что и johnson, но с соответствующими изменениями для различных получателей. Команда diff идентифицирует эти изменения следующим образом:

3, 6сЗ, 6

< Mr. Ron Johnson

< Layton Printing

< 52 Hudson Street

< New York, N.Y.

----

> Mr. J.J.McDonough

> Ubu Press

> 37 Chico Place

> Springfield, N.J.

9c9

< Dear Mr. Johnson:

----

> Dear Mr. McDonough:

Первая строка команды diff:

3, 6сЗ, 6

Это означает, что если вы хотите, чтобы johnson соответствовал mcdonough, вы должны изменить (с) строки с 3-й по 6-ю в файле Johnson на строки с 3-й по 6-ю из файла mcdonough. Команда diff затем отобразит оба набора строк. Если вы выполните эти изменения (с помощью таких редакторов как ed или vi), файл johnson будет идентичен файлу mcdonough. Помните, что команда diff ищет различия только между двумя указанными файлами.

В первой выводной строке команды diff могут появиться следующие буквы:

а - добавить; с - изменить; d - удалить.

Цифры, приведенные вместе с буквой, указывают на строки, которые должны быть модифицированы.

**Команда greр**

Вы можете выдать системе UNIX инструкцию поиска в файле указанного слова, фразы, группы символов с помощью команды grep. Поиск осуществляется по шаблону.

**Синтаксис команды:**

grep шаблон файл(ы)<СR>

Например, чтобы отыскать любую строку, содержащую слово automation в файле johnson, введите:

grep automation johnson<CR> Система ответит:

$ grep automation johnson<CR> and office automation software

$

Вывод состоит из всех строк в файле Johnson, которые содержат шаблон, указанный вами.

Если шаблон содержит много слов или любых символов, которые имеют специальное значение для системы UNIX (например, $, |, \*, ? и т.д.) вводимый шаблон должен быть заключен в кавычки. Например, вы хотите найти строки, содержащие шаблон office automation. Ваша командная строка и ответ системы будут выглядеть следующим образом;

$ grep 'office automation' johnson<CR> and office automation software

$

Но если вы не знаете, какое письмо содержит шаблон office automation, письмо johnson или sanders, можете ввести команду в следующем виде:

$ grep 'office automation' johnson sanders<CR> johnson:and office automation software

$

Выводная строка говорит вам, что шаблон office automation найден только в файле Johnson.

Дополнительно к команде grep система UNIX обеспечивает разновидности этой команды - это команды egrep и fgerp с несколькими ключами, облегчающими поиск.

**Команда sort**

Система обеспечивает эффективное средство для сортировки и слияния файлов. **Синтаксис команды:**

sort фaйл(ы)<CR>

Эта команда приводит к тому, что строки в указанном файле будут отсортированы и слиты в следующем порядке:

строки, начинающиеся с цифры, будут отсортированы по цифрам и перечислены после строк, начинающихся с буквы;

строки, начинающиеся с большой буквы, перечисляются до строк, начинающихся с маленькой буквы;

строки, начинающиеся с таких символов, как "%", "\*" сортируются на основе символьного представления ASCII.

Например, вы имеете 2 файла group 1 и group2, каждый из которых содержит перечень имен. Вы хотите отсортировать каждый список по алфавиту и затем объединить два списка в один. Вначале отобразите содержание файлов, выполнив команду cat для каждого файла. Экран будет выглядеть следующим образом:

$cat group 1<CR>Smith, Allyn

Jones, Barbara

Cook, Karen

Moore, Peter

Wolf, Robert

$ cat group2<CR> Frank, H. Jay

Nelson, James

West, Donna

Hill, Chales

Morgan Kristina

$

Вместо того, чтобы печатать эти два файла индивидуально, вы можете сделать это в одной командной строке. Если вы введете команду:

$ cat group 1 group2<CR> результат будет такой же.

Теперь отсортируйте и слейте эти два файла, выполнив команду sort. Результат выполнения команды sort будет распечатан на экране терминала, если вы не укажете иначе.

$ cat group 1 group2<CR>

Cook, Karen

Frank, H. Jay

Hill, Chales

Jones, Barbara

Moore, Peter

Morgan, Kristina Nelson, James

Smith, Allyn

West, Donna

Wolf, Robert

$

Дополнительно к комбинированию списков (как в примере), команда sort может перестанавливать строки и части строк (называемые полями) в соответствии с другими спецификациями, которые вы назначили в командной строке. Возможными спецификациями являются complex и beyond.