

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра сетей и устройств телекоммуникаций

***КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ
В СЕТЯХ ТЕЛЕКОММУНИКАЦИЙ***

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторно-практическим занятиям
для студентов специальности 45 01 03 «Сети телекоммуникаций»
дневной, вечерней и заочной форм обучения

В 3-х частях

Часть 2

Операционная система DRS-Unix.
Основные службы

Минск 2003

УДК 004 + 621.391 (075.8)
ББК 32.973 + 32.811 я 7
К 63

С о с т а в и т е л и:
В.А. Мельников, Я.В. Рощупкин

Компьютерные технологии в сетях телекоммуникаций: Метод.
К 63 указания к лабораторно-практическим занятиям для студ. спец. 45 01 03
«Сети телекоммуникаций» дневной, вечерней и заочной форм обучения: В
3 ч. Ч. 2. Операционная система DRS-Unix. Основные службы / Сост.
В.А. Мельников, Я.В. Рощупкин. – Мн.: БГУИР, 2003. – 19 с.

Методические указания включают основные теоретические положения, методику выполнения работ и подготовки к защите лабораторно-практических работ.

УДК 004+621.391 (075.8)
ББК 32.973+32.811 я 7

Ч. 1. Операционная система DRS-Unix. Основные особенности работы, работа с файлами, регистрация в системе. Издана в БГУИР в 2003 г.

© Мельников В.А., Рощупкин Я.В.,
составление, 2003
© БГУИР, 2003

1. ЛАБОРАТОРНАЯ РАБОТА № 2

«Работа с электронной почтой»

1.1. Теоретические сведения

Электронная почта – это персональная связь с миром. Все те, кто пользуются электронной сетью, имеют свой адрес электронной почты, или электронный адрес. Все возрастающее число «шлюзов» («gateways») каждый день связывает с сетью все больше и больше людей. Когда пользователь зарегистрировался в своей местной системе, она автоматически создает (генерирует) для него адрес.

Основные принципы, на которых построена электронная почта, подобны основным концепциям построения обычной почты – пользователь посылает людям письма по их конкретным адресам. Но электронная почта имеет серьезные преимущества по сравнению с обычной почтой. Наиболее очевидное – скорость: сообщение будет идти на другой конец мира не несколько дней, а несколько часов, минут или даже секунд (в зависимости от места посылки сообщения и состояния связи между этим местом и адресатом).

Электронная почта имеет преимущества и перед телефоном. Второе преимущество: сообщение отправляется тогда, когда это удобно отправителю, а адресат отвечает тогда, когда это удобно ему. Не надо организовывать одновременное присутствие двух абонентов на двух концах линии. И если телефонный разговор через всю страну быстро выливается в солидный счет, то электронная почта позволяет обмениваться большими объемами информации за гораздо меньшую сумму, даже если адресат живет в Новой Зеландии.

В состав операционной системы UNIX входят утилиты электронной почты, которые позволяют передавать сообщения другим пользователям. Передавать и принимать сообщения можно разными способами, выбор которых зависит от конкретной утилиты электронной почты. В работе далее будет описано использование одной из наиболее популярных программ **mail**. В утилите **mail** используется интерфейс в виде командной строки, имеющей собственный интерпретатор команд. Эта утилита применяется в большинстве UNIX-систем и считается стандартной.

Помимо передачи сообщений по электронной почте вы можете непосредственно связываться с другими зарегистрированными в системе пользователями при помощи утилит **talk** и **write**.

1.1.1. Локальная и internet-адресации

В системе UNIX каждому пользователю присваивается свой почтовый адрес, который нужно указывать при отправке этому пользователю сообщений. Адреса локальных пользователей UNIX-системы могут состоять просто из их регистрационных имен. Если же необходимо отправить сообщение пользователю других систем, то следует указать также и адрес этой системы.

В Internet-адресах используется так называемая доменная адресация: ка-

ждая система входит в определенный домен, имя которого в совокупности с именем системы образует ее уникальный адрес. Имя домена отделяется от имени системы точкой. Имена имеют следующий формат:

Регистрационное_имя@имя_системы.имя_домена

Системам, являющимся частью одной локальной сети, присваивается одинаковое доменное имя. Например, системы в нашем университете, расположенные в аудиториях 506 и 501, называются room506 и room501 соответственно и имеют доменное имя edu. Для того чтобы послать сообщение пользователю term01 системы room506 в систему room501, нужно включить в адрес имя домена: term01@room501.edu. Продемонстрируем пример отправки сообщения этому пользователю при помощи программы **mail**:

```
$ mail term01@room501.edu <mydata
```

1.1.2. Утилита mail

Утилита **mail** – эффективное средство обмена сообщениями с другими пользователями. Процесс отправки сообщения очень прост и включает в себя ввод адреса получателя и текста сообщения. Процесс приема состоит из выбора сообщения из списка полученных сообщений. При отправке сообщения с помощью утилиты **mail** можно пользоваться командами, позволяющими редактировать текст и заголовок сообщения. Получая сообщения, можно отвечать на них, сохранять в файл и просто удалять.

Передача сообщений

Пользуясь утилитой **mail**, можно посылать данные, вводимые с клавиатуры, или содержимое файла. В процессе ввода сообщения с клавиатуры его можно редактировать специальными тильда-командами. Эти команды позволяют сохранять сообщение, выводить набранную информацию на экран и изменять его.

Для того чтобы послать сообщение, необходимо ввести команду **mail** и адрес получателя. После нажатия клавиши **Enter** система попросит ввести тему сообщения. Следует ввести тему и нажать **Enter**. Программа перейдет в режим ввода текста сообщения. Вся вводимая информация воспринимается как текст сообщения, при каждом нажатии **Enter** к тексту добавляется новая строка. Закончив набор сообщения, следует перейти на новую строку и нажать **[Ctrl+d]**, указывая тем самым, что сообщение закончено и его можно отправлять.

Приведем пример отправки сообщения пользователю term05. Тема сообщения «Birthday»:

```
$ mail term05
Subject: Birthday
Hello term05. Happy birthday.
^D
```

EOT
\$

Стандартный ввод и переадресация

Утилита **mail** принимает входную информацию со стандартного ввода. По умолчанию стандартным вводом считаются данные, вводимые с клавиатуры. Стандартный ввод можно организовать так, чтобы ввод данных осуществлялся из файла, тогда содержимое файла будет являться текстом сообщения. Можно создать текст редактором **vi** и затем отправить его по электронной почте. Приведем пример отправки файла по электронной почте:

```
$ mail term05 <datafile
```

Отправляя файл посредством переадресации, пользователь лишается возможности ввести тему сообщения. Для этих целей у команды **mail** есть опция **-s**, которая позволяет указать тему сообщения в командной строке. Ниже представлен список опций команды **mail**:

Опции команды	Описание
-f имя_файла_почтового_ящика	– вызывает mail для чтения сообщений не из почтового ящика, а из отдельного файла в вашем каталоге;
-H	– выдает только список заголовков сообщений;
-s тема	– указывает тему при отправке сообщения;
-v	– выводит на экран последовательность операций mail , используемых для передачи сообщений.

В следующем примере отправляется сообщение пользователю **term05** с темой сообщения «hello»:

```
$ mail -s hello term05 <datafile
```

Передача сообщений нескольким пользователям

Сообщение можно послать сразу нескольким пользователям. Для этого необходимо перечислить адреса получателей в командной строке после команды **mail**:

```
$ mail term05 term06 term07  
Subject: Birthday  
Hello term05. Happy birthday.  
^D  
EOT  
$
```

Также в этом варианте возможна переадресация:

```
$ mail term05 term06 term07 <datafile
```

1.1.3. Редактирование сообщений: тильда-команды

Сообщение состоит из двух частей: заголовка и текста. Заголовок содержит информацию о сообщении, в частности адреса получателей и тему.

Набор специальных команд, известных как тильда-команды, позволяет вам редактировать текст и заголовок сообщения. Тильда-команда состоит из тильды (~) и символа команды и вводится в отдельной строке. Тильда (здесь действует как специальный символ) и символ команды не считаются частью сообщения. Если тильду нужно ввести в сообщение как часть сообщения, то вводится двойная тильда: (~~).

Ниже приведены перечни основных тильда-команд. Получить перечень всех тильда-команд можно при помощи тильда-команды ~?.

Тильда-команды для текста сообщения

В отличие от редактора **vi** во встроенном редакторе **mail** не предусмотрен командный режим. После указания темы сообщения и нажатия **Enter** программа сразу переходит в режим ввода, в котором набирается текст сообщения. Пользователь просто набирает последовательность символов. Единственно возможное корректирующее действие – нажатие клавиши **Backspace** для стирания символа слева от курсора. Никакие другие исправления производить нельзя, перемещать курсор по тексту тоже нельзя.

Все ограничения можно устранить, вызвав редактор **vi** командой ~v. Нужно дать ее в отдельной строке и нажать **Enter**.

Команды сохранения редактора **vi** обеспечивают запись отредактированного текста в сообщение утилиты **mail**, а не в файл. Выйдя из **vi** посредством команды **zz**, пользователь сохраняет информацию в сообщении **mail** и возвращается в режим ввода. Текст сообщения при этом не воспроизводится, вместо него на экране появляется слово **continue**, заключенное в круглые скобки. После этого можно продолжить обычную работу с **mail**.

Существуют тильда-команды, которые позволяют записывать сообщение в файл (~w) и вставлять содержимое файла в сообщение (~r). После ввода команды необходимо указать имя файла и нажать **Enter**. В следующем примере команда ~w **myfile** вызывает запись введенного сообщения в файл **myfile**, затем команда ~r **mynames** вставляет содержимое файла в сообщение:

```
$ mail term05
Subject: Birthday
Hello term05. Happy birthday.
~w myfile
"myfile" 1/29
~r mynames
"mynames" 3/15
```

~p

Message contains:

To: term05

Subject: Birthday

Hello term05. Happy birthday.

term01

term02

term03

(continue)

^D

EOT

\$

Если пользователь в процессе составления сообщения передумает послать сообщение, можно выйти из **mail** с помощью команды **~x** или **~q**:

\$ **mail term05**

Subject: **Birthday**

Hello term05. Happy birthday.

~x

\$

Ниже приведены ряд тильда-команд и их описание:

- ~v** – вызывает редактор **vi**, изменения записываются в тексте сообщения;
- ~p** – повторно выводит текст сообщения на экран;
- ~q** – используется для выхода из сообщения и утилиты **mail**;
- ~w имя_файла** – записывает сообщение в файл;
- ~r имя_файла** – считывает содержимое файла в текст сообщения;
- ~e** – вызывает текстовый редактор по умолчанию;
- ~m список_сообщений** – при отправке сообщений и ответов на полученную почту вставляет содержимое полученных сообщений (с отступом). Применяется при подготовке сообщений;
- ~f список_сообщений** – при отправке сообщений и ответов на полученную почту вставляет содержимое полученных сообщений (без отступа). Применяется при подготовке сообщений.

Тильда-команды для заголовка сообщения

Существуют тильда-команды, которые позволяют изменять компоненты заголовка сообщения. Заголовок может состоять из четырех частей: списка адре-

сов получателей, темы сообщения, списков рассылки копий и неполных копий. Последние два компонента не обязательны. Всегда обязателен только список адресов получателей.

Список адресатов вводится при вызове **mail** в командной строке; если при вводе сообщения возникла необходимость добавить в этот список еще кого-либо, то следует воспользоваться командой **~t**.

Команда **~s** позволяет ввести новую тему сообщения. В следующем примере проиллюстрирован процесс изменения темы сообщения и добавления еще одного адресата в список:

```
$ mail term05
Subject: Birthday
Hello term05. Happy birthday.
~t term06
~s Your birthday
~p
-----
Message contains:
To: term05 term06
Subject: Your birthday
Hello term05. Happy birthday.
(continue)
^D
```

EOT

```
$
```

Часто возникает необходимость послать копию сообщения кому-либо еще. В этой копии будут указаны адрес получателя, текст и заголовки сообщения. В таких случаях необходимо с помощью команды **~c** создать список рассылки копий:

```
$ mail term05
Subject: Birthday
Hello term05. Happy birthday.
~c term06
~p
-----
Message contains:
To: term05
Subject: Your birthday
Cc: term06
Hello term05. Happy birthday.
(continue)
^D
```

EOT

```
$
```


Ниже приведен список тильда-команд для сообщений:

- ~h** – приглашает пользователя ввести адреса, тему и список рассылки копий;
- ~s тема** – вводит новую тему;
- ~t адреса** – добавляет адреса к списку адресов;
- ~c адреса** – добавляет адреса к списку рассылки копий;
- ~b адреса** – добавляет адреса к неполному списку рассылки копий.

Тильда-команды общего назначения:

- ~?** – выводит на экран список всех тильда-команд;
- ~~** – вставляет тильду в текст сообщения;
- ~! команда** – выполняет указанную команду shell.

1.1.4. Прием почты

Mail – сложная программа управления сообщениями с собственным интерпретатором команд и своим символом приглашения в командной строке.

Вход в среду **mail** возможен лишь в случае наличия у вас сообщений, а при их отсутствии вы получите уведомление. В следующем примере проиллюстрирован такой случай:

```
$ mail
Sorry? no mail
$
```

Когда пользователь вызывает программу **mail**, ему выдается список заголовков сообщений. Информация представлена в виде полей. В первом указан статус сообщения, обозначается одной прописной буквой (N – новое, U – непрочитанное). За полем статуса идет номер сообщения, следующее поле – адрес отправителя, далее дата и время получения, количество строк и знаков в сообщении и в конце тема сообщения. После списка сообщений **mail** выдает свое приглашение – знак амперсанда. Например:

```
$ mail
Mail version 5.5-kw 5/30/95. Type ? for help.
"/var/spool/mail/term01": 3 messages 3 new
>N 1 term05 Tue Feb 11 10:14:33 5/44 "Test"
  N 2 term04 Wed Feb 12 12:24:21 24/144 "Test 2"
  N 3 term02 Fri Feb 14 8:10:45 11/244 "Test 3"
&
```

Утилита **mail** обращается к сообщениям либо через их список, либо посредством маркера текущего сообщения (>), отмечающего в списке сообщений текущее.

Обращаться к сообщениям можно через их список. Многие команды утилиты могут обрабатывать несколько сообщений одновременно. Для обращения ко всем сообщениям из предыдущего примера нужно указать диапазон 1-3. Также утилита «понимает» специальные символы для обозначения отдельных сообщений из списка. Например, комбинация **^-3** задает диапазон с первого по третье сообщение, а **2-\$** – со второго по последнее. Точка обозначает текущее положение, а звездочка ***** – все сообщения.

К сообщениям можно обращаться и по статусу. Для этого вводятся двоеточие и символ статуса. Например, для обращения ко всем новым сообщениям можно применить следующую комбинацию: **:n**, команда **p :n** выводит на экран все новые сообщения.

1.1.5. Вывод сообщений на экран

Для вывода сообщения на экран нужно просто ввести его номер и сообщение будет выведено на экран постранично. Для отображения следующей страницы необходимо нажать клавишу пробела или Enter.

Для вывода сообщений на экран применяется также команда **p**, которая «понимает» абсолютную и относительную нумерацию. При вводе команды **p** и указании номера сообщения (или диапазона сообщений) будут выведены соответствующие сообщения, а при указании **p+2** будет выведено второе по счету следующее за текущим сообщение. Соответственно возможно применение команды **p-3**. Ниже приведены примеры использования команды **p**:

- p \$** – выдает на экран последнее сообщение;
- p *** – выдает на экран все сообщения;
- p ^-3** – выдает на экран сообщения с первого по третье;
- p .-\$** – выдает на экран сообщения с текущего по последнее;
- p n** – выдает на экран следующее новое сообщение;
- p +2** – выдает на экран второе относительно текущего сообщение;
- p /budget** – выдает на экран сообщение, в поле темы которого указано «budget»;
- p term02** – выдает на экран сообщения, посланные пользователем term02;
- p :n** – выдает на экран новые сообщения;
- p :u** – выдает на экран непрочитанные сообщения;
- p :r** – повторно выдаст на экран уже прочитанные сообщения.

После вывода сообщения на экран вновь появится приглашение **&**, но без списка сообщений. Для повторного вывода списка дайте команду **h**.

1.1.6. Удаление и восстановление сообщений

При выходе из программы все прочитанные сообщения сохраняются. Для удаления сообщения служит команда **d**. Ее нужно вводить с номером сообщения, например команда **d 2** удалит сообщение за номером два. Также возможно применение списков. Так, команда **d 1-3** удалит сообщения с первого по третье. Если команда **d** используется без указания номера сообщения, то удаляться будет текущее сообщение. В следующем примере удаляется сообщение за номером два:

```
$ mail
Mail version 5.5-kw 5/30/95. Type ? for help.
"/var/spool/mail/term01": 3 messages 3 new
>N 1 term05  Tue Feb 11 10:14:33 5/44  "Test"
  N 2 term04  Wed Feb 12 12:24:21 24/144 "Test 2"
  N 3 term02  Fri Feb 14  8:10:45 11/244 "Test 3"
& d 2
& h
>N 1 term05  Tue Feb 11 10:14:33 5/44  "Test"
  N 3 term02  Fri Feb 14  8:10:45 11/244 "Test 3"
&
```

Для восстановления одного из удаленных сообщений, не выходя из программы, введите команду **u [номер]**. Восстановление возможно потому, что реально удаление сообщений происходит не сразу, а при выходе из программы.

1.1.7. Отправка ответов на сообщения

Для подготовки и отправки ответа на полученное сообщение используются команды **r** и **R**. Команда **R** посылает ответ только отправителю, а команда **r** отправителю и всем получателям исходного сообщения. На данные команды распространяются все правила выбора сообщения. Данные команды формируют ответное сообщение, подставляют в поле получателя(ей) необходимые адреса; в поле темы перед темой отправляемого сообщения вставляется **Re:**. Далее программа переходит в режим ввода текста сообщения.

1.1.8. Выход из программы mail

Команда **q** обеспечивает выход из программы в **shell**. Прочитанные сообщения перемещаются в файл **mbox** вашего домашнего каталога. Если по каким-либо причинам пользователю нежелательно перемещение какого-либо сообщения, то перед выходом необходимо дать команду **pre** с номером сообщения. Указанное сообщение останется в почтовом ящике.

Не прочитанные в предыдущей сессии сообщения при очередном входе в программу будут отмечены буквой **U**:

```
>N 1 term05  Tue Feb 11 10:14:33 5/44  "Test"
  N 2 term04  Wed Feb 12 12:24:21 24/144 "Test 2"
```

```
N 3 term02  Fri Feb 14 8:10:45 11/244 "Test 3"  
& q  
Saved 1 message in mbox  
$
```

Командой **x** при выходе из программы все сообщения, удаленные в данном сеансе, будут восстановлены.

1.2. Утилиты отправки коротких сообщений

Система UNIX имеет набор программ для отправки коротких сообщений пользователям системы. Они могут использоваться и для оперативного уведомления пользователей. Ярким примером использования данных программ является процесс останова (выключения) системы, когда система автоматически уведомляет всех активных пользователей о скором останове системы и предлагает завершить работу. Ниже приведены дополнительные команды:

1. finger – программа получения информации о пользователях. Формат использования команды следующий:

```
$ finger [-lmsp] [user ...] [user@host ...]
```

Программа имеет следующие опции:

- s – показывает имя пользователя, краткую информацию о пользователе, название терминала, время нахождения пользователя в системе;
- l – предоставляет ту же информацию, что и с ключом –s, а также показывает дополнительную информацию о пользователе, содержащуюся в файлах ".plan", ".project" и ".forward".

Если не было указано никаких аргументов, программа покажет информацию о всех зарегистрированных в данный момент в системе пользователях.

Программа **finger** может использоваться для получения информации о пользователях другой системы, в этом случае необходимо указать аргумент в формате "**пользователь@машина**" или "**@машина**". В данном случае может использоваться только опция –l;

2. write – программа для отправки короткого сообщения другому пользователю. Формат использования:

```
$ write пользователь [терминал]
```

После ввода команды следует набрать сообщение и в конце нажать сочетание клавиш <Ctrl+d>. При использовании программы удаленный пользователь получает следующее сообщение:

```
Message from yourname@yourhost on yourtty at hh:mm  
...
```

После заголовка сообщения следуют строки, посылаемые пользователем.

Если адресат сообщения зарегистрировался на нескольких терминалах, при запуске программы можно указать имя терминала. В противном случае программа выберет терминал автоматически с наименьшим временем бездействия (отсутствия ввода пользователем);

3. wall – программа для отправки широковещательного сообщения всем зарегистрированным в данный момент пользователям. Формат использования:

```
$ wall [сообщение]
```

2. ЛАБОРАТОРНАЯ РАБОТА № 3 «Система передачи файлов ftp»

Одним из неоспоримых преимуществ подключения к Интернету является возможность легкого доступа к свободно распространяемому программному обеспечению. К сожалению, процесс соединения с удаленной машиной и получения интересующего программного обеспечения не так прост.

В простейшем случае две UNIX-системы соединены в локальную сеть и есть возможность использовать программу **ftp** для копирования файлов с удаленной системы и на удаленную систему. Название **ftp** – это сокращение от File Transfer Protocol (протокол передачи файлов). В настоящее время программа **ftp** является самым популярным способом передачи программного обеспечения по сети. Использовать ее очень просто. Для запуска следует ввести:

```
$ ftp  
ftp>
```

После запуска программы приглашение командной строки изменилось на приглашение программы **ftp**. Теперь можно вводить команды программы.

Установить соединение с удаленной системой можно двумя способами. Во-первых, можно указать имя системы при запуске **ftp**:

```
$ ftp machine_kevin
```

или же использовать команду **open**:

```
$ ftp  
ftp> open machine_kevin  
Connected to machine_kevin
```

2.1. Анонимное FTP соединение

Обычно для применения команды **ftp** с целью передачи файлов необходимо быть зарегистрированным на удаленной машине пользователем, т.е. иметь

системный идентификатор (имя пользователя) и соответствующий пароль для входа в удаленную систему. Однако при большом количестве пользователей регистрация каждого из них становится неудобной, и в этом случае используется так называемое анонимное **ftp**-соединение. При входе на удаленную машину с помощью программы **ftp** пользователь должен в качестве системного идентификатора пользователя ввести ключевое слово **anonymous**. При этом права на удаленной машине будут урезаны до минимума – можно лишь получать и иногда отправлять файлы, что и требовалось.

Чтобы использовать анонимное **ftp**-соединение, необходимо открыть **ftp**-соединение с удаленной системой, ввести в качестве системного идентификатора пользователя ключевое слово **anonymous** и свой адрес электронной почты в качестве пароля:

```
ftp> open machine_kevin
Connected to machine kevin
Name (machine kevin): anonymous
220 Guest login ok, send ident as password.
Password: pupkin@tyt.by
220 Guest login ok, access restriction apply.
```

Теперь можно использовать обычные команды **ftp**.

2.2. Список команд FTP

Ниже приведен список команд, которые можно использовать после установки соединения с удаленной системой:

- ? – вывод справки по указанной команде;
- Append** – файл1 файл2 – присоединение файла с именем **файл1** на локальной системе к файлу с именем **файл2** на удаленной системе;
- Ascii** – установка режима передачи в состояние «текст» – это значение по умолчанию;
- Bell** – подача звукового сигнала по окончании передачи файла;
- Binary** – установка режима передачи в значение «двоичные данные»;
- Bye** – закрытие соединения и завершение работы программы **ftp**;
- cd** <каталог> – смена текущего каталога на удаленной системе – это обычная команда UNIX;
- Cdup** – переход в родительский каталог текущего каталога – это то же, что и **cd**;
- Close** – закрытие соединения без завершения работы программы **ftp**;
- Debug** – включение/выключение режима отладки (по умолчанию выключен);

delete <файл>	– удаление указанного файла на удаленной системе;
Dir <каталог><файл>	– вывод списка файлов в указанном каталоге удаленной системы. Если указан аргумент файл , то полученный список сохраняется в этом файле;
Disconnect	– закрытие соединения и завершение работы программы ftp ;
get <файл1> <файл2>	– получить с удаленной системы файл файл1 и сохранить его под именем файл2 . Если аргумент файл2 не указан, то файл будет сохранен под исходным именем. Те же действия выполняет команда recv ;
Hash	– вывод состояния процесса передачи файлов – будет отмечаться получение каждого блока;
Help <команда>	– вывод справки об указанной команде; если аргумент команда не указан, выводится общая справочная информация;
lcd <каталог>	– переход в указанный каталог на локальной системе;
ls <каталог/файл>	– вывод содержимого каталога (если каталог не указан, выводится содержимое текущего каталога). Если указано имя файла, выводится информация о данном файле – это обычная команда UNIX;
mdelete <файл(ы)>	– удаление указанных файлов на удаленной системе;
mdir <файл(ы)>	– вывод информации об указанных файлах;
mget <файл(ы)>	– получение указанных файлов с удаленной системы;
mkdir <каталог>	– создание нового каталога с заданным именем на удаленной системе – это обычная команда UNIX;
mput <файл(ы)>	– отправка указанных файлов на удаленную систему;
open <имя_системы>	– открытие соединения с удаленной системой; если аргумент имя_системы не указан, программа запросит у вас имя системы, с которой нужно установить соединение;
put <файл1> <файл2>	– копирование файла файл1 с локальной системы на удаленную и сохранение его под именем файл2 . Если аргумент файл2 не указан, файл будет сохранен под исходным именем. Те же действия выполняет команда send ;
Pwd	– вывод имени текущего каталога на удаленной машине;
Quit	– разрыв соединения и завершение работы программы ftp ;
recv <файл1> <файл2>	– то же, что и get ;
Remotehelp <команда>	– запрос у удаленной системы справки по указанной команде;

rename <файл1> <файл2> – переименование файла на удаленной системе;
rmdir <каталог> – удаление каталога на удаленной системе – это стандартная команда UNIX;
send <файл1> <файл2> – то же, что и **put**.

Ниже приведен полный пример анонимного сеанса связи с хостом `primergy350`, в котором будет принят файл `file3.txt`:

```
$ ftp primergy350.sis.com.by
220 Connected to primergy350.SIS.Com.By.
User (primergy350.SIS.Com.By:(none)): anonymous
220 Guest login ok, send ident as password.
Password: user@myhost.edu
220 Guest login ok, access restriction apply.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
file1.txt
file2.txt
file3.txt
file4.txt
file5.txt
file6.txt
file7.txt
file8.txt
226 Transfer complete.
ftp: 133 bytes received in 0.05Seconds 2.66Kbytes/sec.
ftp> get file3.txt
200 PORT command successful.
150 Opening BINARY mode data connection for file1.txt(9090 bytes).
226 Transfer complete.
ftp: 9090 bytes received in 5.66Seconds 25.92Kbytes/sec.
ftp> quit
221 PRIMERGY350> Bye!
$
```


3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 3.1. Изучить теоретические сведения и методические указания.
- 3.2. После предварительной беседы с преподавателем получить у него конкретное задание и номер рабочего места.
- 3.3. После запуска сервера преподавателем (лаборантом) приступить к конкретному выполнению задания.
- 3.4. После окончания выполнения задания предъявить результаты преподавателю (в электронном либо бумажном виде).
- 3.5. Выйти из системы, сдать рабочее место.

4. СОДЕРЖАНИЕ ОТЧЕТА

- 4.1. Описание процедуры входа в систему и регистрации.
- 4.2. Описание конкретного задания по набору и форматированию текста.
- 4.3. Текст созданных файлов.
- 4.4. Описание работы с файлами – создание, копирование, пересылка по электронной почте и т.д.
- 4.5. Ответы на пункты, конкретно заданные преподавателем.

ЛИТЕРАТУРА

1. Банахан М., Раттер Э. Введение в операционную систему UNIX. – М.: Радио и связь, 1986. – 341 с.
2. Браун С. Операционная система UNIX. – М.: Мир, 1986. – 462 с.
3. Браун П. Введение в операционную систему UNIX. – М.: Мир, 1987. – 287 с.
4. Готье Р. Руководство по операционной системе UNIX. – М.: Финансы и статистика, 1985. – 232 с.
5. Керниган Б.В., Пайк Р. UNIX – универсальная среда программирования. – М.: Финансы и статистика, 1992, – 304 с.
6. Кристиан К. Введение в операционную систему UNIX. – М.: Финансы и статистика, 1985.
7. Немет Э. UNIX – руководство системного администратора. – Киев: ВНУ, 1997.
8. Петерсен Р. Linux – руководство по операционной системе. – Киев: ВНУ, 1997.
9. Тихомиров В.П., Давидов М.И. Операционная система UNIX: Инструментальные средства программирования. – М.: Финансы и статистика, 1988. – 206 с.
10. Томас Р., Йейтс Дж. Операционная система UNIX. Руководство для пользователей. – М.: Радио и связь, 1986. – 352 с.
11. Топхем Д., Чьюнг Х.В. UNIX и XENIX. – М.: Мир, 1988. – 392 с.
12. Ирвин Дж., Харль Д. Передача данных в сетях: Инженерный подход. Учеб. пособие. – СПб.: ВНУ, 2003.

Учебное издание

**КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ
В СЕТЯХ ТЕЛЕКОММУНИКАЦИЙ**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторно-практическим занятиям
для студентов специальности 45 01 03 «Сети телекоммуникаций»
дневной, вечерней и заочной форм обучения

В 3-х частях
Часть 2
Операционная система DRS-Unix.
Основные службы

С о с т а в и т е л и :
Мельников Владимир Александрович,
Рощупкин Яков Викторович

Редактор Н.А. Бебель
Корректор Е.Н. Батурчик
Компьютерная верстка Т.В. Шестакова

Подписано в печать 19.11.2003.
Печать ризографическая
Уч.-изд. л. 0,9.

Формат 60x84 1/16.
Гарнитура «Таймс».
Тираж 50 экз.

Бумага офсетная.
Усл. печ. л. 1,28.
Заказ 220.

Издатель и полиграфическое исполнение:
Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Лицензия ЛП № 156 от 30.12.2002.
Лицензия ЛВ № 509 от 03.08.2001.
220013, Минск, П. Бровка, 6.