

4. Файловая структура и периферийные устройства

Содержание

4. Файловая структура и периферийные устройства.....	4-1
4.1. Введение.....	4-2
4.2. Литература.....	4-2
4.3. Периферийные устройства как часть аппаратных средств компьютера	4-2
4.3.1. Компоненты среды аппаратных средств.....	4-2
4.3.2. Периферийные устройства	4-4
4.3.3. Имена устройств в OpenVMS.....	4-6
4.4. Имена файлов и хранение их в каталогах	4-7
4.4.1. Спецификации файлов OpenVMS.....	4-7
4.4.2. Использование шаблонов в спецификациях файлов.....	4-8
4.4.3. Организация каталогов файлов	4-9
4.4.4. Просмотр информации о файлах.....	4-13
4.5. Создание и редактирование файлов	4-16
4.5.1. EVE, расширяемый редактор для VAX.....	4-16
4.5.2. Вызов EVE.....	4-17
4.5.3. Формат экрана EVE	4-17
4.5.4. Ввод команд EVE	4-18
4.5.5. Ввод текста	4-20
4.5.6. Замена текста.....	4-21
4.5.7. Передвижение по документу.....	4-22
4.5.8. Окончание сеанса редактирования и сохранение файла	4-22
4.5.9. Получение вспомогательной информации	4-23
4.5.10. Некоторые расширенные возможности EVE.....	4-24
4.5.11. Приложение: текстовый редактор EDT.....	4-25
4.6. Работа с файлами.....	4-29
4.6.1. Копирование файлов	4-29
4.6.2. Просмотр содержимого файлов	4-31
4.6.3. Печать файлов.....	4-32
4.6.4. Переименование файлов	4-39
4.6.5. Удаление файлов.....	4-40
4.7. Выводы	4-43
4.8. Письменные упражнения	4-44
4.8.1. Спецификации файлов VMS.....	4-44
4.8.2. Использование шаблонов	4-45
4.8.3. Каталоги.....	4-45
4.8.4. Команды работы с файлами.....	4-45
4.9. Лабораторные упражнения.....	4-46
4.9.1. Имена файлов и хранение их в каталогах.....	4-46
4.9.2. Создание и редактирование файлов.....	4-46

4.9.3. Работа с файлами	4-47
4.10. Решения к письменным упражнениям	4-48
4.10.1. Спецификации файлов VMS.....	4-48
4.10.2. Использование шаблонов	4-49
4.10.3. Каталоги	4-49
4.10.4. Команды работы с файлами	4-50

4.1. Введение

Когда Вы работаете в ОС OpenVMS, Вы входите в среду, состоящую из устройств, программ и данных. Устройства, которые физически составляют ЭВМ, называются **аппаратным обеспечением**. Программы, которые управляют аппаратным обеспечением, называются **программным обеспечением**. Для решения в системе стоящих перед программистом или пользователем задач необходимо использование как программного, так и аппаратного обеспечения. Периферийные устройства обеспечивают ввод, вывод и долговременное хранение информации. Они являются частью аппаратного обеспечения компьютера. Информация хранится на периферийных устройствах в виде файлов. Почти каждая операция, выполняемая пользователем в процессе терминальной сессии, включает в себя использование файлов. Чтобы использовать файл, Вы должны знать его имя и местоположение. Если Вы создали много файлов, полезно сгруппировать их в структуру подкаталогов, чтобы облегчить доступ к ним.

В данной главе кратко рассматривается состав аппаратного обеспечения компьютера VAX, а также более подробно рассматривается организация файлов, создание и редактирование текстовых файлов и различные файловые операции, наиболее часто выполняемые пользователями OpenVMS.

4.2. Литература

VMS DCL Concepts Manual

VMS DCL Dictionary

VMS Installation and Operation Guide для Вашей конкретной системы VAX

VAXSystems/DECSystems Systems and Options Catalog

VMS EVE Reference Manual

Guide to VMS Text Processing

VAX Text Processing Utility Manual

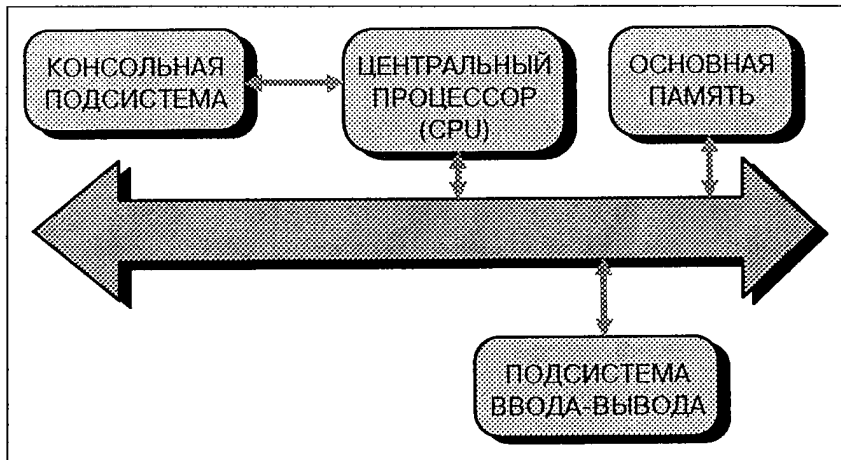
4.3. Периферийные устройства как часть аппаратных средств компьютера

4.3.1. Компоненты среды аппаратных средств

Аппаратное обеспечение системы VAX делится на четыре подсистемы:

- Центральный процессор
- Консольная подсистема
- Основная память
- Подсистема ввода/вывода

Рис. 4-1. Подсистемы аппаратного обеспечения системы VAX



4.3.1.1. Центральный процессор (ЦП)

Основным назначением ЦП является исполнение инструкций. Только одна инструкция может исполняться в один момент времени.

Существует ряд различных ЦП VAX которые отличаются по:

- Размерам
- Скорости обработки

4.3.1.2. Консольная подсистема

Консольная подсистема обменивается информацией непосредственно с ЦП и в основном используется для:

- Загрузки и останова системы
- Установки программного обеспечения
- Удаленной диагностики аппаратного обеспечения

Консольная подсистема состоит из:

- Индикаторов и управления (реализовано программно на некоторых моделях процессоров VAX)
- Консольного терминала
- Консольного накопителя (отсутствует на некоторых моделях процессоров VAX)

Консольная подсистема работает в двух режимах:

- **Консольный режим:** Воспринимает команды консоли
- **Программный режим:** Используется как обычный терминал операционной системы

4.3.1.3. Основная память

Основная память используется для хранения инструкций и данных

Существует два типа памяти в системах VAX:

- Постоянное запоминающее устройство (ПЗУ)
 - Специальная память, не допускающая стирания.
 - Используется для хранения информации доступной только на чтение.

- ↪ Это могут быть программы/данные для загрузки или программы которые должны быть доступны в любое время вне зависимости от состояния компьютера.
- ↪ ПЗУ не чувствительно к сбоям питания, так как информация "прошита" в нем
- Оперативное запоминающее устройство (ОЗУ)
 - ↪ Запоминающее устройство с произвольной выборкой данных (ЗУПВ), память допускающая как чтение, так и запись.
 - ↪ Внутренняя рабочая область компьютера
 - ↪ Вся информация должна находиться в ОЗУ до того как ЦП сможет ее использовать.
 - ↪ ОЗУ содержит программы и данные которые могут быть смешаны.
 - ↪ ОЗУ является временной и не может хранить информацию без электрического питания.

Различные типы систем VAX поддерживают разные объемы памяти.

4.3.1.4. Подсистема ввода/вывода

Подсистема ввода/вывода состоит из устройств которые обеспечивают ввод и вывод информации из системы. Эти устройства называются **периферийными устройствами**.

Как правило, периферийные устройства включают:

- Терминалы
- Печатающие устройства
- Накопители на магнитных дисках (НМД)
- Накопители на магнитных лентах (НМЛ)

4.3.2. Периферийные устройства

Digital Equipment Corporation производит и поддерживает широкий набор терминалов, печатающих устройств и НМД для использования в системах VAX.

4.3.2.1. Терминалы

Терминалы - это периферийные устройства которые имеют клавиатуру (иногда дополненную устройством типа "мышь") и видеомонитор или печатающее устройство.

Под управлением программного обеспечения терминал позволяет пользователю набирать команды и данные на клавиатуре и получать сообщения на видеомонитор или печатающее устройство.

Существует два типа терминалов:

- Печатающий терминал
- Видеотерминал
 - ↪ Текстовый - построчный видеотерминал, который не поддерживает графику (например, VT220, VT320, VT420, VT510, VT520, VT525 - текстовые терминалы, различающиеся по своим возможностям)
 - ↪ Графический: оконная система которая позволяет пользователю выводить графику (например, VT240, VT340, VT340+, а также семейство X-терминалов и графические адаптеры рабочих станций)

4.3.2.2. Печатающие устройства и графопостроители

Печатающие устройства (принтеры) обеспечивают вывод информации в системе. Они выпускаются различных размеров и типов, например:

Построчные печатающие устройства

Высокоскоростные, рассчитанные на большую загрузку печатающие устройства, они используются для больших объемов печати, как правило с низким качеством печати. При печати используется технология, когда вся строка печатается одновременно.

Высококачественные печатающие устройства

Используются для вывода из текстовых процессоров, могут использовать технологию пишущей машинки ("ромашка") или использовать головки контактирующие с красящей лентой в нужный момент. Качество печати высокое, но как правило, скорость печати не велика. Эти печатающие устройства, как правило используют различные гарнитуры (начертания шрифтов).

Лазерные принтеры

Высококачественные печатающие (графические) устройства, использующие комбинацию лазерной и фотокопировальной технологий. Это позволяет им использовать большое количество гарнитур и выводить качественные графические изображения. Скорость печати варьируется от средней до высокой.

Графические печатающие устройства (принтеры/плоттеры)

Текстовый и графический вывод

4.3.2.3. Накопители на дисках и лентах

Накопители на магнитных дисках (НМД) и лентах (НМЛ) - это устройства которые записывают и считывают информацию с магнитного носителя. Кроме того существуют накопители на оптических дисках, которые с точки зрения работы программного обеспечения аналогичны НМД.

4.3.2.3.1. Накопители на дисках

Иногда называются внешней памятью

Иногда называются устройствами массовой памяти

Диски, используемые в накопителях, называются носителями информации

Различные типы дисков включают:

- Кассетные
- Пакеты дисков
- Гибкие диски (так же "дискеты")
- Два типа оптических дисков:

Read-Only - только на чтение (CDROM)

Write-Once, Read-Many - для однократной записи (WORM)

Диски бывают сменные и фиксированные.

4.3.2.3.2. Накопители на магнитной ленте (НМЛ)

НМЛ записывают и считывают информацию с магнитной ленты (МЛ)

На МЛ обычно хранят:

- Резервные копии данных
- Редко используемые данные

Катушечные ленты

- Различаются по длине
- Различаются по плотности записи

Кассетные ленты

- Похожи на бытовые аудио- и видеокассеты

4.3.2.3.3. Иерархический контроллер внешней памяти HSC

HSC это интеллектуальное устройство по обслуживанию НМД и НМЛ.

- Выполняет все операции ввода/вывода для НМД
- Позволяет совместное использование данные в системах VMScluster
- Может использоваться в одиночной системе VAX

Примеры HSC контроллеров:

подсистема HSC50

подсистема HSC70

4.3.3. Имена устройств в OpenVMS

Все устройства в системе имеют уникальные имена в формате **ddcu**.

dd Двухбуквенный код устройства, который определяет тип устройства

c Однобуквенный код, который определяет аппаратный номер контроллера устройства. (Контроллер обеспечивает интерфейс между шиной и устройством или между двумя шинами.)

Аппаратный номер контроллера:

Определяет контроллер устройства

Номер представляется буквой от A до Z

Присваивается системой

u Номер устройства:

Определяет позицию устройства на контроллере

Может быть изменён:

Включением кнопок или переключателей на

устройстве

Установкой перемычек на устройстве

Таблица 4-1 показывает некоторые двухбуквенные коды устройств.

Таблица 4-1. Некоторые двухбуквенные коды устройств

Код	Устройство
CS	Устройство консольного накопителя
DU	НМД типа RA80 или RA81
LP	Построчное печатающее устройство
MF	НМЛ типа TU78

4.4. Имена файлов и хранение их в каталогах

4.4.1. Спецификации файлов OpenVMS

Спецификации файлов используются для предоставления пользователю и системе информации, необходимой для идентификации файла.

4.4.1.1. Что такое файл

Файл - это собрание информации. Эта информация может представлять собой:

- Данные, которые компьютер может прочитать и понять
- Текст, который вводит и с которым работает пользователь

Текстовый файл может содержать:

- Текст памятной записки
- Программу, написанную на языке программирования, таком как VAX FORTRAN
- Список адресов
- Выходную информацию команды DCL

4.4.1.2. Структура спецификации файла OpenVMS

Ниже следует пример простой спецификации файла:

APRIL_REPORT.TXT

Формат полной спецификации файла OpenVMS показан на Рис. 4-2

Рис. 4-2. Спецификация файла OpenVMS

УЗЕЛ::УСТРОЙСТВО:[КАТАЛОГ]ИМЯ_ФАЙЛА.ТИП;ВЕРСИЯ

Компоненты полной спецификации файла описывает Таблица 4-2

Таблица 4-2. Компоненты спецификации файла OpenVMS

Компонент	Описание
Узел (Node)	Имя узла сети, максимум 6 символов
Устройство (Device)	Имя устройства

Компонент	Описание
Каталог (Directory)	Имя каталога
Имя файла (Filename)	Имя файла, максимум 39 символов. Допустимые символы: Буквы от А до Z, верхнего и нижнего регистров Цифры от 0 до 9 Знак подчеркивания (_) Дефис (-) Знак доллара (\$)
Тип (Type)	Тип файла, максимум 39 символов, слева отделяется точкой (.) Допустимые символы те же, что и у имени файла
Версия	Номер версии файла, десятичное число от 1 до 32767, должно предваряться точкой с запятой (;) или точкой (.)

4.4.2. Использование шаблонов в спецификациях файлов

Иногда полезно специфицировать группу файлов, имеющих общие части файловой спецификации.

Например, Вам может понадобиться удалить все файлы данного типа в определенном каталоге.

Шаблонные символы используются:

- Для указания более одного файла
- Для сокращения спецификации файла
- Для отождествления с одним или несколькими символами в именах каталогов, именах файлов и типах файлов
- Для указания всех версий файла или файлов
- Совместно друг с другом или по отдельности

Типы шаблонных символов:

- Звездочка (*)
- Знак процента (%)

Шаблонные символы НЕ могут использоваться вместо или как часть:

- Имени узла
- Имени устройства

Шаблонные символы описывает Таблица 4-3.

Таблица 4-3. Шаблонные символы, используемые в именах, типах и версиях файлов

Символ	Значение
Звездочка (*)	Соответствует 0-39 символам в имени, типе или номере версии файла
Знак процента (%)	Соответствует ровно одному символу в имени или типе

файла

Таблица 4-4 иллюстрирует использование шаблонов в командах DCL. Сами команды рассматриваются в других главах.

Таблица 4-4. Использование шаблонов в командах DCL

Пример команды	Комментарии
\$ DIRECTORY *.TXT	Перечисляет имена всех версий файлов, имеющих тип TXT, независимо от их имени файла
\$ TYPE MY*.*	Выводит на экран содержимое последней (с наибольшим номером) версии каждого файла, имеющего имя файла, две первые буквы которого MY, независимо от типа файла.
\$ COPY CURRENT_MONTH.TXT MARCH.*	Копирует содержимое CURRENT_MONTH.TXT в файл MARCH.TXT. Шаблон в спецификации выходного файла говорит "сохрани эту часть такой же, как и в спецификации входного файла". Это верно также и для команды RENAME.
\$ DELETE REPORT.TXT;*	Удаляет все версии файла REPORT.TXT.)
\$ DELETE/CONFIRM %%%STATUS.TXT;*	Удалить все версии любого файла, который имеет девятисимвольное имя файла, заканчивающееся на "STATUS", и тип TXT. Когда Вы удаляете файл с использованием шаблона, неплохая идея включить квалификатор /CONFIRM, чтобы предотвратить случайное удаление файла, который Вы хотели бы сохранить.
\$ PURGE DATA%.TMP	Чистит (удаляет старые версии) все файлы, имеющие пятисимвольное имя файла, начинающееся с DATA, и тип файла TMP.

4.4.3. Организация каталогов файлов

В операционной системе VMS информация хранится в иерархической структуре.

На вершине этой иерархии находится *головной каталог файлов* (master file directory, MFD).

Ваш *каталог файлов пользователя* (user file directory, UFD) перечислен наряду с UFD других пользователей в головном каталоге файлов.

Ваш UFD часто называют Вашим *входным каталогом* или *каталогом по умолчанию*, потому что это тот каталог, который Вы используете по умолчанию, когда входите в систему.

4.4.3.1. Подкаталоги

В дополнение к головному каталогу файлов и каталогам файлов пользователей, созданных администратором системы, Вы можете создавать свои собственные *подкаталоги*.

Подкаталоги используются для того, чтобы:

- Организовать структуру каталогов
- Предотвратить случайное уничтожение или модификацию файлов
- Уменьшить время поиска файлов системой

Файлы обычно группируют по:

- Функции (например, все командные файлы)
- Приложению (например, все файлы для данного проекта)
- Типу (например, все файлы на языке VAX FORTRAN)

Каждый каталог файлов пользователя (UFD) может иметь под собой максимум семь уровней подкаталогов.

- Каждый уровень каталогов имеет идентификатор уровня из 1-39 символов, который составляет часть имени каталога.
- Подкаталоги могут содержать как файлы, так и другие подкаталоги.

Рис. 4-3 иллюстрирует иерархию каталогов.

Рис. 4-3. Иерархия каталогов

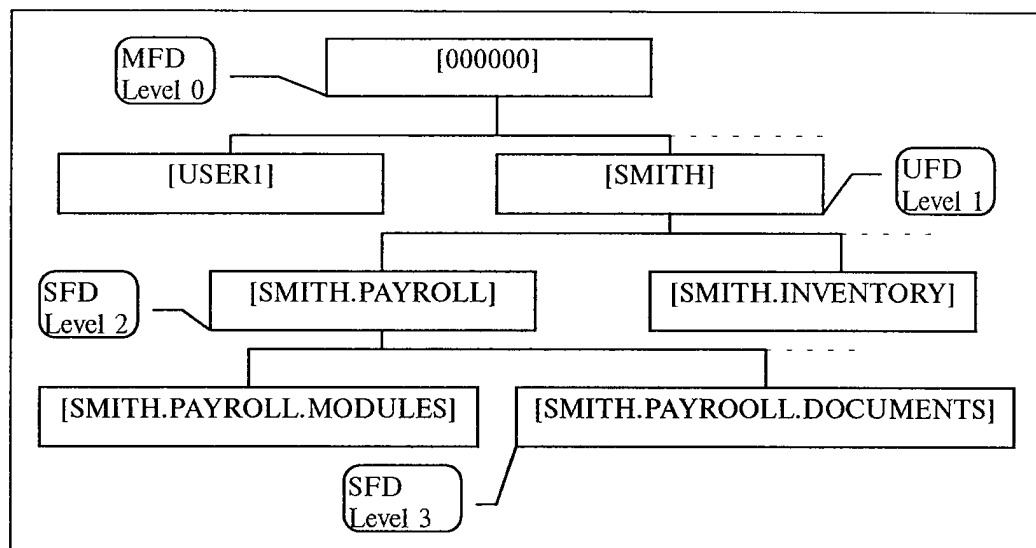


Рис. 4-3, замечания:

"MFD" - это головной каталог файлов, который содержит все каталоги пользователей.

"UFD" - это каталог файлов пользователя.

"SFD" относится к каталогам субфайлов (subfile directories), или подкаталогам:

- USER1 не имеет подкаталогов.
- SMITH имеет два уровня подкаталогов.

4.4.3.1.1. Группирование файлов в подкаталоги

Представьте, что у Вас есть каталог файлов пользователя, такой как показывает Пример 4-1.

Он содержит разнообразные файлы, некоторые из которых имеют отношение друг к другу. Как Вы могли бы сгруппировать файлы для большего удобства?

Пример 4-1. Типичный каталог файлов пользователя

\$ DIRECTORY

Directory \$1\$DUA1:[ROUNDS]

```
EQUIPMENT_REQUEST.07MAR91;1      EXPENSES.BEMIDJI;1 EXPENSES.LONDON;1
EXPENSES.MUNICH;1      EXPENSES.NEW_ORLEANS;1 EXPENSES.NEW_YORK;1
EXPENSE_GUIDELINES.MEMO;1      INVENTORY.DATA;1      LOGIN.COM;1
MAIL$00040090C2C604AD.MAI;1      MAIL$00040090C2C61AE5.MAI;1
MAIL$0004009108C2E5B0.MAI;1      MAIL$000400910A4DA596.MAI;1
MAIL$00040091B0261759.MAI;1      MAIL$00040091B02A1ECB.MAI;1
MAIL$00040091BDA5624C.MAI;1      MAIL$00040091BF32BFA6.MAI;1
MAIL$00040092423D0E72.MAI;1      MAIL$0004009242533668.MAI;1
MAIL$00040092650290B8.MAI;1      MAIL$0004009265293416.MAI;1
MAIL$0004009283820FD2.MAI;1      MAIL$0004009283B5125D.MAI;1
MAIL$0004009289FA0182.MAI;1      MAIL$000400928ACE3AAF.MAI;1
MAIL$00040093A5DC63C0.MAI;1      MAIL$00040093A5DDA697.MAI;1
MAIL$321F446600050094.MAI;1      MAIL$32411AF400050094.MAI;1
MAIL.MAI;1      MINUTES.09JAN91;1      MINUTES.10APR91;1
      MINUTES.13FEB91;1
MINUTES.13MAR91;1      MINUTES.23JAN91;1      MINUTES.24APR91;1
      MINUTES.27FEB91;1
MINUTES.27MAR91;1      MINUTES.TEMPLATE;1
MONTHLY_REPORT.APR91;1 MONTHLY_REPORT.FEB91;1 MONTHLY_REPORT.JAN91;1
MONTHLY_REPORT.TEMPLATE;1      PROJECT_STATUS.FEB;1
PROJECT_STATUS.JAN;1      PROJECT_STATUS.MAR;1
PROJECT_STATUS.TEMPLATE;1      REQUISITION.FORM;1      SUPPLIES.MEMO;1
SUPPLIES.MEMO1;1      SUPPLIES.MEMO2;1      TRAVEL.CHECKLIST;1
```

Total of 52 files.

4.4.3.2. Создание подкаталогов

Чтобы создать подкаталог, используйте команду **CREATE/DIRECTORY**.

Формат:

\$ CREATE/DIRECTORY [каталог.подкаталог]

Синтаксические правила:

- Имя подкаталога должно быть заключено в квадратные скобки.
- Используйте точку (.) для разделения имен подкаталогов.

Каталог или подкаталог сам является файлом:

- Каталог или подкаталог имеет тип файла DIR.
- Номер версии файла типа DIR равен 1.

Пример команды и ответа системы:

```
$ CREATE/DIRECTORY/LOG [ROUNDS.TRAVEL]
%CREATE-I-CREATED, $1$DUA1:[ROUNDS.TRAVEL] created
```

Квалификатор /LOG заставляет систему вывести информационное сообщение, когда подкаталог будет создан.

Разновидность этой команды:

```
$ CREATE/DIRECTORY/LOG [.TRAVEL]
```

```
%CREATE-I-CREATED, $1$DUA1:[ROUNDS.TRAVEL] created
```

Общая ошибка при создании подкаталога - это пропуск точки (.) в спецификации подкаталога:

```
$ CREATE/DIRECTORY/LOG [TRAVEL]
%CREATE-E-DIRNOTCRE, [TRAVEL] directory file not created
-SYSTEM-F-NOPRIV, no privilege for attempted operation
```

Без точки команда представляет собой попытку создать каталог файлов пользователя.

4.4.3.3. Перемещение по структуре каталогов

Команда SET DEFAULT изменяет имя устройства и/или каталога по умолчанию для Вашего текущего процесса.

Формат:

\$ SET DEFAULT имя-устройства:[имя-каталога]

Имя устройства

- Значение для *имя-устройства* не обязательно указывать, если новый каталог по умолчанию находится на том же устройстве.
- Имя физического устройства должно быть завершено двоеточием (:).

Имя каталога

- *Имя-каталога* может включать спецификацию подкаталога.
- Заключите имя каталога или подкаталога в квадратные скобки.

Сокращение

- Используйте дефис (-) для указания каталога, расположенного на один уровень выше текущего каталога по умолчанию.
- Пример команды:

```
$ SET DEFAULT [-]
```

Команда SHOW DEFAULT показывает имена Ваших текущих устройства и каталога по умолчанию. Пример 4-2 показывает, как пользователь перемещается по структуре каталогов.

Пример 4-2. Команды SET DEFAULT и SHOW DEFAULT

```
❶ $ SHOW DEFAULT
$ WORK3 : [ROUNDS]
$
❷ $ SET DEFAULT [ROUNDS.COMMANDS]
$ SHOW DEFAULT
$ WORK3 : [ROUNDS.COMMANDS]
$
❸ $ SET DEFAULT [-]
$ SHOW DEFAULT
$ WORK3 : [ROUNDS]
$
❹ $ SET DEFULT [.COMMANDS]
$ SHOW DEFAULT
$ WORK3 : [ROUNDS.COMMANDS]
$
❺ $ SET DEFAULT WORK7 : [REGNELL.BUILD]
$ SHOW DEFAULT
$ WORK7 : [REGNELL.BUILD]
$
```

```
⑥ $ SET DEFAULT [-.MEMOS]
$ SHOW DEFAULT
WORK7: [REGNELL.MEMOS]
```

Пример 4-2, замечания:

- ① Показать начальное устройство и каталог по умолчанию.
- ② Установить каталог по умолчанию COMMANDS. Устройство по умолчанию не изменяется.
- ③ Используйте дефис для установки каталога по умолчанию на один уровень выше текущего каталога по умолчанию. Устройство по умолчанию не изменяется
- ④ Вновь установить каталог по умолчанию COMMANDS, на этот раз с использованием сокращения.

Точка (.) указывает, что это подкаталог.

Так как вышестоящий уровень подкаталога не указан, предполагается, что это текущий каталог по умолчанию.

- ⑤ Изменить текущее устройство и каталог по умолчанию.
- ⑥ Использовать дефис для изменения каталога по умолчанию и показать новое значение по умолчанию.

4.4.3.3.1. Ошибки при перемещении по структуре каталогов

Вы можете сделать своим каталогом по умолчанию несуществующий каталог. Вы не получите сообщения об ошибке, пока не попытаетесь получить доступ к этому каталогу.

Пример 4-3. Установка несуществующего каталога в качестве каталога по умолчанию

```
① $ SET DEFAULT [NONEXISTENT]
② $ SHOW DEFAULT
WORK3: [NONEXISTENT]
%DCL-I-INVDEF, WORK3:[NONEXISTENT] does not exist
③ $ DIRECTORY
%DIRECT-E-OPENIN, error opening WORK3:[NONEXISTENT]*.*;* as input
-RMS-E-DNF, directory not found
-SYSTEM-W-NOSUCHFILE, no such file
$
```

Пример 4-3, замечания:

- ① Установка каталогом по умолчанию несуществующего каталога. Ошибки не возникает.
- ② Команда SHOW DEFAULT показывает новый каталог по умолчанию и дает информационное сообщение, но не ошибку.
- ③ Когда пользователь пытается получить доступ к несуществующему каталогу, возникает ошибка.

4.4.4. Просмотр информации о файлах

Используйте команду DIRECTORY для нахождения конкретного файла или для получения информации о файлах.

Формат:

\$ DIRECTORY [спец-файла[,...]]

Спецификация файла определяет, какие файлы должны быть перечислены.

Если Вы опустите файловую спецификацию, вы получите каталоговую информацию о всех файлах в Вашем текущем каталоге по умолчанию.

Если Вы указываете более одного файла, разделяйте спецификации файлов запятыми, либо знаками плюс.

Вы можете использовать шаблонные символы в полях спецификации каталога, имени файла, типа файла или номера версии файловой спецификации для получения списка всех файлов, которые удовлетворяют указанным Вами полям.

4.4.4.1. Управление просмотром каталога

Вы можете затребовать различные виды каталоговой информации с помощью добавления квалификаторов к команде DIRECTORY. Таблица 4-5 перечисляет некоторые из наиболее часто используемых квалификаторов.

Таблица 4-5. Часто используемые квалификаторы команды DIRECTORY

Квалификатор	Описание
/COLUMNS=n	Дает список файлов с использованием указанного числа колонок на каждой строке дисплея. По умолчанию в кратком формате четыре колонки.
/DATE	Включает дату создания каждого указанного файла.
/EXCLUDE=(спец-файла[,...])	Любые файлы, которые соответствуют перечисленным файловым спецификациям, исключаются из операции DIRECTORY. Если Вы указываете только одну спецификацию файла, скобки можно опустить. В файловых спецификациях поддерживаются шаблонные символы.
/FULL	Перечисляет следующие предметы для каждого файла: Имя файла, тип и номер версии Идентификационный номер файла (FID) Количество использованных и выделенных блоков Код идентификации владельца файла Даты создания, последнего сохранения, последней модификации, истечения срока хранения Организация файла и другие файловые атрибуты Информация о журнализации Информация о защите
/OUTPUT=спец-файла	Указывает, куда будет направлена выходная информация команды
/OWNER	Показывает код идентификации пользователя (UIC) владельца файла
/PROTECTION	Показывает код защиты файла для каждого файла
/SIZE	Показывает размер файла в использованных блоках

4.4.4.2. Информация, выводимая командой DIRECTORY

Пример 4-1 показывает информацию, выводимую командой DIRECTORY/FULL. Все другие квалификаторы запрашивают подмножество этой информации.

Пример 4-4. Выходная информация команды DIRECTORY/FULL

```
$ DIRECTORY/FULL TERMINAL.COM
```

```
Directory WORK3:[ROUNDS.COMMANDS]
```

```
TERMINAL.COM;37                               File ID: (2156,11,0)
Size:                2/3                       Owner:    [GROUP11,ROUNDS]
Created:  31-MAY-1991 10:42:06.08
Revised:  31-MAY-1991 10:42:06.49 (2)
Expires:   <None specified>
Backup:    31-MAY-1991 19:26:42.85
File organization: Sequential
File attributes:  Allocation: 3, Extend: 0, Global buffer count: 0
                  Version limit: 2
Record format:    Variable length, maximum 69 bytes
Record attributes: Carriage return carriage control RMS attributes:  None
Journaling enabled: None
File protection:  System:RWED, Owner:RWED, Group:RE, World:
Access Cntrl List: None
Total of 1 file, 2/3 blocks.
```

4.4.4.3. Использование шаблонов и сокращений в команде DIRECTORY

Вы можете использовать шаблоны для указания файлов команде DIRECTORY. Используйте дефис (-) для указания каталога на один уровень выше текущего каталога по умолчанию. Используйте три точки (...) для ссылки на все уровни подкаталогов ниже указанного каталога. Таблица 4-6 дает несколько примеров использования шаблонов и сокращений в спецификациях файлов для команды DIRECTORY.

Таблица 4-6. Использование шаблонов и сокращений с командой DIRECTORY

Команда	Описание полученного списка файлов
\$ DIRECTORY PAY.FOR;*	Все версии файла PAY.FOR
\$ DIRECTORY *.DIR	Все файлы с типом DIR (подкаталоги текущего каталога по умолчанию)
\$ DIRECTORY *.*;*	Все файлы, типы и версии (эквивалентно вводу просто DIRECTORY)
\$ DIRECTORY PAY%.FOR;*	Все версии файлов с типом FOR и именами, начинающимися с символов PAY, за которыми следует ровно один символ
\$ DIRECTORY PAY*.*;*	Все файлы, чьи имена начинаются с PAY, включая все типы файлов и все версии
\$ DIRECTORY [...]	Все файлы в текущем каталоге и во всех подкаталогах ниже него
\$ DIRECTORY [MOREAU...]*PAY*	Все файлы, имеющие в своем имени символы PAY, независимо от типа файла, содержащиеся в каталоге [MOREAU] и всех его подкаталогах
\$ DIRECTORY [-]	Все файлы в каталоге на один уровень выше текущего каталога по умолчанию

4.5. Создание и редактирование файлов

Одной из наиболее распространенных задач для пользователя является создание и модификация текстовых файлов. Текстовые файлы могут принимать различные формы и служить многим целям. Они могут быть:

- Записками и письмами
- Файлами данных, используемыми другими программами и утилитами
- Компьютерными программами, написанными на таком языке, как, например, VAX FORTRAN

Программное обеспечение VMS предоставляет несколько способов создания, обслуживания и модификации текстовых файлов. Для создания текстовых файлов Вы можете использовать:

- Текстовый редактор
- Команду CREATE
- Квалификатор /OUTPUT во многих командах DCL

Двумя наиболее популярными текстовыми редакторами являются:

- EVE (Extensible VAX Editor - расширяемый редактор для VAX), рассматриваемый в этом разделе
- EDT, рассматриваемый в приложении

4.5.1. EVE, расширяемый редактор для VAX

Вы можете использовать EVE, расширяемый редактор для VAX, для редактирования текста в новых и существующих файлах.

Возможности EVE включают:

- Два режима ввода текста
 - Режим вставки (текст передвигается вправо по мере вставки символов)
 - Режим замены (вводимый текст заменяет существующий текст)
- Возможность получения вспомогательной информации в диалоговом режиме
- Средство журнализации, позволяющее Вам восстановить свою работу в случае неожиданного прерывания сеанса редактирования.
- Многооконность, позволяющая Вам просматривать содержимое одновременно нескольких файлов
- Автоматический переход на новую строку при достижении правой границы текста
- Ввод командной строки
- Возможность настройки на нужды конкретного пользователя
- Возможность использования на терминалах серий VT100, VT200, VT300 и более поздних, а также на рабочих станциях, работающих под управлением системы OpenVMS.
- Предоставление по желанию дополнительной клавиатуры в стиле EDT
- Редактор создан на основе утилиты обработки текстов TPU, включающей в себя мощный язык программирования, позволяющий пользователям создавать собственные текстовые редакторы либо встраивать в EVE новые команды.

ЗАМЕЧАНИЕ: EVE нельзя использовать на печатающих терминалах.

4.5.2. Вызов EVE

Для вызова EVE используйте команду EDIT/TPU.

Формат:

\$ EDIT/TPU имя-файла.тип

Эта команда может быть использована для создания нового файла и для редактирования существующего файла.

4.5.2.1. Создание нового файла с помощью EVE

Существует два способа создания нового файла с помощью EVE:

Один способ - это ввести имя файла при создании:

\$ EDIT/TPU имя-файла.тип

Другой способ - оставить имя файла незаполненным, тогда при окончании работы с файлом Вас попросят ввести имя файла:

\$ EDIT/TPU

·
·
·

Type filename for buffer MAIN (press RETURN to not write it): имя-файла.тип

4.5.3. Формат экрана EVE

Рис. 4-2 показывает части экрана EVE.

Рис. 4-4. Формат экрана EVE

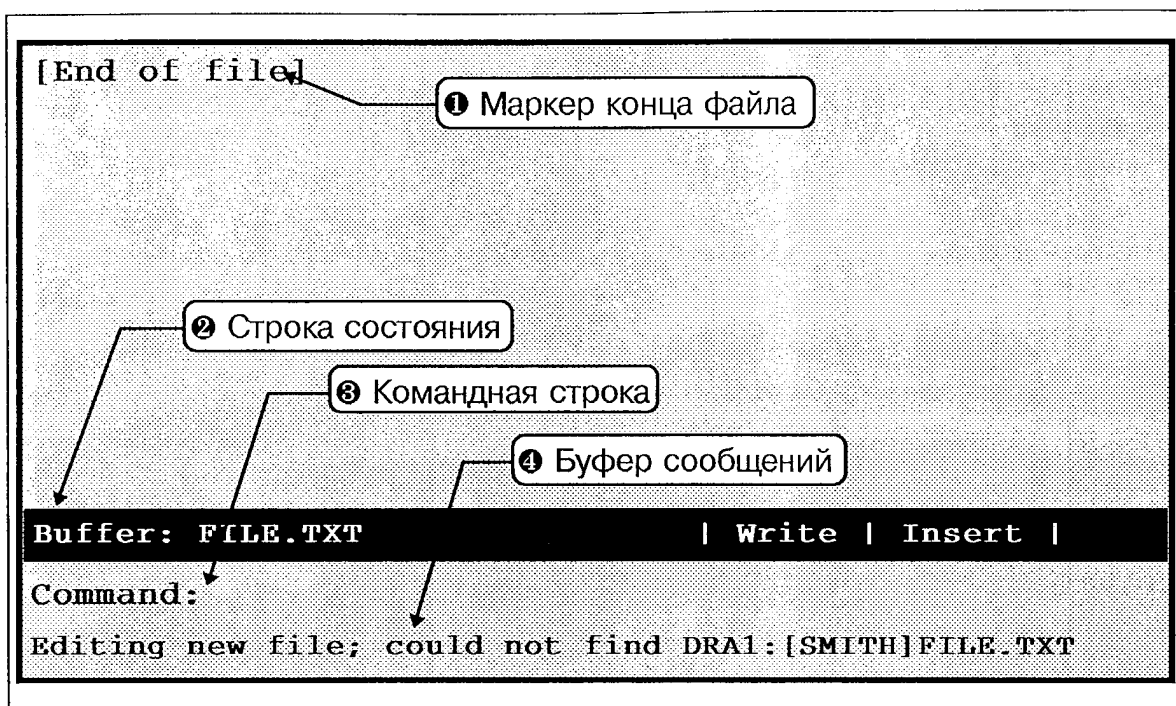


Рис. 4-4, замечания:

- ❶ Маркер *[End of file]* указывает конец файла.
- ❷ Строка состояния показывает:
 - Имя буфера, который Вы в данный момент редактируете
 - Режим ввода текста (на рисунке режим вставки - *Insert*)
 - Направление курсора (на рисунке направление вперед - *Forward*)
- ❸ Командная строка используется для ввода команд строкового режима. Чтобы ввести команду строкового режима, нажмите DO (PF4 на терминалах VT100) для получения подсказки *Command:*.
- ❹ Буфер сообщений хранит сообщения, которые Вы получаете во время сеанса редактирования.

4.5.4. Ввод команд EVE

Существует два способа ввода команд EVE:

- Путем нажатия predetermined функциональных клавиш
- Путем набора команд

4.5.4.1. Использование predetermined клавиш для ввода команд EVE

EVE определяет некоторые редактирующие клавиши по умолчанию.

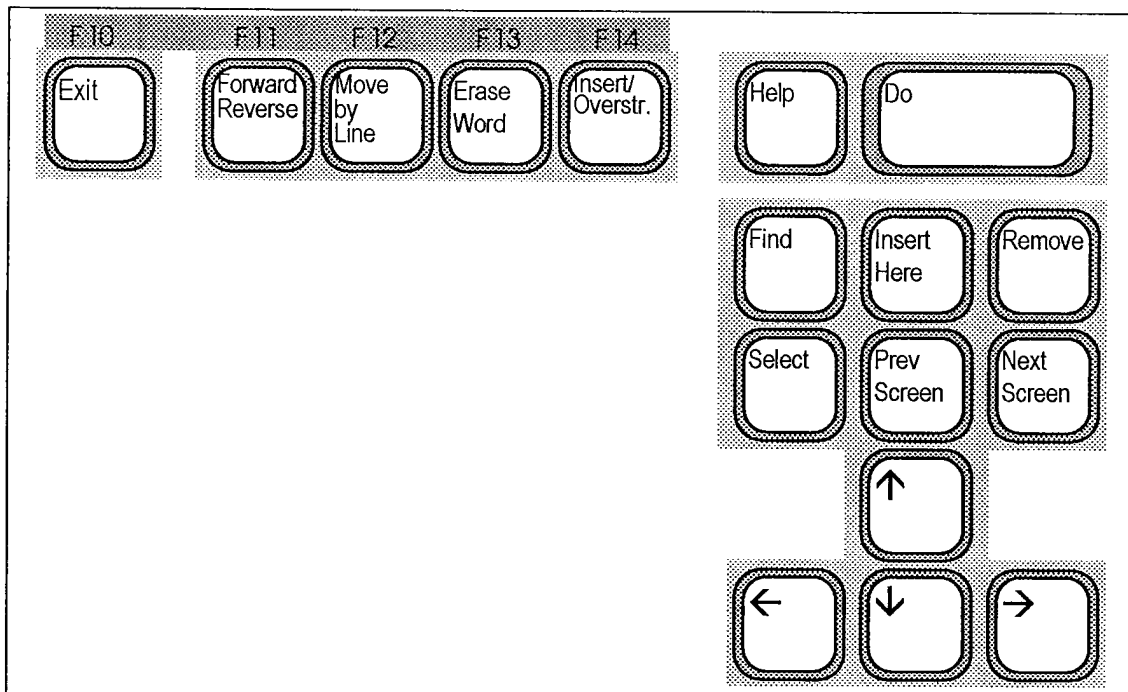
Предetermined функциональные клавиши на терминалах серий VT200 и VT300 включают малую дополнительную клавиатуру, некоторые функциональные клавиши и некоторые управляющие комбинации клавиш.

На терминалах серии VT100 EVE автоматически определяет большинство клавиш дополнительной цифровой клавиатуры, клавиши со стрелками и некоторые управляющие клавиши.

Каждая предопределенная функциональная клавиша выполняет одну команду редактирования.

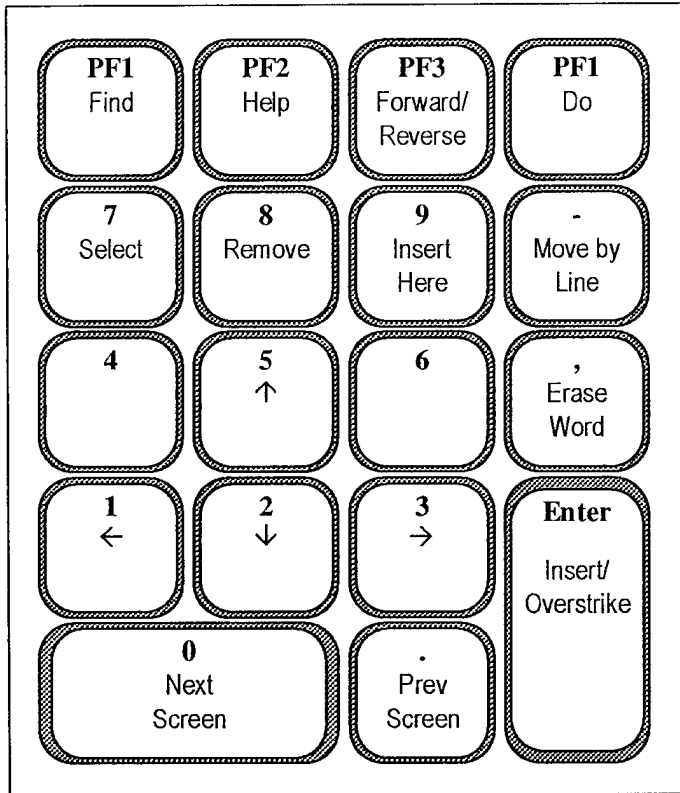
Рис. 4-5 показывает определения клавиш дополнительной клавиатуры EVE для терминалов серии VT200.

Рис. 4-5. Определения клавиш дополнительной клавиатуры EVE (Терминалы серии VT200)



Определения клавиш цифровой дополнительной клавиатуры EVE на терминалах серии VT100 показаны на Рис. 4-6

Рис. 4-6. Определения клавиш дополнительной клавиатуры EVE (Терминалы серии VT100)



4.5.4.2. Набор команд EVE

Кроме использования определенных клавиш для ввода команд EVE, Вы можете набирать команды. Набор команды включает в себя три шага:

1. Нажмите клавишу DO (терминалы серий VT300 и VT200) или клавишу PF4 (терминалы серии VT100). Курсор переместится в окно команд.
2. У подсказки *Command:* наберите команду, которую хотите выполнить. Вы можете сокращать команды.
3. Нажмите либо клавишу DO, либо клавишу RETURN. EVE выполнит команду или подскажет Вам о необходимости ввода дополнительной информации.

Простые команды, которые могут Вам пригодиться, описаны далее в этой главе.

4.5.5. Ввод текста

Следуйте данному образцу для ввода текста в файл:

- Поместите курсор в рабочей области туда, куда Вы хотите ввести текст.
- Наберите символы на клавиатуре.
- Курсор отмечает текущую точку вставки.
- Используйте клавишу DELETE для исправления ошибок.

В EVE существует два режима ввода текста:

- Вставка
 - ↳ Каждый набираемый символ вставляется слева от точки вставки.
 - ↳ Существующий текст сдвигается вправо.
 - ↳ DELETE удаляет символ слева от курсора и передвигает влево весь текст, находившийся справа от удаленного символа.
 - ↳ Этот режим установлен по умолчанию.
- Замена
 - ↳ Каждый набираемый Вами символ пишется поверх существующего текста.
 - ↳ DELETE заменяет существующие символы пробелами.

Функциональная клавиша F14 (или CTRL/A) меняет один режим на другой.

4.5.6. Замена текста

Команда **REPLACE** позволяет Вам заменять одну последовательность символов на другую.

Для замены последовательности символов:

1. Нажмите клавишу DO:
2. Введите команду REPLACE, указывая старую и новую последовательности.

Command: REPLACE старая_последовательность новая_последовательность

EVE также подсказывает Вам ввод последовательностей символов:

1. Нажмите клавишу DO.
2. Введите команду REPLACE и нажмите RETURN.
3. Наберите последовательность, которую надо заменить, после подсказки *Old string:*, затем нажмите RETURN.

Old string: старый_текст

4. Наберите новую последовательность символов у подсказки *New string:* и нажмите RETURN.

New String: новый_текст

5. EVE передвигает курсор к первому появлению заменяемой последовательности и выдает подсказку:

Replace? Type Yes, No, All, Last or Quit?

6. Когда замена завершена, EVE показывает сообщение следующего формата:

Replaced 2 occurrences.

EVE предлагает Вам пять вариантов при замене последовательностей символов. Их описывает Таблица 4-7.

Таблица 4-7. Ответ на подсказки REPLACE

Ответ	Действия EVE
YES	Заменяет последовательность символов и пытается найти следующее появление этой последовательности в текущем направлении.
NO	Не заменяет последовательность символов и пытается найти следующее появление этой последовательности в текущем направлении.

<i>Ответ</i>	<i>Действия EVE</i>
ALL	Заменяет последовательность символов и все другие появления этой последовательности в текущем направлении. Курсор перемещается в позицию, в которой произошла последняя замена.
LAST	Заменяет данное появление строки и останавливает процедуру REPLACE. Курсор не передвигается.
QUIT	Не заменяет данное появление строки и останавливает процедуру REPLACE. Курсор не передвигается.

4.5.7. Передвижение по документу

Существует два способа перемещения курсора по документу в EVE:

- С использованием клавиш перемещения курсора, которые описывает Таблица 4-8.
- С использованием команд перемещения курсора, вводимых после нажатия клавиши DO. Их описывает Таблица 4-9.

Таблица 4-8. Клавиши перемещения курсора EVE

<i>Клавиша</i>	<i>Куда перемещается курсор</i>
↑	Перемещает курсор на одну строку вверх
↓	Перемещает курсор на одну строку вниз
←	Перемещает курсор на один символ влево
→	Перемещает курсор на один символ вправо
CTRL/E	Перемещает курсор в конец текущей строки
CTRL/H	Перемещает курсор в начало текущей строки
Prev Screen	Перемещает курсор в предыдущий экран текущего буфера
Next Screen	Перемещает курсор в следующий экран текущего буфера

Таблица 4-9. Простые команды перемещения курсора

<i>Команда</i>	<i>Определение</i>	<i>Пример</i>
Bottom	Перемещает курсор в конец файла	Command: BOTTOM
Top	Перемещает курсор в начало файла	Command: TOP
Move by word	Перемещает курсор на одно слово в текущем направлении	Command: MOVE BY WORD
Move by page	Перемещает курсор на одну страницу в текущем направлении	Command: MOVE BY PAGE
Line n	Перемещает курсор в начало указанной строки (n - это номер строки)	Command: LINE 5
What line	EVE укажет текущую строку	Command: WHAT LINE

4.5.8. Окончание сеанса редактирования и сохранение файла

Существует два способа окончания сеанса редактирования EVE.

Вы можете:

- ↳ ПОКИНУТЬ (QUIT) сеанс, чтобы отменить сделанные Вами в тексте изменения
- ↳ ВЫЙТИ (EXIT) из сеанса, чтобы записать изменения в файл

Чтобы ПОКИНУТЬ сеанс без сохранения изменений:

Нажмите DO (VT200) или PF4 (VT100) и наберите QUIT после подсказки.

Чтобы ВЫЙТИ и сохранить изменения:

Нажмите CTRL/Z.

или

Нажмите DO (VT200) или PF4 (VT100) и наберите EXIT после командной подсказки

или

Нажмите функциональную клавишу F10.

4.5.8.1. Восстановление файлов

Если во время редактирования файла прервалась работа системы или Вы нажали CTRL/Y, Вы можете восстановить свой сеанс.

EVE записывает нажатые Вами клавиши в журнальный файл с типом TJI или TPU\$JOURNAL.

- Обычно сеанс редактирования заканчивается нормально и система удаляет журнальный файл.
- Если произошло прерывание, журнальный файл сохраняется.
- Могут не сохраниться самые последние нажатые Вами клавиши.

Вы должны восстанавливать сеанс редактирования за терминалом того же типа, который был использован во время восстанавливаемого сеанса редактирования.

Формат:

\$ EDIT/TPU/RECOVER первоначальный-файл

4.5.9. Получение вспомогательной информации

EVE предоставляет диалоговую помощь по:

- Элементам меню
- Командам
- Функциям дополнительной клавиатуры

Чтобы запросить вспомогательную информацию, нажмите DO, наберите HELP, и выберите один из следующих элементов меню Help:

- Список предметов
 - ↳ Показывает список всех команд EVE и других предметов
 - ↳ Наберите имя предмета по которому Вы хотите вспомогательную информацию
- Дополнительная клавиатура
 - ↳ Рисует диаграмму дополнительной клавиатуры
 - ↳ Нажмите любую клавишу, по которой Вы хотите получить вспомогательную информацию

- Клавиши
 - ↳ Перечисляет все текущие определения клавиш
 - ↳ Нажмите любую клавишу, по которой Вы хотите получить вспомогательную информацию

Вы можете также получить вспомогательную информацию, набрав команду Help и указав предмет.

- Большинство предметов имеют список связанных команд, который Вы можете просмотреть
- Чтобы увидеть список всех предметов, по которым возможна вспомогательная информация, наберите вопросительный знак
- Чтобы выйти из режима получения вспомогательной информации, нажмите клавишу RETURN.

4.5.10. Некоторые расширенные возможности EVE

4.5.10.1. Переключение дополнительной клавиатуры в режим EDT

Команда **SET KEYPAD EDT** определяет дополнительную клавиатуру в стиле EDT, но не полностью исполняющую функции EDT. Существуют отличия:

- CTRL/Z заставляет EVE записывать буфер в файл и выходить к подсказке DCL.
- Когда дополнительная клавиатура установлена в режиме EDT, клавиша GOLD/KP7 определена как клавиша DO.
- Клавиша GOLD/KP8 определена как FILL для переформатирования выделенного текста или текущего абзаца. Если Вы хотите, чтобы эта клавиша заполняла только выделенный текст (как в настоящем режиме EDT), переопределите эту клавишу как FILL RANGE.
- EVE определяет клавишу ENTER как RETURN.

4.5.10.2. Включение в редактируемый файл существующего файла

Чтобы включить содержимое другого файла в файл, который Вы сейчас редактируете:

- Поместите курсор туда, куда Вы хотите вставить файл.
- Нажмите клавишу DO.
- Наберите **INCLUDE спец-файла** и нажмите RETURN.

4.5.10.3. Выполнение команд DCL из EVE

Когда Вы редактируете файл, у Вас может возникнуть желание выполнить команду DCL. Например, Вам может понадобиться увидеть каталог Ваших файлов или узнать текущий каталог по умолчанию. Вы можете сделать это, не выходя из EVE.

1. Нажмите DO и наберите команду DCL в ответ на командную подсказку, например:

Command: DCL DIRECTORY

2. Любая выходная информация команды DCL будет показана в нижней части экрана.
3. Для удаления изображения этой выходной информации с экрана нажмите DO и наберите ONE WINDOW.

4.5.11. Приложение: текстовый редактор EDT

Редактор EDT - это стандартный текстовый редактор, поставляемый с системой OpenVMS.

- EDT позволяет редактировать на печатающем терминале.
- EDT предоставляет два режима интерактивного редактирования:
 - ↳ Строковый режим
 - ↳ Режим с использованием дополнительной клавиатуры

4.5.11.1. Вызов EDT

Для вызова EDT введите команду:

\$ EDIT/EDT1 имя-файла

4.5.11.2. Режимы редактирования EDT

- Строковый режим в основном предназначен для работы с печатающими терминалами.
 - ↳ Автоматически включается при входе в EDT
 - ↳ Индицируется подсказкой в виде звездочки (*)
 - ↳ Работает с файлом по принципу "строка за строкой"
- Режим с дополнительной клавиатурой требует видеотерминал
 - ↳ Включается командой **CHANGE** у подсказки строкового режима (*)
 - ↳ Единицей работы является файл
 - ↳ Изменения, сделанные на экране, становятся изменениями в файле

4.5.11.3. Строковый режим

По умолчанию, Вы находитесь в строковом режиме, когда запускаете редактор EDT. Таблица 4-10 описывает команды строкового режима EDT.

Таблица 4-10. Часто используемые команды строкового режима EDT

Строковая команда	Функция
CHANGE	Изменяет строковый режим на режим с дополнительной клавиатурой
COPY	Копирует указанный текст, помещая его над указанной строкой
DELETE	Удаляет указанные строки
EXIT	Заканчивает сеанс редактирования
FIND	Перемещает EDT в новую позицию в текущем буфере
HELP	Дает доступ к вспомогательной информации HELP
INCLUDE	Копирует внешний файл в текстовый буфер EDT
INSERT	Добавляет текст к буферу
MOVE	Передвигает строки с одного места в другое
QUIT	Заканчивает сеанс редактирования без копирования текста во

¹ В ранних версиях OpenVMS (до версии 6.0) квалификатор /EDT не требовался, так как EDT был редактором по умолчанию. В настоящее время редактором по умолчанию является EVE.

<i>Строковая команда</i>	<i>Функция</i>
	внешний файл
REPLACE	Удаляет строки в указанном диапазоне
RESEQUENCE	Присваивает строкам новые номера
SET WRAP	Включает автоматический переход на новую строку (заворачивание строк) при вводе нового текста
SHOW BUFFER	Дает список всех доступных буферов
SHOW KEY	Показывает определения всех клавиш, имеющих редактирующие функции в режиме использования дополнительной клавиатуры
SUBSTITUTE	Заменяет одну последовательность символов на другую в указанном диапазоне
TYPE	Показывает строки текста
WRITE	Копирует текст из буфера во внешний файл

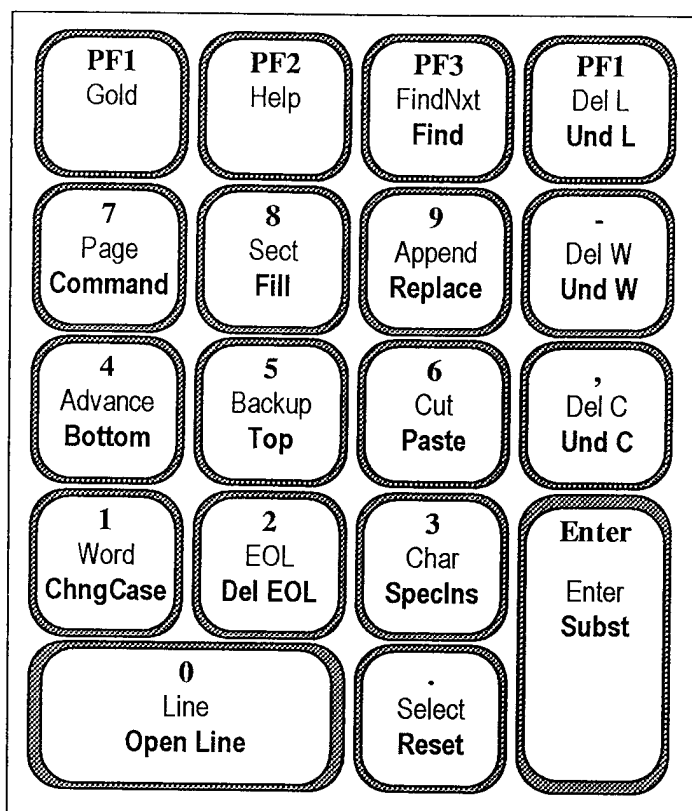
4.5.11.4. Режим с дополнительной клавиатурой

Чтобы войти в режим с дополнительной клавиатурой, когда Вы находитесь в строковом режиме, наберите букву C (сокращение для *change*) и нажмите RETURN.

- Вы, возможно, будете использовать этот режим большую часть времени.
- Теперь Вы можете вводить текст в свой файл и использовать клавиши дополнительной клавиатуры для ввода команд этого режима.
- Для возврата в строковый режим нажмите CTRL/Z.

Рис. 4-7 показывает определения клавиш дополнительной клавиатуры для режима EDT, использующего дополнительную клавиатуру.

Рис. 4-7. Определения клавиш дополнительной клавиатуры EDT



4.5.11.5. Перемещение курсора в EDT

В режиме с дополнительной клавиатурой текст вставляется в текущую позицию курсора. Таблица 4-11 перечисляет способы перемещения курсора.

Таблица 4-11. Перемещение курсора в EDT

Функция	Клавиша
Перейти на один символ в любом направлении	←, →, ↑, ↓
Перейти на начало следующего слова	KP1 (Word)
Перейти на начало следующей строки	KP0 (Line)
Перейти на конец строки	KP2 (EOL)
Перейти на следующую границу страницы (символ <i>form feed</i> - перевод формата)	KP7 (Page)
Перейти на конец буфера	PF1 (Gold), затем KP4 (Bottom)
Перейти к началу буфера	PF1 (Gold), затем KP5 (Top)
Перейти к следующему сегменту текста (16 строк)	KP8 (Sect)
Установить текущее направление курсора вперед	KP4 (Advance)
Установить текущее направление курсора назад	KP5 (Backup)
Только для терминалов VT200, VT300 и более новых, а также рабочих станций	
Перейти к следующему сегменту текста (16 строк вперед)	Next Screen

	на центральной дополнительной клавиатуре
Перейти к предыдущему сегменту текста (16 строк назад)	Prev Screen на центральной дополнительной клавиатуре

4.5.11.6. Завершение сеанса EDT

Для завершения сеанса EDT в строковом режиме:

- Наберите EXIT для окончания сеанса и сохранения изменений.
- Наберите QUIT для окончания сеанса без сохранения изменений.

Существует два способа завершить сеанс EDT, находясь в режиме с дополнительной клавиатурой:

- 1) Нажмите CTRL/Z для возвращения в строковый режим и наберите EXIT или QUIT.
- 2) Нажмите PF1, затем KP7 (GOLD/COMMAND).
У подсказки *Command*: наберите одно из:
 - EXIT для окончания сеанса и сохранения изменений
 - QUIT для окончания сеанса без сохранения изменений

Замечание

В отличие от EVE, EDT не переспрашивает, действительно ли Вы хотите покинуть сеанс без сохранения изменений. Все внесенные Вами изменения будут потеряны.

4.5.11.7. Восстановление файлов

Если сеанс редактирования EDT завершился ненормально, например, из-за сбоя питания или другого неожиданного перерыва в работе, или Вы нажали CTRL/Y, Вы можете восстановить большую часть или все внесенные изменения.

Редактор EDT записывает нажатые Вами клавиши в журнальный файл, пока Вы не выйдете и не сохраните свой текстовый файл.

- Журнальный файл может быть использован для восстановления прерванного сеанса редактирования.
- Несколько последних изменений файла могут быть не восстановлены.

Журнальный файл:

- По умолчанию имя файла то же, что и имя входного файла.
- Тип файла по умолчанию JOU.

Для восстановления с использованием журнального файла введите следующую команду:

\$ EDIT/RECOVER имя-файла.тип

Укажите первоначальный тип файла (не JOU).

Процедура восстановления воссоздаст все изменения, сделанные во время прерванного сеанса.

4.6. Работа с файлами

Во время терминальной сессии пользователю часто бывает необходимо манипулировать файлами. Существует несколько общеупотребительных команд DCL, которые должен уметь применять каждый пользователь.

Эта глава обсуждает использование команд COPY, PRINT, RENAME, DELETE, и PURGE.

4.6.1. Копирование файлов

Для копирования одного или нескольких существующих файлов в новый файл используйте команду COPY.

Команда COPY может выполнять следующее:

- Копировать входной файл в выходной файл
- Объединять два или более входных файлов в один выходной файл
- Копировать группу входных файлов в группу выходных файлов

Формат:

\$ COPY входной-файл[,...] выходной-файл

Входные файлы не изменяются.

Выходные файлы содержат тот же текст или те же данные, что и входные файлы.

Если Вы пропустите входную или выходную спецификации файлов, Вас попросят их ввести:

```
$ COPY
 _From: входной-файл
 _To: выходной-файл
```

Таблица 4-12 показывает некоторые примеры использования команды COPY.

Таблица 4-12. Использование команды COPY

Операция	Команда и комментарии
Скопировать входной файл в выходной файл.	Скопировать один файл в другой в том же самом каталоге. \$ COPY TEMPLATE.REPORT APRIL.REPORT В следующем примере выходной файл будет помещен в подкаталог текущего каталога по умолчанию. \$ COPY MARCH.REPORT - _ \$ [.REPORTS] MARCH_91.STATUS
Объединить два или более входных файлов в один выходной файл	Второй файл добавляется к первому файлу и результат помещается в третий файл. Входные файлы не изменяются. \$ COPY DEPT1.DIS,DEPT2.DIS - _ \$ MASTER.DIS
Скопировать группу входных файлов в группу выходных файлов с использованием шаблонов.	Все файлы в текущем каталоге, имеющие тип файла TXT, копируются в указанный каталог, с сохранением тех же имен и типов файлов. \$ COPY *.TXT;* [.NEWBOOK] *

Операция	Команда и комментарии
Скопировать файл с устройства, отличного от Вашего устройства по умолчанию.	Скопировать SAMPLE.TXT с указанных устройства и каталога на Ваши устройство и каталог по умолчанию, с сохранением того же имени и типа файла. \$ COPY DJA0:[SOURCE]SAMPLE.TXT *
Скопировать файл с другой компьютерной системы	Скопировать SAMPLE.TXT с указанных узла, устройства и каталога на Ваши устройство и каталог по умолчанию, с сохранением того же имени и типа файла. \$ COPY - _ \$ NEAT: :DJA0:[SOURCE]SAMPLE.TXT *

4.6.1.1. Изменение действия команды COPY

Вы можете добавлять квалификаторы к команде COPY для изменения способа выполнения ею требуемой операции.

Таблица 4-13 описывает некоторые наиболее употребительные квалификаторы команды COPY.

Таблица 4-13. Широко используемые квалификаторы команды COPY

Квалификатор	Описание
/CONFIRM /NOCONFIRM (по умолчанию)	Определяет, будет ли перед каждой операцией копирования запрашиваться подтверждение пользователя на выполнение операции над данным файлом.
/EXCLUDE=(спец-файла[,...])	Исключает указанные файлы из операции копирования.
/LOG /NOLOG (по умолчанию)	Определяет, будет ли команда COPY показывать файловую спецификацию для каждого копируемого файла.
/PROTECTION=(x)	Указывает защиту выходного файла.

Пример 4-5 показывает использование квалификаторов /CONFIRM и /LOG при копировании файлов. Это особенно полезно при использовании шаблонов в операциях копирования.

Пример 4-5. Копирование файлов

```

❶ $ COPY/CONFIRM/LOG *.DAT [ .DATA]*.*
❷ COPY WORK3:[ROUNDS]HISTORY.DAT;1 to WORK3:[ROUNDS.DATA]HISTORY.DAT;1 ? [
  N]: N
❸ COPY WORK3:[ROUNDS]INVENTORY.DAT;1 to WORK3:[ROUNDS.DATA]INVENTORY.DAT;1
  ? [N]: Y
❹ %COPY-S-COPIED, WORK3:[ROUNDS]INVENTORY.DAT;1 copied to WORK3:[ROUNDS.DA
  TA]INVENTORY.DAT;1 (7 blocks)
  COPY WORK3:[ROUNDS]PROJECT.DAT;1 to WORK3:[ROUNDS.DATA]PROJECT.DAT;1 ? [
  N]: Y
  %COPY-S-COPIED, WORK3:[ROUNDS]PROJECT.DAT;1 copied to WORK3:[ROUNDS.DATA
  ]PROJECT.DAT;1 (10 blocks)
❺ %COPY-S-NEWFILES, 2 files created
$

```

Пример 4-5, замечания:

- ❶ Введите команду для копирования всех файлов, имеющих тип файла DAT в подкаталог DATA. Укажите квалификаторы /CONFIRM и /LOG.
- ❷ Это запрос о подтверждении команды COPY. Наберите N или нажмите RETURN для указания, что данный файл не надо копировать.
- ❸ Наберите Y, чтобы подтвердить копирование файла.
- ❹ Это протокольное сообщение показывает, что файл скопирован.
- ❺ Это итоговое протокольное сообщение, показывающее, что все файлы, соответствующие спецификации, обработаны.

4.6.1.2. Типичные ошибки при копировании файлов

Ошибки при копировании файлов обычно связаны с открытием входного или выходного файла. Проверьте текст сообщения об ошибке, чтобы определить, какого вида ошибка произошла.

Пример 4-6. Типичные ошибки при копировании файлов

- ```
$ COPY SAMPLE.COM [.COMMANDS]*.*
❶ %COPY-E-OPENIN, error opening WORK3:[ROUNDS]SAMPLE.COM; as input
-RMS-E-FNF, file not found
$
$ COPY TEST.LOG [BUNNELL]*.*
❷ %COPY-E-OPENOUT, error opening WORK3:[BUNNELL]TEST.LOG;1 as output
-RMS-E-PRV, insufficient privilege or file protection violation
%COPY-W-NOTCOPIED, WORK3:[ROUNDS]TEST.LOG;1 not copied
$
```

##### Пример 4-6, замечания

- ❶ Это ошибка в спецификации входного файла. В данном случае входной файл не найден. Входной файл может быть также защищен от Вас.
- ❷ Это ошибка в спецификации выходного файла. Выходной каталог защищен от Вас.

#### 4.6.2. Просмотр содержимого файлов

Для вывода содержания файла на Ваш терминал используйте команду TYPE .

Формат:

##### \$ TYPE параметр

Указание параметра:

- Параметр состоит из одной или более спецификаций файлов, разделенных запятыми.
- Нет необходимости указывать все части спецификации файла, так как команда TYPE обеспечивает значения по умолчанию.

Просмотр нескольких файлов:

- Если больше одного файла перечислено в команде TYPE, файлы выводятся в указанном порядке.
- Если используются шаблоны, то файлы указаны в алфавитном порядке.

Управление просмотром:

- Для временной приостановки движения текста на экране используйте клавишу HOLD SCREEN на терминалах серий VT200, VT300 и более новых или комбинацию CTRL/S и CTRL/Q на более старых терминалах.
- Для поэкранного просмотра введите TYPE/PAGE.

Таблица 4-14 показывает используемые командой TYPE значения по умолчанию для спецификаций файлов.

**Таблица 4-14. Значения по умолчанию для спецификаций файлов команды TYPE**

| Часть спецификации файла | Значение по умолчанию           |
|--------------------------|---------------------------------|
| Устройство               | Текущее устройство по умолчанию |
| Каталог                  | Текущий каталог по умолчанию    |
| Имя файла                | Нет значения по умолчанию       |
| Тип файла                | LIS                             |
| Версия                   | Наибольший номер версии         |

Пример 4-7 показывает использование команды TYPE для просмотра содержимого файла.

### Пример 4-7. Использование команды TYPE

```
$ TYPE DEPT2.DIS
NEAT: :BUNNELL
TCHR: :MOREAU
TCHR: :PICCADACI
TIDY: :REGNELL
TIDY: :ROUNDS
$
```

### 4.6.3. Печать файлов

Команда PRINT посылает содержимое файлов на устройство вывода.

Формат:

#### \$ PRINT параметр

Параметр - это список из одной или более спецификаций файлов, разделенных запятыми.

Команда PRINT имеет те же значения по умолчанию для спецификаций файлов, что и команда TYPE.

Когда Вы введете команду PRINT, Вы увидите сообщение, похожее на то, которое показывает Пример 4-8

### Пример 4-8. Ввод команды PRINT

```
$ PRINT MOD5.TXT
❶ Job MOD5 (queue SYS$PRINT, entry 776) started on LPA0
$
$ PRINT MOD6.TXT
❷ Job MOD6 (queue SYS$PRINT, entry 777) pending
 pending status caused by queue busy
```

#### Пример 4-8, замечания

- ❶ Файл начал печататься.  
Заметьте, что система назначила запросу номер элемента очереди.
- ❷ Это сообщение появляется, если файл не может быть напечатан в данный момент.

#### 4.6.3.1. Указание характеристик задания на печать

Вы можете указывать характеристики Вашего запроса на печать, добавляя квалификаторы к команде PRINT. Некоторые из наиболее распространенных квалификаторов показывает Таблица 4-15

**Таблица 4-15. Указание характеристик задания на печать**

| Квалификатор                                    | Комментарий и пример(ы)                                                                                                                                                                                                                              |
|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /COPIES=n                                       | Количество копий<br><br>Напечатать две копии файла MEMO.TXT:<br>\$ PRINT/COPIES=2 MEMO.TXT<br>Напечатать две копии Q1.TXT и три копии Q2.TXT:<br>\$ PRINT Q1.TXT/COPIES=2,Q2.TXT/COPIES=3                                                            |
| /JOB_COUNT=n                                    | Показывает, сколько раз должно быть напечатано полностью все задание<br><br>Запросить трехкратную печать набора из двух файлов:<br>\$ PRINT/JOB_COUNT=3 DEPT1.DIS,DEPT2.DIS                                                                          |
| /AFTER=время                                    | Время, когда задание может быть напечатано<br><br>Напечатать Q1_REPORT.TXT сегодня после 6 вечера:<br>\$ PRINT/AFTER=18:00 Q1_REPORT.TXT                                                                                                             |
| /PAGES=(<br>[нижний-предел,<br>верхний-предел]) | Количество страниц, которые должны быть напечатаны<br><br>Напечатать страницы с 6 по 8 включительно файла Q1_REPORT.TXT:<br>\$ PRINT/PAGES=(6,8) Q1_REPORT.TXT<br><br>Напечатать с 6 страницы до конца файла:<br>\$ PRINT/PAGES=(6,"") Q1_REPORT.TXT |
| /[NO]NOTIFY                                     | Должна ли система известить Вас о завершении или сбросе задания<br><br>Попросить об извещении, когда Q1_REPORT.TXT будет напечатан:<br>\$ PRINT/NOTIFY Q1_REPORT.TXT                                                                                 |

#### **4.6.3.2. Просмотр заданий в очереди на печать**

Вы можете использовать команду SHOW ENTRY или команду SHOW QUEUE для просмотра состояния заданий в очереди на печать.

##### **4.6.3.2.1. Команда SHOW ENTRY**

Используйте команду SHOW ENTRY для просмотра информации о Ваших заданиях на печать или об определенных заданиях в очереди:

Формат:

**\$ SHOW ENTRY [номер-элемента,...]**

Выводимая информация показывает текущий статус элемента очереди, а также его атрибуты, включая имя задания, владельца, номер элемента очереди и номер очереди.

Добавьте квалификатор /FULL, чтобы вывести более детальную информацию.

Если Вы не указали номер элемента, будет показана информация только о Ваших элементах очереди.

Пример 4-9 показывает, какого вида информация возвращается по команде SHOW ENTRY.

#### Пример 4-9. Использование команды SHOW ENTRY

```
❶ $ SHOW ENTRY
Jobname Username Entry Blocks Status

RA0293_CHAP_ONE ROUNDS 970 265 Pending
 On Generic printer queue POST_20

RA0293_CHAP_TWO ROUNDS 971 287 Pending
 On Generic printer queue POST_20

$
❷ $ SHOW ENTRY/FULL
Jobname Username Entry Blocks Status

RA0293_CHAP_ONE ROUNDS 970 265 Pending (check execution q
ueues)
 On Generic printer queue POST_20
 Submitted 27-NOV-1990 13:32 /FORM=DEFAULT /NOTIFY /PRIORITY=100
 File: _1DUA7:[REGNELL.BUILD]RA0293_CHAP_ONE.PS;1

RA0293_CHAP_TWO ROUNDS 971 287 Pending (check execution q
ueues)
 On Generic printer queue POST_20
 Submitted 27-NOV-1990 13:32 /FORM=DEFAULT /NOTIFY /PRIORITY=100
 File: _1DUA7:[REGNELL.BUILD]RA0293_CHAP_TWO.PS;1

$
❸ $ SHOW ENTRY 970
Jobname Username Entry Blocks Status

RA0293_CHAP_ONE ROUNDS 970 265 Pending
 On Generic printer queue POST_20

$
```

#### Пример 4-9, замечания:

- ❶ Выводится краткий список всех Ваших элементов очереди.
- ❷ Выводится детальная информация о всех Ваших элементах очереди.
- ❸ Выводится информация об определенном элементе очереди.

#### 4.6.3.2.2. Команда SHOW QUEUE

Команда SHOW QUEUE может выводить информацию об очередях, а также о заданиях в очереди.

Формат:

**\$ SHOW QUEUE [имя-очереди]**

Если Вы не указали имя очереди, будут показаны все очереди.

По умолчанию будут показаны только задания, помещенные в очередь Вами.

Добавьте квалификатор /ALL\_ENTRIES, чтобы вывести информацию о заданиях, помещенных в очередь другими пользователями.

Добавьте квалификатор /FULL, чтобы вывести детальную информацию об очереди и заданиях в ней.

Пример 4-10 показывает, какого вида информация возвращается командой SHOW QUEUE.

#### Пример 4-10. Использование команды SHOW QUE

```
❶ $ SHOW QUEUE
Printer queue LPS20$CRIER, on TIDY::CRIER, mounted form LPS$$FORM
(stock=DEFAULT)

Generic printer queue POSTB
```

```

Generic printer queue POSTO

Generic printer queue POST_20

 Jobname Username Entry Blocks Status

 (1 intervening job containing 238 blocks)
 RA0293_CHAP_ONE ROUNDS 970 265 Pending
 RA0293_CHAP_TWO ROUNDS 971 287 Pending
 .
 .
 .
❷ $ SHOW QUEUE POST_20

Generic printer queue POST_20

 Jobname Username Entry Blocks Status

 (1 intervening job containing 238 blocks)
 RA0293_CHAP_ONE ROUNDS 970 265 Pending
 RA0293_CHAP_TWO ROUNDS 971 287 Pending
 .
 .
 .
$
❸ $ SHOW QUEUE POST_20/ALL
Generic printer queue POST_20
 Jobname Username Entry Blocks Status

 no privilege 931 238 Pending
 RA0293_CHAP_ONE ROUNDS 970 265 Pending
 RA0293_CHAP_TWO ROUNDS 971 287 Pending
 .
 .
 .
$
❹ $ SHOW QUEUE POST_20/FULL
Generic printer queue POST_20
 /GENERIC=(LPS20$CRIER) /OWNER=[GROUP1,SYSTEM] /PROTECTION=(S:E,O:D,G
:R,W:W)
 /SCHEDULE=(NOSIZE)
 Jobname Username Entry Blocks Status

 (1 intervening job containing 238 blocks)
 RA0293_CHAP_ONE ROUNDS 970 265 Pending
 Submitted 27-NOV-1990 13:32 /FORM=DEFAULT /NOTIFY /PRIORITY=100
 File: _1DUA7:[REGNELL.BUILD]RA0293_CHAP_ONE.PS;1

 RA0293_CHAP_TWO ROUNDS 971 287 Pending
 Submitted 27-NOV-1990 13:32 /FORM=DEFAULT /NOTIFY /PRIORITY=100
 File: _1DUA7:[REGNELL.BUILD]RA0293_CHAP_TWO.PS;1
$

```

#### Пример 4-10, замечания

- ❶ Введите команду `SHOW QUEUE` без указания имени очереди или каких-либо квалификаторов.  
Будут показаны только задания, помещенные в очередь пользователем.  
Будет показано количество мешающих заданий (если есть).
- ❷ Затребуйте информацию об определенной очереди.
- ❸ Используйте квалификатор `/ALL`, чтобы запросить список всех заданий в указанной очереди.  
Вы можете не иметь привилегий для вывода информации о заданиях, помещенных в очередь другими пользователями.

- ❶ Используйте квалификатор /FULL для запроса детальной информации о Ваших заданиях в указанной очереди.

#### 4.6.3.3. Изменение характеристик Вашего задания на печать

Вы можете использовать команду SET ENTRY для изменения определенных характеристик Вашего задания на печать.

Формат:

**\$ SET ENTRY номер-элемента/квалификатор**

*Номер-элемента* - это номер Вашего задания в очереди, показываемый командой SHOW QUEUE или командой SHOW ENTRY.

Используйте квалификаторы для указания характеристик, подлежащих изменению.

Эта команда не имеет действия на задания, выполняемые в данный момент.

Таблица 4-16 перечисляет некоторые широко используемые квалификаторы команды SET ENTRY.

**Таблица 4-16. Некоторые квалификаторы команды SET ENTRY**

| Операция                                                                                                                  | Квалификатор         | Комментарий                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Потребовать, чтобы задание было задержано, пока не наступит указанное время.                                              | /AFTER=время         | Если указанное время уже прошло, то задание помещается в очередь для немедленной обработки.                                                                                                                                                                     |
| Указать количество печатаемых копий.                                                                                      | /COPIES=n            | Значение n может быть любым числом от 1 до 255.<br><br>Количество копий применимо только к заданию на печать в целом.<br><br>Вы не можете использовать этот квалификатор для указания разных количеств копий для отдельных файлов внутри многофайловых заданий. |
| Указать, будет ли задание доступно для немедленной обработки или оно будет задержано для обработки в более позднее время. | /HOLD                | Если Вы укажете /HOLD, задание не будет отпущено для обработки, пока Вы не введете SET ENTRY/RELEASE.                                                                                                                                                           |
| Потребовать, чтобы все задание на печать было напечатано n раз.                                                           | /JOB_COUNT=n         | Отменяет квалификатор /JOB_COUNT команды PRINT.<br><br>Значение n может быть любым числом от 1 до 255.                                                                                                                                                          |
| Указать, хотите ли Вы быть извещенным о завершении или сбросе задания.                                                    | /NOTIFY<br>/NONOTIFY | Извещение посылается ко всем терминальным сессиям в том же кластере, где Вы вошли в систему.                                                                                                                                                                    |
| Отпустить задание для обработки.                                                                                          | /RELEASE             | Отпускает задания, помещенные в очередь с квалификатором /HOLD или /AFTER.                                                                                                                                                                                      |

| Операция                                                       | Квалификатор                | Комментарий                                                       |
|----------------------------------------------------------------|-----------------------------|-------------------------------------------------------------------|
| Потребовать, чтобы задание было переставлено в другую очередь. | /REQUEUE=<br>имя-очереди[:] | Перемещает задание из первоначальной очереди в указанную очередь. |

Пример 4-11 иллюстрирует использование команды SET ENTRY для изменения характеристик задания на печать.

#### Пример 4-11. Использование команды SET ENTRY

```

❶ $ PRINT/HOLD . MARCH.REPORT
❷ Job MARCH (queue SYS$PRINT, entry 112) holding
.
.
❸ $ SET ENTRY 112/RELEASE/JOB_COUNT=3

```

#### Пример 4-11, замечания:

- ❶ Потребуйте, чтобы файл MARCH.REPORT был помещен в очередь к системному принтеру, но помещен в задержанное состояние.
- ❷ Заметьте, что сообщение "holding" о задержке включает номер элемента очереди.
- ❸ Позднее отпустите задание для печати и потребуйте, чтобы были напечатаны три копии задания.

#### 4.6.3.4. Удаление Вашего задания на печать из очереди

Вы можете использовать команду DELETE/ENTRY для удаления Вашего задания из очереди на печать.

Формат:

**\$ DELETE/ENTRY=(номер-элемента[,...])**

*Номер-элемента[,...]* указывает номер элемента в очереди (или список номеров элементов) заданий, которые должны быть удалены.

- Если Вы указали только один номер, Вы можете опустить скобки.
- Чтобы узнать номер элемента, введите команду SHOW ENTRY или SHOW QUEUE.

Задание может выполняться или ожидать очереди.

Добавьте квалификатор /LOG к команде DELETE/ENTRY, чтобы попросить ее показывать номер элемента для каждого удаляемого задания (/NOLOG - значение по умолчанию).

Пример 4-12 иллюстрирует использование команды DELETE/ENTRY.

#### Пример 4-12. Удаление задания на печать из очереди

```

❶ $ PRINT/QUEUE=ANSI CHAPTER8.MEM
Job CHAPTER8 (queue ANSI, entry 25) pending
.
.
❷ $ SHOW QUEUE ANSI
Generic printer queue ANSI
Jobname Username Entry Blocks Status

CHAPTER8 SMITH 25 976 Pending
❸ $ DELETE/ENTRY=25
❹ Job CHAPTER8 (queue ANSI, entry 25) terminated with error status

```

```
%JBC-F-JOBDELETE, job deleted before execution
%DELETE-I-DELETED, entry 25 aborting or deleted
```

### Пример 4-12, замечания:

- ❶ Запросите печать файла CHAPTER8.MEM в очереди ANSI.  
Система показывает имя задания, имя очереди, в которую помещено задание, номер элемента очереди и его статус.  
Система не может напечатать файл в данный момент, поэтому он помещен в состояние ожидания ("pending").
- ❷ Позднее выяснилось, что перед печатью файл необходимо отредактировать.  
С помощью команды SHOW QUEUE проверьте, что задание еще ожидает и заметьте его номер элемента в очереди.
- ❸ С помощью команды DELETE/ENTRY удалите задание из очереди.
- ❹ Это системные сообщения, связанные с удалением ожидающего задания из очереди.

### 4.6.3.5. Типичные ошибки при манипулировании заданиями на печать

Пример 4-13 иллюстрирует некоторые типичные ошибки при специфицировании и манипулировании заданиями на печать.

### Пример 4-13. Типичные ошибки при манипулировании заданиями на печать

- ❶ \$ PRINT MISSING  
%PRINT-E-OPENIN, error opening \$1\$DUA1:[ROUNDS]MISSING.LIS; as input  
-RMS-E-FNF, file not found  
%PRINT-F-CREJOB, error creating job  
-JBC-E-EMPTYJOB, no files specified in job request  
\$
- ❷ \$ PRINT/QUE=LPS40 LOGIN.COM  
%PRINT-F-CREJOB, error creating job  
-JBC-E-NOSUCHQUE, no such queue  
\$
- ❸ \$ DELETE/ENTRY=2  
%DELETE-W-SEARCHFAIL, error searching for 2  
-JBC-E-NOSUCHENT, no such entry  
\$
- ❹ \$ SHOW ENTRY  
%JBC-E-NOSUCHENT, no such entry

### Пример 4-13, замечания:

- ❶ Вы попытались напечатать несуществующий файл.
  - Проверьте что Вы набрали: не ошиблись ли Вы в имени файла?
  - Проверьте Ваш каталог по умолчанию: находится ли интересующий Вас файл в текущем каталоге?
  - Обратите внимание, что для команды PRINT тип файла по умолчанию LIS. Не нужно ли Вам указать другой тип файла?
- ❷ Вы указали несуществующую очередь.
  - Используйте команду SHOW QUEUE/DEVICES для вывода на экран очереди на печать в Вашей системе.
- ❸ Вы указали несуществующее задание.
  - Используйте команду SHOW ENTRY чтобы вывести список Ваших заданий и проверьте их номера.

- Типичная ошибка - указывать размер в блоках вместо номера элемента в очереди. Номер элемента для следующего задания 735, а не 2.

```
$ SHO ENTRY
Jobname Username Entry Blocks Status

LOGIN ROUNDS 735 2 Pending
On Printer queue SYS$PRINT
```

- ④ Ваше задание выполнено. Это сообщение не указывает на проблему.

#### 4.6.4. Переименование файлов

Используйте команду RENAME, чтобы изменить имя файла или переместить файл в другой каталог на том же диске.

Формат:

**\$ RENAME входной-файл выходной-файл**

Спецификация выходного файла может содержать имя другого каталога, но не должна указывать на другой диск.

Вы можете использовать шаблоны в спецификациях файлов.

- Во входной спецификации: все файлы, чьи спецификации удовлетворяют полям шаблона, будут переименованы.
- В выходной спецификации: шаблон указывает, что данное поле не будет изменено.
- Шаблонные символы в соответствующих полях входной и выходной спецификациях файлов приводят к многократной операции переименования.

Если Вы опустите одну или обе спецификации файлов, команда RENAME подскажет Вам о них:

```
$ RENAME
 _From: TEST.COM
 _To: PRACTICE.FILE
```

Квалификатор /LOG заставляет систему показывать имя каждого переименовываемого файла.

Квалификатор /CONFIRM полезен, когда Вы используете шаблоны для указания подлежащих переименованию файлов. Это защитит от случайного переименования не тех файлов.

Таблица 4-17 показывает примеры операций переименования.

**Таблица 4-17. Использование команды RENAME**

| Пример команды                   | Комментарий                                                                                                                                                                    |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \$ RENAME AVERAGE.OBJ OLDAVERAGE | Изменить имя файла наивысшей существующей версии файла AVERAGE.OBJ<br><br>Если в данный момент не существует файла OLDAVERAGE.OBJ, новому файлу будет присвоен номер версии 1. |
| \$ RENAME *.TXT;* *.OLD;*        | Переименовать все версии всех файлов с типом файла TXT так, чтобы их тип файла стал OLD.<br><br>Имена файлов и номера версий не изменяются.                                    |
| \$ RENAME TEST.COM [.COMMANDS]   | Переименовать наивысшую версию файла                                                                                                                                           |

TEST.COM в подкаталог COMMANDS (на один уровень ниже текущего каталога). В результате это переместит файл из одного каталога в другой.

---

### 4.6.5. Удаление файлов

Две команды позволяют Вам удалять файлы с диска:

- Команда DELETE удаляет один файл или группу файлов.
- Команда PURGE удаляет все, кроме имеющих наибольшие номера, версии одного файла или группы файлов.

#### 4.6.5.1. Использование команды DELETE

Используйте команду DELETE для удаления файлов и освобождения дискового пространства.

Формат:

##### **\$ DELETE файл**

Вы должны указать номер версии файла, который хотите удалить.

- Определенная версия:

```
$ DELETE INVENTORY.LIS;3
```

- Наивысшая версия:

```
$ DELETE INVENTORY.LIS;
```

- Все версии:

```
$ DELETE INVENTORY.LIS;*
```

Квалификатор /LOG заставляет систему показывать имя каждого удаляемого файла.

Квалификатор /CONFIRM полезен, когда Вы используете шаблоны для указания файлов, подлежащих удалению. Это защитит Вас от случайного удаления не тех файлов.

Пример 4-14 иллюстрирует использование квалификаторов /LOG и /CONFIRM с командой DELETE.

#### Пример 4-14. Использование команды DELETE

```
❶ $ DIRECTORY
Directory WORK3:[ROUNDS.TEST]

NOVEMBER.DAT;6 NOVEMBER.DAT;5 OCTOBER.DAT;2 OCTOBER.DAT;1
SEPTEMBER.DAT;1 STATUS.MEMO;1

Total of 6 files.
$
❷ $ DELETE/LOG STATUS.MEMO;1
❸ %DELETE-I-FILDEL, WORK3:[ROUNDS.TEST]STATUS.MEMO;1 deleted (3 blocks)
$
❹ $ DELETE/CONFIRM *.DAT;
❺ DELETE WORK3:[ROUNDS.TEST]NOVEMBER.DAT;6 ? [N]: Y
❻ DELETE WORK3:[ROUNDS.TEST]OCTOBER.DAT;2 ? [N]:
DELETE WORK3:[ROUNDS.TEST]SEPTEMBER.DAT;1 ? [N]: Y
$
```

❶ \$ DIRECTORY

Directory WORK3:[ROUNDS.TEST]

NOVEMBER.DAT;5 OCTOBER.DAT;2 OCTOBER.DAT;1

Total of 3 files.

\$

#### Пример 4-14, замечания:

- ❶ В каталоге есть по несколько версий некоторых файлов.
- ❷ Удалите определенный файл, запросив протокол операции.
- ❸ Это протокольное сообщение.
- ❹ Удалите высшие версии всех файлов с типом файла DAT и затребуйте подтверждение перед удалением каждого файла.
- ❺ Наберите Y (или YES), чтобы удалить файл.
- ❻ Наберите N или нажмите RETURN, если Вы не хотите, чтобы файл был удален.
- ❼ Сравните с первым списком каталога.

#### 4.6.5.2. Использование команды PURGE

Обычно VMS создает новую версию файла каждый раз, когда Вы изменяете файл каким-либо способом<sup>2</sup>.

Используйте команду PURGE для удаления с диска всех версий файла, кроме высшей.

##### \$ PURGE [файл]

Чтобы сохранить больше одной версии, используйте квалификатор /KEEP.

##### \$ PURGE/KEEP=n [файл]

Если Вы опустите спецификацию файла, все файлы в текущем каталоге по умолчанию будут вычищены.

Не включайте в спецификацию файла номер версии.

Квалификатор /LOG заставляет систему показывать имя каждого удаляемого файла.

Квалификатор /CONFIRM полезен, когда Вы используете шаблоны для указания файлов, подлежащих чистке. Это предохраняет от случайного удаления не тех файлов.

Пример 4-15 показывает использование команды PURGE с двумя из наиболее употребительных ее квалификаторов.

<sup>2</sup> Некоторые программы, в том числе и написанные пользователями системы, могут изменять файл, не создавая при этом новой версии. Например, система управления реляционными базами данных DEC Rdb производит изменения в файлах базы данных, размер которых может достигать многих мегабайт, без создания новой версии.

### Пример 4-15. Использование команды PURGE

```

❶ $ DIRECTORY

Directory WORK3:[ROUNDS.TEST]

NOVEMBER.DAT;5 NOVEMBER.DAT;4 NOVEMBER.DAT;3 NOVEMBER.DAT;2
NOVEMBER.DAT;1 OCTOBER.DAT;7 OCTOBER.DAT;6 OCTOBER.DAT;5
OCTOBER.DAT;4 OCTOBER.DAT;3 OCTOBER.DAT;2 OCTOBER.DAT;1

Total of 12 files.
$
❷ $ PURGE/LOG NOVEMBER.DAT
❸ %PURGE-I-FILPURG, WORK3:[ROUNDS.TEST]NOVEMBER.DAT;4 deleted (3 blocks)
%PURGE-I-FILPURG, WORK3:[ROUNDS.TEST]NOVEMBER.DAT;3 deleted (3 blocks)
%PURGE-I-FILPURG, WORK3:[ROUNDS.TEST]NOVEMBER.DAT;2 deleted (3 blocks)
%PURGE-I-FILPURG, WORK3:[ROUNDS.TEST]NOVEMBER.DAT;1 deleted (3 blocks)
❹ %PURGE-I-TOTAL, 4 files deleted (12 blocks)
$
❺ $ PURGE/KEEP=2 OCTOBER.DAT
$
❻ $ DIRECTORY

Directory WORK3:[ROUNDS.TEST]

NOVEMBER.DAT;5 OCTOBER.DAT;7 OCTOBER.DAT;6

Total of 3 files.
$

```

\$ DELETE MEMO.TXT;3

Удалить определенную версию

\$ DELETE MEMO.TXT;\*

Удалить все версии

- ❷ Команда PURGE не позволяет Вам указывать номер версии. Не включайте в команду точку с запятой (;).

\$ PURGE TEST.\*

## 4.7. Выводы

**Периферийные устройства как часть аппаратных средств компьютера VAX**

Аппаратное обеспечение системы VAX делится на четыре подсистемы:

- Центральный процессор
- Консольная подсистема
- Основная память
- Подсистема ввода/вывода

Подсистема ввода/вывода состоит из устройств которые обеспечивают ввод и вывод информации из системы. Эти устройства называются периферийными устройствами.

Как правило, периферийные устройства включают:

- Терминалы
- Печатающие устройства
- Накопители на магнитных дисках (НМД)
- Накопители на магнитных лентах (НМЛ)

Терминалы - это периферийные устройства которые позволяет пользователю набирать команды и данные на клавиатуре и получать сообщения на видеомонитор или печатающее устройство.

Печатающие устройства (принтеры) обеспечивают вывод информации в системе.

Накопители на магнитных дисках (НМД) и лентах (НМЛ) - это устройства которые записывают и считывают информацию с магнитного носителя.

**Имена файлов и их хранение в каталогах**

Спецификация файла обеспечивает операционную систему информацией. необходимой для идентификации файла.

Шаблоны могут подставляться вместо имени каталога, имени файла, типа файла или номера версии в спецификации файла.

Для организации файлов могут быть использованы каталоги и подкаталоги.

Используйте команду DIRECTORY для нахождения отдельного файла или для получения информации о файлах.

## Создание и редактирование файлов

Программное обеспечение OpenVMS предоставляет несколько способов создания, обслуживания и модификации файлов. Два наиболее популярных текстовых редактора - это EVE и EDT.

EVE имеет разнообразные возможности для редактирования текста на видеотерминалах. Его нельзя использовать на печатающих терминалах.

Вы можете перемещать курсор по документу с использованием как клавиш перемещения курсора, так и команд перемещения курсора.

EVE предоставляет обширную вспомогательную информацию в диалоговом режиме.

Расширенные возможности EVE включают возможность устанавливать дополнительную клавиатуру в стиле EDT, включать существующий файл в файл, который Вы редактируете, и выполнять команды DCL без выхода из EVE.

## Работа с файлами

Используйте команду COPY для копирования одного файла или группы файлов в новый файл.

Используйте команду TYPE для просмотра содержимого файла.

Используйте команду PRINT, чтобы послать содержимое файла на устройство вывода.

Используйте команду SHOW ENTRY или команду SHOW QUEUE для просмотра информации о состоянии заданий в очереди на печать.

Используйте команду SET ENTRY для изменения характеристик Вашего задания на печать.

Используйте команду DELETE/ENTRY для удаления Вашего задания из очереди на печать.

Используйте команду RENAME для изменения имени файла или перемещения файла в другой каталог на том же диске.

Используйте команду DELETE или команду PURGE для удаления файлов с диска.

## 4.8. Письменные упражнения

### 4.8.1. Спецификации файлов VMS

Поставьте в соответствие элементам спецификации файла слева определения справа.

Спецификация файла:

TIDY : : DUA3 : [PERSONNEL] MONTHLY\_REPORT . DATA ; 7

| Элементы           | Определения   |
|--------------------|---------------|
| ___ TIDY::         | a. Устройство |
| ___ DUA3:          | b. Каталог    |
| ___ [PERSONNEL]    | c. Имя файла  |
| ___ MONTHLY_REPORT | d. Тип файла  |
| ___ .DATA          | e. Имя узла   |
| ___ ;7             | f. Версия     |

## 4.8.2. Использование шаблонов

Представьте, что Ваш каталог по умолчанию содержит следующие файлы:

|               |               |               |                    |
|---------------|---------------|---------------|--------------------|
| A.DAT;1       | A.FOR;2       | AREA.FOR;2    | AREA.FOR;1 B.DAT;3 |
| B.FOR;1       | C.DAT;4       | C.FOR;1       |                    |
| MAILD22.DAT;2 | MAILF22.DAT;2 | MAILIST.COB;1 | MAILJ14.DAT;1      |

Обращайтесь к этому списку при ответе на следующие вопросы.

1. Перечислите файлы, которые специфицированы следующими спецификациями файлов:
  - a. \*.FOR;2
  - b. \*.FOR
  - c. A\*.\*;\*
  - d. A%%%.\*;\*
  - e. %.DAT
  - f. \*.\*;\*
2. Предложите единственную спецификацию, которая описывает следующие списки файлов:
  - a. A.DAT;1, A.FOR;2
  - b. A.DAT;1, B.DAT;3, C.DAT;4
  - c. MAILD22.DAT;2, MAILF22.DAT;2, MAILJ14.DAT;1
  - d. A.DAT;1, MAILJ14.DAT;1

## 4.8.3. Каталоги

Рядом с каждой из операций, связанных с каталогами, поставьте букву, соответствующую команде VMS, которая лучше всего подходит для выполнения задачи. Вы можете некоторые команды использовать несколько раз, а другие - ни разу.

| Операции                                                    | Команды             |
|-------------------------------------------------------------|---------------------|
| ___ Посмотреть имя текущего каталога по умолчанию.          | a. CREATE           |
| ___ Посмотреть содержание иерархии каталогов.               | b. CREATE/DIRECTORY |
| ___ Добавить каталог к иерархии каталогов.                  | c. DIRECTORY        |
| ___ Изменить Ваш текущий каталог по умолчанию.              | d. SET DEFAULT      |
| ___ Посмотреть имя Вашего текущего устройства по умолчанию. | e. SHOW DEFAULT     |
| ___ Изменить Ваше текущее устройство по умолчанию.          |                     |

## 4.8.4. Команды работы с файлами

Рядом с каждой операцией обслуживания файлов поставьте букву, которая соответствует команде VMS, наиболее подходящей для ее выполнения.

| Операции                                                             | Команды      |
|----------------------------------------------------------------------|--------------|
| ___ Вывести содержимое файла на Ваш терминал.                        | a. COPY      |
| ___ Вывести содержание Вашего каталога по умолчанию на Ваш терминал. | b. DELETE    |
|                                                                      | c. DIRECTORY |

| Операции                                                                              | Команды              |
|---------------------------------------------------------------------------------------|----------------------|
| ___ Удалить файл из Вашего каталога по умолчанию.                                     | d. PRINT             |
| ___ Удалить все, кроме самой последней, версии файла из Вашего каталога по умолчанию. | e. PRINT<br>f. PURGE |
| ___ Создать точный дубликат файла в Вашем каталоге по умолчанию.                      | g. RENAME            |
| ___ Распечатать содержимое файла на системном принтере по умолчанию.                  | h. TYPE              |
| ___ Изменить имя файла.                                                               |                      |

## 4.9. Лабораторные упражнения

### 4.9.1. Имена файлов и хранение их в каталогах

#### 4.9.1.1. Подкаталоги

Выполните следующие шаги в порядке нумерации.

1. Создайте подкаталог с именем [.MEMO].
2. Сделайте подкаталог [.MEMO] своим каталогом по умолчанию.
3. Получите в файле MEMO.LIST список всех файлов в подкаталоге.
4. Сделайте своим каталогом по умолчанию входной каталог, не набирая имени каталога.
5. Используя только одну команду, посмотрите список всех Ваших файлов.
6. Имея каталог по умолчанию все еще установленным во входном каталоге, посмотрите список всех файлов в подкаталоге MEMO.

#### 4.9.1.2. Команда DIRECTORY

Для выполнения этих упражнений сделайте своим каталогом по умолчанию входной каталог.

1. Выведите на терминал список имен файлов в своем каталоге по умолчанию.
2. Выведите на терминал имена этих файлов в одну колонку.
3. Выведите на терминал имена, даты создания и размеры этих файлов.
4. Создайте файл MYFILES.LIST, содержащий список всех файлов во всех Ваших каталогах.
5. Выведите на терминал полный список всех файлов каталога с типом файла LIST.

### 4.9.2. Создание и редактирование файлов

#### 4.9.2.1. Создание файла

1. Используйте редактор EVE для создания файла TESTFILE.TEXT.
2. Наберите строки, приведённые ниже, нажимая RETURN в конце каждой строки.  

Это пример использования редактора EVE.  
 Этот редактор позволяет Вам вводить текст в файл.
3. Введите команду TOP.
4. Выделите весь текст в файле с начала до конца.

5. Удалите этот текст.
6. Вставьте этот текст шесть раз.
7. Используйте клавишами PREV SCREEN и NEXT SCREEN для перемещения курсора по файлу.
8. Используйте клавишу HELP для получения вспомогательной информации по клавишам дополнительной клавиатуры.
9. Закончите сеанс редактирования и сохраните свою работу.

#### **4.9.2.2. Редактирование файла**

1. Начните другой сеанс редактирования, используя файл TESTFILE.TEXT.
2. Измените текст первой строки, чтобы читалось:  
Это образец использования редактора EVE.
3. Вставьте новый абзац после первого предложения  
Это упражнение показывает, как вставлять текст в существующий текстовый файл.  
  
Нажимая дважды RETURN, я могу делать промежутки между абзацами, чтобы они визуально разделялись на странице.  
  
Вставьте еще два раза RETURN после окончания этого последнего предложения.
4. Закончите сеанс редактирования и сохраните файл.
5. Введите команду DIRECTORY TESTFILE.TEXT.
6. Сколько файлов Вы нашли? В чем различие их спецификаций?

#### **4.9.3. Работа с файлами**

##### **4.9.3.1. Копирование и переименование файлов**

1. Введите следующую команду - и у Вас будет файл для использования в данном упражнении:  
\$ DIRECTORY/OUTPUT=SAMPLE.TXT
2. Создайте подкаталог [.MEMO].
3. Скопируйте SAMPLE.TXT из Вашего входного каталога в подкаталог [.MEMO].
4. Переименуйте SAMPLE.TXT из Вашего входного каталога в подкаталог [.MEMO].
5. В чем разница между предыдущими двумя операциями?

##### **4.9.3.2. Печать файлов**

###### **Замечание**

---

Некоторые из данных лабораторных упражнений могут оказаться невыполнимыми в зависимости от аппаратной конфигурации учебного центра. Пожалуйста, проверьте это со своим инструктором.

---

Выполните следующие упражнения за интерактивным терминалом.

1. С использованием редактора или команды DCL, которая создает выходной файл, создайте два текстовых файла в своем каталоге по умолчанию: TEST1.TXT и TEST2.TXT. Содержание этих файлов не имеет значения.
2. Напечатайте TEST1.TXT, используя родовую очередь печати SYS\$PRINT.
3. Используя одну команду PRINT, напечатайте две копии TEST2.TXT.
4. Выведите на экран список всех принтерных очередей на Вашей системе и всех заданий в очередях.
5. Выберите принтерную очередь из показанных на экране. (Принтерная очередь будет иметь то же имя, что и соответствующее устройство, без двоеточия.) Напечатайте тот же самый файл, поместив его непосредственно в принтерную очередь.
6. Напечатайте TEST1.TXT и TEST2.TXT так, чтобы получить две копии первого файла и три копии второго.
7. Пошлите TEST2.TXT в принтерную очередь, запросив, чтобы файл не печатался в течение часа с настоящего момента.
8. Выведите на экран статус задания, ожидающего печати. Удалите это задание из очереди.

### 4.9.3.3. Сообщения об ошибках

1. Введите с терминала следующую командную строку:

```
$ PRODUCE NONESUCH.FIL
```

Что происходит?

2. Введите с терминала следующую командную строку:

```
$ TYPE NONESUCH.FIL
```

Что происходит?

3. Обратите внимание на разницу между двумя сообщениями об ошибках.
4. Продолжайте экспериментировать с вводом неправильных команд.

## 4.10. Решения к письменным упражнениям

### 4.10.1. Спецификации файлов VMS

Поставьте в соответствие элементам спецификации файла слева определения справа.

Спецификация файла:

```
TIDY: :DUA3: [PERSONNEL] MONTHLY_REPORT.DATA;7
```

| Элементы                | Определения   |
|-------------------------|---------------|
| <u>e</u> TIDY::         | a. Устройство |
| <u>a</u> DUA3:          | b. Каталог    |
| <u>b</u> [PERSONNEL]    | c. Имя файла  |
| <u>c</u> MONTHLY_REPORT | d. Тип файла  |
| <u>d</u> .DATA          | e. Имя узла   |
| <u>f</u> ;7             | f. Версия     |

### 4.10.2. Использование шаблонов

Представьте, что Ваш каталог по умолчанию содержит следующие файлы:

|               |               |               |               |         |
|---------------|---------------|---------------|---------------|---------|
| A.DAT;1       | A.FOR;2       | AREA.FOR;2    | AREA.FOR;1    | B.DAT;3 |
| B.FOR;1       | C.DAT;4       | C.FOR;1       |               |         |
| MAILD22.DAT;2 | MAILF22.DAT;2 | MAILIST.COB;1 | MAILJ14.DAT;1 |         |

Обращайтесь к этому списку при ответе на следующие вопросы.

1. Перечислите файлы, которые специфицированы следующими спецификациями файлов:

a. \*.FOR;2

A.FOR;2 AREA.FOR;2

b. \*.FOR

A.FOR;2 AREA.FOR;1 AREA.FOR;2 B.FOR;1 C.FOR;1

c. A\*.\*;\*

A.DAT;1 A.FOR;2 AREA.FOR;1 AREA.FOR;2

d. A%%%.\*;\*

AREA.FOR;1 AREA.FOR;2

e. %.DAT

A.DAT;1 B.DAT;1 C.DAT;4

f. \*.\*;\*

Весь список файлов

2. Предложите единственную спецификацию, которая описывает следующие списки файлов:

a. A.DAT;1, A.FOR;2

A.\*;\*

b. A.DAT;1, B.DAT;3, C.DAT;4

%.DAT;\*

c. MAILD22.DAT;2, MAILF22.DAT;2, MAILJ14.DAT;1

MAIL\*.DAT;\*

d. A.DAT;1, MAILJ14.DAT;1

\*.DAT;1

### 4.10.3. Каталоги

Рядом с каждой из операций, связанных с каталогами, поставьте букву, соответствующую команде VMS, которая лучше всего подходит для выполнения задачи. Вы можете некоторые команды использовать несколько раз, а другие - ни разу.

| Операции                                                | Команды             |
|---------------------------------------------------------|---------------------|
| <u>e</u> Посмотреть имя текущего каталога по умолчанию. | a. CREATE           |
| <u>c</u> Посмотреть содержание иерархии каталогов.      | b. CREATE/DIRECTORY |
| <u>b</u> Добавить каталог к иерархии каталогов.         | c. DIRECTORY        |

| <i>Операции</i>                                                  | <i>Команды</i>  |
|------------------------------------------------------------------|-----------------|
| <u>d</u> Изменить Ваш текущий каталог по умолчанию.              | d. SET DEFAULT  |
| <u>e</u> Посмотреть имя Вашего текущего устройства по умолчанию. | e. SHOW DEFAULT |
| <u>d</u> Изменить Ваше текущее устройство по умолчанию.          |                 |

#### **4.10.4. Команды работы с файлами**

Рядом с каждой операцией обслуживания файлов поставьте букву, которая соответствует команде VMS, наиболее подходящей для ее выполнения.

| <i>Операции</i>                                                                            | <i>Команды</i>            |
|--------------------------------------------------------------------------------------------|---------------------------|
| <u>d</u> Вывести содержимое файла на Ваш терминал.                                         | a. COPY                   |
| <u>c</u> Вывести содержание Вашего каталога по умолчанию на Ваш терминал.                  | b. DELETE<br>c. DIRECTORY |
| <u>b</u> Удалить файл из Вашего каталога по умолчанию.                                     | d. PRINT                  |
| <u>e</u> Удалить все, кроме самой последней, версии файла из Вашего каталога по умолчанию. | e. PRINT<br>f. PURGE      |
| <u>a</u> Создать точный дубликат файла в Вашем каталоге по умолчанию.                      | g. RENAME                 |
| <u>d</u> Распечатать содержимое файла на системном принтере по умолчанию.                  | h. TYPE                   |
| <u>f</u> Изменить имя файла.                                                               |                           |