

3. DCL - язык взаимодействия с операционной системой

Содержание

3. DCL - язык взаимодействия с операционной системой.....	3-1
3.1. Вход и выход из системы OpenVMS.....	3-3
3.1.1. Вход в систему.....	3-3
3.1.2. Создание нескольких сессий.....	3-8
3.1.3. Доступ к удаленной системе OpenVMS.....	3-11
3.1.4. Изменение пароля.....	3-11
3.1.5. Блокирование терминальной сессии.....	3-14
3.1.6. Выход из системы.....	3-15
3.1.7. Выводы.....	3-16
3.2. Ввод команд DCL.....	3-16
3.2.1. Командный язык Digital (DCL).....	3-16
3.2.2. Структура команды DCL.....	3-17
3.2.3. Ввод команд DCL.....	3-18
3.2.4. Повторный вызов команд.....	3-21
3.2.5. Редактирование командных строк.....	3-22
3.2.6. Использование клавиши CTRL.....	3-23
3.2.7. Интерпретация системных сообщений.....	3-24
3.2.8. Выводы.....	3-26
3.2.9. ПРИЛОЖЕНИЕ - ПРОСМОТР И УСТАНОВКА ХАРАКТЕРИСТИК ТЕРМИНАЛА.....	3-27
3.3. Рабочая среда пользователя.....	3-29
3.3.1. ВВЕДЕНИЕ.....	3-29
3.3.2. Цели.....	3-29
3.3.3. Литература.....	3-29
3.3.4. Использование логических имён.....	3-29
3.3.5. Использование логических символов.....	3-37
3.3.6. Создание определений для дополнительной клавиатуры.....	3-40
3.3.7. Командные процедуры.....	3-44
3.3.8. Командная процедура, исполняемая при входе в систему.....	3-46
3.3.9. ВЫВОДЫ.....	3-48
3.4. Письменные упражнения.....	3-49
3.4.1. Терминология входа в систему.....	3-49
3.4.2. Компоненты команды DCL.....	3-49
3.4.3. Клавиши терминала.....	3-50
3.4.4. Системные сообщения.....	3-50
3.5. Лабораторные упражнения.....	3-51
3.5.1. Вход и выход из системы OpenVMS.....	3-51
3.5.2. Ввод команд.....	3-52
3.6. Решения к письменным упражнениям.....	3-52

3.6.1. Терминология входа в систему.....	3-52
3.6.2. Компоненты команды DCL	3-53
3.6.3. Клавиши терминала.....	3-53
3.6.4. Системные сообщения.....	3-54

3.1. Вход и выход из системы OpenVMS

3.1.1. Вход в систему

Когда Вы входите в систему, Вы должны ввести Ваши имя пользователя и пароль.

Ваше имя пользователя состоит из 1 до 12 символов.

Назначается администратором системы

Часто совпадает с Вашей фамилией

Ваш пароль состоит из 1 до 31 символов.

Допустимые символы:

А до Z (верхнего или нижнего регистра)

Цифры от 0 до 9

Знак доллара (\$)

Дефис (-)

Знак подчеркивания (_)

Первоначально назначается системным администратором, но обычно Вы можете изменять его в любое время.

Не отображается на экране при вводе. Рис. 3-1 показывает связь между терминалом и элементами системы OpenVMS

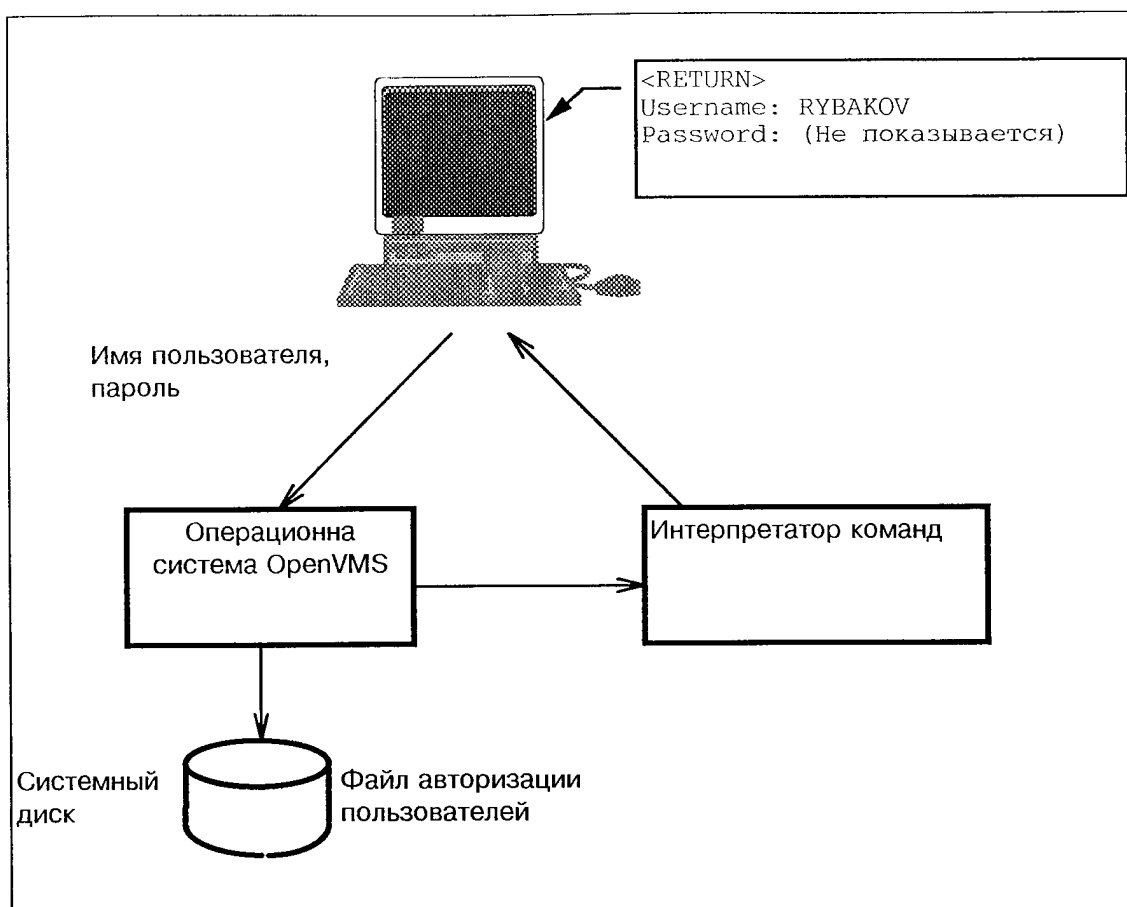


Рис. 3-1. Вход в систему

Рис. 3-1 замечания:

1. Пользователь за терминалом нажимает клавишу RETURN и вводит имя пользователя и пароль в ответ на подсказки.
2. Операционная система VMS сравнивает имя пользователя и пароль с записями в файле авторизации (полномочий) пользователей.
3. Если пользователю разрешено войти в систему, управление передается интерпретатору командного языка VMS, который общается с пользователем через терминал.

3.1.1.1. Действия для входа в систему

Выполните следующие действия для входа в систему:

1. Включите свой терминал.
2. Нажмите клавишу RETURN на клавиатуре терминала.
3. В ответ на подсказку Username:, введите свое имя пользователя и затем нажмите RETURN.
4. В ответ на подсказку Password: введите свой пароль и нажмите RETURN.

Система не показывает Ваш пароль.

Если Ваш счет установлен таким образом, что требуется второй пароль, Вы будете должны ответить на подсказку Password: второй раз.

Пример 3-1 показывает успешный вход пользователя в систему VMS.

Пример 3-1. Успешный вход в систему

```
Username: SHONE
Password:
Welcome to OpenVMS VAX V5.4

      Last interactive login on Friday, 17-MAY-1991 11:11
      Last non interactive login on Friday, 17-MAY-1994 12:00
      You have 14 new Mail messages.

$
```

Пример 3-2 показывает сообщение об ошибке, полученное при вводе неправильной комбинации имени пользователя и пароля

Пример 3-2. Неудачная попытка входа в систему

```
Username: SHON
Password:
User authorization failure
```

3.1.1.2. Вход в систему через терминальный сервер

Терминал может не быть соединен непосредственно с компьютерной системой. В этом случае, возможно, Вам придется получать доступ к системе через терминальный сервер.

Терминальный сервер (или коммуникационный сервер) - это компонент локальной сети, который соединяет устройства, такие как терминалы, принтеры, модемы и компьютеры с сетью.

Каждое устройство присоединяется к серверу с помощью разъема, называемого порт (port).

Сессия (session) - это подсоединение Вашего терминала к обслуживанию с помощью сервера.

Обслуживание (service) предоставляет пользователю терминала один или несколько ресурсов, например, доступ к компьютерной системе или к модему.

На Рис. 3-2 показана конфигурация, содержащая терминальные серверы.

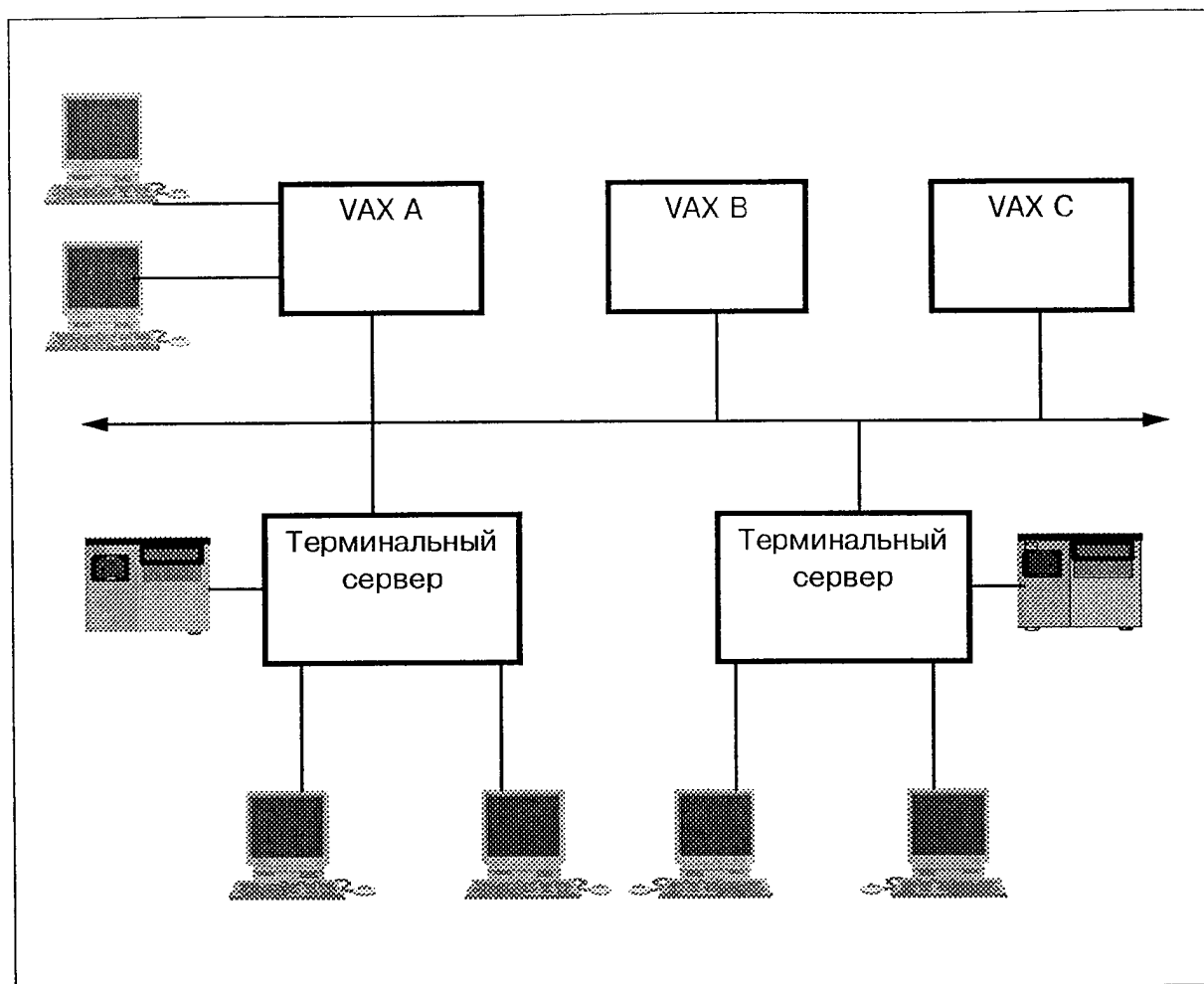


Рис. 3-2. Терминальные серверы

Таблица 3-1 перечисляет наиболее широко используемые команды терминального сервера

Таблица 3-1. Широко используемые команды терминального сервера

Команда	Описание
SHOW SERVICES	Показывает информацию о сетевых обслуживаньях, в настоящий момент доступных для вашего порта.
CONNECT имя_обслуживания	Соединяет с указанным обслуживанием
DISCONNECT SESSION номер_сессии	Отсоединяет указанную сессию или все сессии.
DISCONNECT ALL	
SHOW SESSIONS	Показывает список Ваших сессий в порядке их создания.
RESUME [номер_сессии]	Возвращает к указанной сессии или к текущей сессии, если Вы не указали номер сессии.
FORWARD	Возвращает к сессии, следующей за текущей в списке сессий.
BACKWARD	Возвращает к сессии, предшествующей

Команда	Описание
	текущей в списке сессий.
SET FORWARD переключатель	Указывает символ переключения, используемый для перемещения между сессиями без возвращения к подсказке терминального сервера (локальному режиму).
LOCK	Позволяет Вам безопасно оставить свой терминал без присмотра, когда Вы имеете, по меньшей мере, одну активную сессию.
LOGOUT	Отсоединяет все Ваши существующие сессии

показывает, как пользователь входит в систему через терминальный сервер.

Пример 3-3. Вход и выход из системы через терминальный сервер

❶ DECserver 500 Terminal Server V2.1.5 - LAT V5.1 ABC-1 LAB

Please type HELP if you need assistance

❷ Enter username> KS

❸ Local> SHOW SERVICES

```
Service name      Status  Identification
APP               Available Application Users
CD                Available Course Development
PEOPLE           Available Personnel - Access restricted
.
.
.
```

❹ Local> CONNECT CD

❺ Local -010- Session 1 to CD on node NEAT established

❻ Username: SHONE

Password:

Welcome to VAX/VMS V5.4 on NEAT

\$

❼ \$LOGOUT

SHONE logged out at 17-MAY-1991 10:41:04.69

Local -011- Session 1 disconnected from CD

❸ Local> LOGOUT

Local -020- Logged out port 16 on server Local

Пример 3-3, замечания:

- ❶ Нажмите клавишу RETURN на Вашей клавиатуре, чтобы связаться с терминальным сервером.
- ❷ Спросите у своего администратора системы, что Вы должны вводить как имя пользователя. В Вашей системе оно может не проверяться.
- ❸ Запросите список обслуживаний, которые доступны через терминальный сервер.
- ❹ Соединитесь с компьютерной системой или обслуживанием. В данном случае, "CD" - это обслуживание, состоящее из двух или более компьютерных

систем.

Терминальный сервер распределяет нагрузку между двумя системами.

- ❶ Терминальная сессия установлена в системе, указанной в сообщении.
- ❷ Войдите в систему, как обычно.
- ❸ Выйдите из своей терминальной сессии.
- ❹ Управление возвращено к терминальному серверу.
Выйдите из связи с терминальным сервером.

3.1.1.2.1. Отсоединение во время сессии

Существует много причин, по которым терминальная сессия может быть отсоединена. В зависимости от причины отсоединения, операционная система VMS приостановит Ваш процесс, пока Вы не подсоединитесь вновь.

Когда Вы успешно восстановите контакт и введете свои правильные имя пользователя и пароль, Вам будет сообщено, что у Вас есть отсоединенный процесс и Вам будет дана возможность воссоединиться с ним.

Администратор системы может установить таймер, чтобы ограничить длительность существования отсоединенного процесса. (По умолчанию 15 минут.)

Пример 3-4 показывает, как восстановить соединение, когда Ваша сессия была отсоединена.

Пример 3-4. Отсоединенная сессия

Username: SHONE

Password:

You have the following disconnected process:

Terminal	Process name	Image name
----------	--------------	------------

VT227: SHONE	(none)	
--------------	--------	--

Connect to above listed process [YES]: Y

\$

3.1.2. Создание нескольких сессий

Полезно иметь возможность устанавливать несколько сессий, чтобы:

Работать на более чем одной компьютерной системе с того же самого терминала

Выполнять более одной задачи одновременно с одного терминала

Если Вы получаете доступ к системе через терминальный сервер, Вы можете иметь несколько работающих сессий одновременно.

Вы можете иметь сессии с различными обслуживающими, а также несколько сессий с одним обслуживанием.

Одновременно Вы можете использовать только одну сессию.

Чтобы установить еще одну сессию:

- Вернитесь в локальный режим, нажав клавишу BREAK (на некоторых терминалах F5).
- Затем введите еще одну команду CONNECT с именем желаемого обслуживания.

Пример 3-5 показывает, как Вы можете иметь две работающие сессии на своем терминале.

Пример 3-5. Несколько терминальных сессий

```

❶ DECserver 500 Terminal Server V2.1.5 - LAT V5.1
ABC-1 LAB

Please type HELP if you need assistance

Enter username> KS
Local> CONNECT CD
❷ Local -010- Session 1 to CD on node NEAT established
❸ Username: SHONE
Password:
Welcome to VAX/VMS V5.4 on NEAT
.
.
.
$
❹ Local> C CD
Local -101- 1 other session(s) connected
Local -010- Session 2 to CD on node TIDY established
❺ Username: SHONE
Password:
Welcome to VAX/VMS V5.4 on TIDY
.
.
.
$
❻ Local> SHOW SESSIONS
Port 16: KS
- Session 1: Connected Interactive Local mode Current session: 2
- Session 2: Connected Interactive CD (NEAT)
Local> RESUME 2
Local -012- CD session 2 resumed
$
❼ Local> FORWARD
Local -012- CD session 1 resumed
.
.
.
$
❽ $ LOGOUT
SHONE logged out at 17-MAY-1991 10:41:04.69
Local -011- Session 1 disconnected from CD
❿ Local> SHOW SESSIONS
Port 16: KS
- Session 2: Connected Interactive Local mode Current session: None
Local> RESUME
Local -102- No other sessions active
Local -012- CD session 2 resumed
❶❷$ LOGOUT
SHONE logged out at 17-MAY-1991 10:45:51.80
Local -011- Session 2 disconnected from CD
❶❸Local> LOGOUT
Local -020- Logged out port 16 on server Local

```

Пример 3-5, замечания:

- ❶ Присоединитесь к терминальному серверу и соединитесь с обслуживанием CD.
- ❷ LE>Терминальная сессия номер 1 установлена на системе с названием NEAT.
- ❸ Войдите в систему NEAT.
- ❹ Вернитесь к подсказке терминального сервера (локальный режим), нажав клавишу BREAK, и соединитесь с обслуживанием CD вновь.

- Заметьте, что Вы можете сокращать команды до тех пор, пока они остаются уникальными (C = connect).
 - На этот раз терминальный сервер соединяет Вас с системой под названием TIDY.
- ⑤ Войдите в систему TIDY.
 - ⑥ Нажмите <BREAK> для возвращения в локальный режим на терминальном сервере и запросите список Ваших терминальных сессий. Заметьте, что они находятся на одном обслуживании, но на разных системах.
 - ⑦ Вернитесь к сессии 2.
 - ⑧ Вернитесь в локальный режим, нажав <BREAK>, и перейдите к следующей сессии в Вашем списке сессий. Заметьте, что список циклический.
 - ⑨ Выйдите из сессии 1, вернувшись в локальный режим.
 - ⑩ Заметьте, что осталась только сессия 2.
 - ⑪ Так как у Вас только одна активная сессия, Вы не должны указывать номер сессии, чтобы возобновить работу.
 - ⑫ Выйдите из сессии 2.
 - ⑬ Отсоединитесь от терминального сервера.

Пример 3-6 показывает использование символа переключения вперед при переходе между несколькими терминальными сессиями.

Пример 3-6. Использование символа переключения вперед

- ① DECserver 500 Terminal Server V2.1.5 - LAT V5.1
ABC-1 LAB

Please type HELP if you need assistance

Enter username> KS
- ② Local> SET FORWARD ^X
- ③ Local> CONNECT CD
Local -010- Session 1 to CD on node NEAT established
- ④ Username: SHONE
Password:
Welcome to VAX/VMS V5.4 on NEAT
.
.
.
\$
- ⑤ Local> C CD
Local -101- 1 other session(s) connected
Local -010- Session 2 to CD on node TIDY established
- ⑥ Username: SHONE
Password:
Welcome to VAX/VMS V5.4 on TIDY
.
.
.
\$
- ⑦ Local -012- CD session 1 resumed

Пример 3-6 замечания:

- ① Подсоединитесь к терминальному серверу
- ② Установите символ переключения вперед CTRL/X (^X).
- ③ Подсоединитесь к обслуживанию CD. Терминальная сессия номер 1 установлена на системе под названием NEAT.

- ④ Войдите в систему NEAT.
- ⑤ Вернитесь к подсказке терминального сервера (локальному режиму), нажав <BREAK>, и вновь подсоединитесь к обслуживанию CD. Сессия 2 установлена на системе TIDY.
- ⑥ Войдите в систему TIDY.
- ⑦ Введите CTRL/X (Ваш символ переключения вперед не отображается на экране) для переключения на сессию 1 без возвращения к терминальному серверу.

3.1.3. Доступ к удаленной системе OpenVMS

Раз Вы вошли в систему VMS, Вы можете войти в другую систему, которая соединена с ней через компьютерную сеть.

Команда SET HOST соединяет Ваш терминал (через процессор текущего узла) с другим процессором, называемым удаленным процессором.

Удаленный процессор попросит ввести имя пользователя и пароль.

- У Вас должен быть счет (запись в файле авторизации пользователей) в этой системе.
- Используйте обычную входную процедуру для входа в систему на удаленном процессоре.

На обоих процессорах должно выполняться программное обеспечение DECnet.

Формат:

\$ SET HOST имя узла

Пример 3-7 показывает использование команды SET HOST для входа в удаленную систему.

Пример 3-7. Команда SET HOST

```

① $ SET HOST TIDY
                               Welcome to VAX/VMS V5.4
② Username: SHONE
   Password:
                               Welcome to VAX/VMS version V5.4 on node TIDY
                               Last interactive login on Tuesday, 14-MAY-1991 09:12
                               Last non-interactive login on Tuesday, 14-MAY-1991 09:14
   $
③ $ SHOW TIME
      17-MAY-1991 09:23:32
      .
      .
④ >$ LOGOUT
      SHONE logged out at 17-MAY-1991 09:23:43.03
⑤ %REM-S-END, control returned to node NEAT::
   $

```

Пример 3-7 замечания:

- ① Введите команду SET HOST для входа в удаленную систему.
- ② Введите свое имя пользователя и пароль для удаленной системы.
- ③ Введите команды DCL на удаленной системе.
- ④ Выйдите из сессии на удаленной системе.
- ⑤ Управление возвращается первоначальной системе.

3.1.4. Изменение пароля

Существует три причины, по которым Вам может понадобиться изменить свой пароль:

Пароли должны периодически меняться по соображениям безопасности.

Администратор системы может принудить Вас к изменению Вашего пароля, установив для него срок действия.

Вновь созданный счет пользователя обычно имеет заранее просроченный пароль, позволяющий пользователю войти в систему с использованием этого пароля только один раз.

3.1.4.1. Рекомендации по выбору пароля

Следующие правила рекомендованы для уменьшения шансов раскрытия Вашего пароля.

Создавайте пароли не короче шести символов.

Избегайте имен или слов, которые легко ассоциируются с Вами.

Изменяйте свой пароль не реже раза в месяц.

Возможно, Вам придется использовать пароль, генерируемый системой, если Ваша система этого требует.

3.1.4.2. Проверка допустимости пароля

Операционная система VMS автоматически проверяет пароли.

Пароли проверяются по словарю и списку предыстории для предотвращения использования слов естественного языка или повторного использования старых паролей.

Кроме того, администратор системы может потребовать, чтобы система проверяла пароли, чтобы удостовериться, что они не являются словами, легко ассоциируемыми с местом работы.

3.1.4.3. Команда SET PASSWORD

Используйте команду **SET PASSWORD** для изменения Вашего пароля.

Когда Вы меняете свой пароль, пользовательский ввод не высвечивается на экране.

Вы должны ввести Ваш пароль дважды. Если два введенных пароля не совпадают, пароль не изменяется.

Пример 3-8 показывает подсказки, выводимые на экране при использовании команды SET PASSWORD. Заметьте, что ответы пользователя не высвечиваются.

Пример 3-8. Успешное изменение пароля

```
$ SET PASSWORD
Old password:
New password:
Verification:
```

Пример 3-9 показывает различные ошибки, которые Вы можете допустить при изменении пароля.

Пример 3-9. Ошибки при изменении пароля

```
$ SET PASSWORD
Old password:
New password:
Verification:
```

- ❶ %SET-F-INVPWDLEN, password length must be between 15 and 32 characters;
password not changed
\$
\$ SET PASSWORD
Old password:
New password:
Verification:
- ❷ %SET-F-PWDNOTDIF, new password must be different from current password
\$
\$ SET PASSWORD
Old password:
New password:
Verification:
- ❸ %SYSTEM-F-PWDINHIS, password found in history list; please choose a new password
\$
\$ SET PASSWORD
Old password:
New password:
- ❹ %SYSTEM-F-PWDINDIC, password found in system dictionary; please choose another string
\$
\$ SET PASSWORD
Old password:
New password:
Verification:
- ❺ %SET-I-PWDNOTVER, new password verification error; please try again
Verification:
\$

Пример 3-9 замечания:

- ❶ Новый пароль короче минимальной длины, требуемой в данной системе. Попробуйте более длинный пароль.
- ❷ Новый пароль тот же самый, что и текущий. Придумайте другой пароль.
- ❸ Вы раньше использовали этот пароль. Придумайте другой пароль.
- ❹ Вы пытаетесь использовать пароль, который есть в системном словаре. Придумайте другой пароль.
- ❺ Новый пароль не совпадает с повторно введенным для проверки. Введите новый пароль еще раз более внимательно.

3.1.4.4. Просроченный пароль

Если срок действия Вашего пароля близок к истечению, при входе в систему Вы увидите следующее сообщение.

Пример 3-10. Пароль почти просрочен

```

Welcome to VAX/VMS V5.4
Username: SHONE
Password:
Welcome to VAX/VMS version V5.4 on node TIDY
Last interactive login on Wednesday, 8-MAY-1991 08:44
Last non-interactive login on Wednesday, 8-MAY-1991 08:45
WARNING - Your password expires on Saturday, 11-MAY-1991 11:06

```

Вам разрешено войти один раз в систему с использованием просроченного пароля; затем от Вас потребуют изменить его, как показывает Пример 3-11.

Пример 3-11. Просроченный пароль

```
Welcome to VAX/VMS V5.4
Username: SHONE
Password:
Welcome to VAX/VMS version V5.4 on node NEAT
Your password has expired; you must set a new password to log in

New password:
Verification:
```

3.1.4.5. Пароли, генерируемые системой

От Вас могут потребовать выбрать пароль из списка, генерируемого операционной системой, или Вы можете сами принять такое решение, воспользовавшись командой SET PASSWORD/GENERATE. Пример 3-12 иллюстрирует процедуру выбора пароля, генерируемого системой.

Пример 3-12, Пароли генерируемые системой

```
$ SET PASSWORD/GENERATE
❶ Old password:
❷ pyffhi      pyff-hi
   koogaldo   koo-gal-do
   aimidder   ai-mid-der
   izuzwa     i-zuz-wa
   uscoilf    us-coilf
❸ Choose a password from this list, or press RETURN to get a new list
❹ New password:
❺ Verification:
```

Пример 3-12, замечания:

- ❶ Когда Вы вводите текущий пароль, он не отображается на экране.
- ❷ Система генерирует список возможных паролей с рекомендуемым произношением.
- ❸ Вы можете потребовать еще один список возможных паролей (нажав RETURN), если не нашли пароль, который хотели бы использовать.
- ❹ Новый пароль не отображается на экране, когда Вы его вводите.
- ❺ Когда Вы вводите новый пароль второй раз, он не отображается.

Если эти два пароля не совпадают, Вам дается новый список, из которого нужно выбрать пароль.

Для отмены команды без выбора нового пароля введите CTRL/C или CTRL/Z.

3.1.5. Блокирование терминальной сессии

Если Вы должны временно оставить свой терминал без присмотра, когда у Вас есть хотя бы одна активная сессия, Вы можете защитить свою сессию (сессии) от других пользователей, введя команду терминального сервера LOCK.

Формат:

Local> LOCK

Если команда LOCK не разрешена на Вашем сервере, появится сообщение об ошибке.

Если команда LOCK разрешена на Вашем сервере, сервер запросит у Вас пароль блокировки.

- Пароль - это от 1 до 16 алфавитно-цифровых символов, он не отображается на экране.
- Вы должны ввести пароль второй раз для проверки.

Ваш терминал остается заблокированным, пока Вы не введете тот же самый пароль еще раз для возвращения в локальный режим.

Пример 3-13 показывает, как заблокировать Ваш терминал.

Пример 3-13. Блокирование Вашего терминала

```
❶ Local> LOCK
❷ Lock password>
❸ Verification>
❹ Local -019- Port 16 locked
❺ Unlock password>
❻ Local>
```

Пример 3-13, замечания:

- ❶ Нажмите клавишу BREAK, чтобы войти в локальный режим, и введите команду LOCK.
- ❷ Сервер попросит ввести пароль блокировки (не отображается на экране).
- ❸ Вы должны ввести тот же самый пароль второй раз для проверки.
- ❹ Пароли совпали, поэтому появилось сообщение о блокировании.
- ❺ Вы должны ввести пароль снова, чтобы разблокировать терминал.
- ❻ Ввод правильного пароля возвращает Вас в локальный режим. Вы можете продолжить терминальную сессию.

3.1.6. Выход из системы

Используйте команду LOGOUT для завершения своей терминальной сессии.

В ответ на подсказку DCL по умолчанию (\$), введите **LOGOUT** и нажмите RETURN.

Команда **LOGOUT/FULL** показывает суммарную учетную информацию о терминальной сессии.

Когда Вы выходите из системы VMS, Вы видите информацию, которую показывает Пример 3-14

Пример 3-14. Выход из системы VMS

```
LOGOUT
SHONE logged out at 29-MAR-1991 13:27:02.58
```

Учетную информацию, выводимую при использовании команды LOGOUT/FULL, показывает Пример 3-15

Пример 3-15. LOGOUT/FULL

```
$LOGOUT/FULL
SHONE logged out at 29-MAR-1991 13:47:18.06
Accounting information:
Buffered I/O count:      83          Peak working set size: 362
Direct I/O count:       44          Peak page file size:   2900
Page faults:           697         Mounted volumes:      0
Charged CPU time: 00:02:04.73  Elapsed time:   0 00:14:30.06
```

Если Вы вошли в систему через терминальный сервер, Вы также должны выйти из соединения с терминальным сервером. Пример 3-16 показывает, как пользователь выходит из соединения с терминальным сервером после ввода команды DCL LOGOUT.

Пример 3-16. Отсоединение от терминального сервера

```
$LOGOUT
SHONE      logged out at 29-MAR-1991 16:04:41.36
Local -011- Session 1 disconnected from CD
Local> LOGOUT
Local -020- Logged out port 16 on server ABC121
```

3.1.7. Выводы

Операционная система управляет ресурсами компьютерной системы.

Когда Вы входите в систему VMS, Вы должны ввести имя пользователя и пароль.

Вам может потребоваться подсоединиться к терминальному серверу, чтобы получить доступ к системе.

Если Вы получаете доступ к своей системе через терминальный сервер, у Вас может быть в работе одновременно более одной сессии.

Используйте команду SET HOST, чтобы войти в другую систему, связанную с Вашей по сети.

Используйте команду SET PASSWORD для изменения Вашего пароля.

Вы можете защитить Вашу сессию (сессии) от других пользователей, введя команду терминального сервера LOCK, когда Вы должны временно оставить свой терминал без присмотра.

Используйте команду LOGOUT для завершения терминальной сессии.

3.2. Ввод команд DCL

3.2.1. Командный язык Digital (DCL)

Интерфейс пользователя - это способ общения пользователя с операционной системой. В системе VMS интерфейсом пользователя является Командный язык Digital (DCL).

Другие операционные системы могут использовать непонятные имена команд или сложный командный синтаксис. DCL использует обычные слова для своих команд и квалификаторов. Использование обычных слов облегчает:

- Запоминание команд DCL

- Ввод команд DCL с терминала

- Обнаружение и исправление синтаксических ошибок

Команды DCL могут использоваться, чтобы:

- Выполнять файловые операции, такие как копирование, переименование и удаление файлов или просмотр каталогов.

- Выводить на экран информацию о системе, других пользователях в системе, устройствах, присоединенных к системе, доступных ресурсах и так далее.

- Выполнять системные утилиты, прикладные программы или программы, написанные пользователем.

Примеры команд DCL:

- PRINT

- COPY

- EDIT

- DELETE

RENAME
SET PASSWORD
SHOW TIME

3.2.2. Структура команды DCL

Основные элементы команды DCL показаны на Рис. 3-3

Рис. 3-3. Структура команды DCL

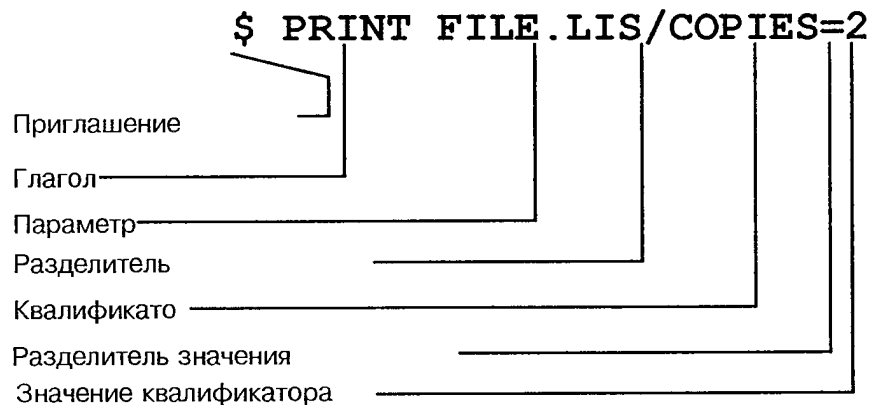


Таблица 3-2. описывает Основные элементы команды DCL.

Таблица 3-2. Элементы команды DCL

Элемент команды	Комментарии
Глагол (Verb)	Глагол командной строки указывает действие, которое Вы запрашиваете.
Ключевое слово (Keyword)	\$ HELP Иногда глагол требует ключевое слово для дальнейшего уточнения действия, которое необходимо выполнить. Ключевое слово - это слово, зарезервированное для использования в некоторых точно определенных синтаксических форматах. \$ SHOW TERMINAL \$ SHOW TIME \$ SHOW USERS В этих командах SHOW - это глагол, а TERMINAL, TIME и USERS - ключевые слова.
Параметр (Parameter)	Параметры принимают на себя действие глагола. Значениями параметров могут быть спецификации файлов, имена очередей, логические имена или другие значения. \$ PRINT FILE1.TXT В этой команде PRINT - это глагол, а FILE1.TXT - параметр.
Квалификатор (Qualifier)	Квалификаторы командной строки описывают или модифицируют действие, выполняемое глаголом. Каждому квалификатору предшествует наклонная черта (/) \$ SHOW PROCESS/ALL Квалификатор /ALL модифицирует действие, которое нужно выполнить.
Значение квалификатора	Некоторые квалификаторы принимают одно или несколько значений. Значение присваивает квалификатору определенную

Элемент команды	Комментарии
(Qualifier value)	величину. \$ PRINT/COPIES=2 FILE1.TXT В вышеприведенной команде /COPIES=2 - это квалификатор глагола PRINT. Величина этого квалификатора - целое число 2.
! (необязателен)	Восклицательный знак указывает на комментарий. DCL игнорирует все, что следует в командной строке за восклицательным знаком. \$ SHOW PROCESS !Этот комментарий игнорируется

3.2.2.1. Квалификаторы команд

Таблица 3-3 описывает две категории командных квалификаторов.

Таблица 3-3. Квалификаторы команд DCL

Тип	Описание
Командный квалификатор	Командные квалификаторы имеют то же самое значение, независимо от того, в каком месте командной строки они появляются. Примеры: \$ PRINT/HOLD FILE1.TXT \$ PRINT FILE1.TXT/HOLD Так как /HOLD - это командный квалификатор, эти две команды имеют одинаковое действие. Обе команды помещают запрос в задержанное состояние.
Позиционный квалификатор	Позиционные квалификаторы имеют различное значение в зависимости от того, в каком месте командной строки они появляются. Пример: \$ PRINT/COPIES=2 FILE1.TXT,FILE2.TXT Позиционный квалификатор, помещенный после глагола, но перед первым параметром, действует на всю командную строку. Следовательно, вышеприведенная команда требует напечатать две копии файла FILE1.TXT и две копии FILE2.TXT. Пример: \$ PRINT FILE1.TXT,FILE2.TXT/COPIES=2 Позиционный квалификатор, помещенный после параметра, действует только на этот параметр. Вышеприведенная командная строка требует напечатать одну копию FILE1.TXT и две копии FILE2.TXT.

3.2.3. Ввод команд DCL

Используйте следующие правила при вводе команд DCL:

Используйте любую комбинацию букв верхнего и нижнего регистра.

Одна командная строка может содержать до 1024 символов.

Один или более пробелов или табуляторов разделяют команды, ключевые слова и параметры.

Наклонная черта (/) отделяет квалификаторы от команд и параметров. Начинайте каждый квалификатор с наклонной черты.

Запятые (,) разделяют элементы в списке параметров.

Если командная строка длиннее одной строки, завершите строку дефисом и нажмите RETURN. Система выдаст подсказку, предваренную знаком подчеркивания, и Вы сможете продолжить ввод.

```
$ SHOW
_ $ TIME
```

Каждый элемент команды не может превышать 255 символов.

Вы можете сокращать любую команду или квалификатор, вводя только первые четыре символа. (Некоторые команды уникальны при сокращении до менее четырех символов.)

Нажатие клавиши RETURN передает командную строку интерпретатору командного языка DCL (Command Language Interpreter - CLI) для выполнения вне зависимости от положения курсора в строке.

3.2.3.1. Возможности DCL

Для удобства использования DCL вводятся дополнительные соглашения.

Продолжать команды DCL на других строках с использованием дефиса (-)

Сокращать команды и другие ключевые слова

Позволить DCL подсказывать о требуемых и необязательных параметрах

Включать комментарии в конце командных строк, используя восклицательный знак (!)

Таблица 3-4 иллюстрирует эти возможности.

Таблица 3-4. Возможности DCL

Возможность	Пример	Комментарии
Продолжение	\$ PRINT/COPIES=2 - _ \$FILE1.TXT, FILE2.TXT - _ \$ FILE3.TXT, FILE4.TXT	Дефис продолжает командную строку на нескольких строках ввода.
Сокращение	\$ LOGOUT \$ LOGO \$ LO	Вы можете также сокращать команды и ключевые слова до четырех или менее символов. Каждое сокращение должно быть уникальным.
Подсказывание	\$ SHOW _ what: TIME 19-OCT-1990 16:18:53	Для команд, имеющих параметры. Если Вы ввели команду и нажали RETURN, DCL подсказывает о недостающих параметрах.
Комментарии	\$!ALL process information \$ SHOW PROCESS/ALL	Восклицательный знак предваряет комментарии, используемые для документирования команд в командных процедурах.

3.2.3.2. Общие соглашения по формированию команд DCL

Многие команды DCL следуют общим соглашениям, как показывает Таблица 3-5

Таблица 3-5. Общие соглашения по формированию команд DCL

Функция	Примеры команд
Используйте команду SHOW для вывода на экран информации.	\$ SHOW TERMINAL \$ SHOW TIME \$ SHOW USERS
Используйте команду SET для изменения характеристик.	\$ SET PASSWORD \$ SET HOST TIDY \$ SET TERMINAL/WIDTH=132
Используйте NO для отрицания параметра или квалификатора.	\$ SET BROADCAST=NOMAIL \$ SET NOVERIFY \$ TYPE/NOPAGE
Используйте команду HELP для запроса вспомогательной информации.	\$ HELP

3.2.3.3. Использование значений по умолчанию

Значение по умолчанию - это значение, которое использует DCL, если Вы не определили значение параметра или квалификатора. Таблица 3-6 показывает некоторые способы использования этой возможности DCL.

Таблица 3-6. Использование значений по умолчанию

Команда DCL	Используемое значение по умолчанию
DIRECTORY	Все файлы, краткий список
PRINT, TYPE	Тип файла .LIS
PURGE	Все файлы, сохранить последние версии
RUN (выполнить программу)	Тип файла .EXE
@ (выполнить командную процедуру)	Тип файла .COM

3.2.3.4. Примеры команд DCL

3.2.3.4.1. Команда SHOW TERMINAL

Используйте команду **SHOW TERMINAL** для просмотра текущих характеристик Вашего терминала.

Каждая характеристика соответствует параметру команды SET TERMINAL.

Пример 3-17 показывает выходную информацию команды SHOW TERMINAL

Пример 3-17. Выходная информация команды SHOW TERMINAL

```
$ SHOW TERMINAL
```

```

Terminal: _VTA145:      Device_Type: PRO_Series      Owner: SMITH
Physical terminal: _LTA88:      Username: SMITH
Input: 9600      Lffill: 0      Width: 80      Parity: None
Output: 9600      Crfill: 0      Page: 24
Terminal Characteristics:
Interactive      Echo      Type_ahead      No Escape
No Hostsync      Ttsync      Lowercase      Tab
Wrap      Scope      No Remote      No Eightbit
Broadcast      No Readsynchron      No Form      Fulldup
No Modem      No Local_echo      No Autobaud      Hangup
No Brdcstmbx      No DMA      No Altypeahd      Set_speed
Line Editing      Overstrikeediting      No Fallback      No Dialup
No Secure server Disconnect      No Pasthru      No Syspassword
No SIXEL Graphics No Soft Characters      Printer port      Numeric keypad
ANSI_CRT      Regis      No Block_mode      Advanced_video
Edit_mode      DEC_CRT      No DEC_CRT2      No DEC_CRT3

```

3.2.3.4.2. Команда SET TERMINAL

Используйте команду **SET TERMINAL**, чтобы изменить одну или несколько характеристик своего терминала для определенных приложений или для отмены системных характеристик по умолчанию.

Формат:

\$ SET TERMINAL /квалификатор[=значение]

Пример команды:

```
$ SET TERMINAL/WIDTH=132
```

Эта команда изменяет ширину экрана терминала и делает ее равной 132 символам.

Таблица 3-7 показывает некоторые другие квалификаторы команды SET TERMINAL.

Таблица 3-7. Некоторые квалификаторы команды SET TERMINAL

Функция	Квалификатор
Выбирает тип терминала.	DEVICE_TYPE=тип
Заставляет сам терминал установить соответствующий тип терминала. (Используйте только с терминалами Digital)	/INQUIRE
Устанавливает число отображаемых строк на странице.	/PAGE[=(n)]
Включает/выключает получение терминалом сообщений оператора.	/BROADCAST /NOBROADCAST
Определяет максимальное число символов в строке.	/WIDTH=(n)

3.2.4. Повторный вызов команд

DCL позволяет пользователю повторно вызывать ранее введенные команды, чтобы модифицировать или повторно ввести их.

3.2.4.1. Команда RECALL

Команда **RECALL** выводит на экран ранее введенные команды, чтобы пользователь мог повторно их использовать. В буфере RECALL хранится до 20 команд.

Нажмите RETURN после того, как вы повторно вывели на экран команду, чтобы система ее выполнила. Курсор при этом может находиться не в конце команды.

Таблица 3-8 показывает различные способы использования команды RECALL.

Таблица 3-8. Использование команды RECALL

Операция	Команда/Квалификатор	Комментарии
Просмотреть буфер RECALL	RECALL/ALL	Выводит на экран пронумерованный список введенных Вами команд, не более 20 команд.
Повторно вызвать определенную команду из списка	RECALL 3	Повторно вызывает третью введенную команду, считая от последней.
Повторно вызывает команду определенного типа	RECALL PRINT	Повторно вызывает последнюю введенную команду PRINT.
Очищает буфер RECALL	RECALL/ERASE	Уничтожает содержимое буфера RECALL.

3.2.4.2. Клавиши СТРЕЛКА ВВЕРХ и СТРЕЛКА ВНИЗ

Клавиши СТРЕЛКА ВВЕРХ и СТРЕЛКА ВНИЗ, (Таблица 3-9), могут быть также использованы для повторного вызова команд.

Таблица 3-9. Использование клавиш СТРЕЛКА ВВЕРХ и СТРЕЛКА ВНИЗ

Операция	Специальная Функциональная клавиша	Комментарии
Вызвать повторно самую последнюю команду	СТРЕЛКА ВВЕРХ	Последовательно вызывает последнюю команду, переданную интерпретатору командных строк DCL. (CTRL/V выполняет ту же функцию.)
Двигаться по буферу команд	СТРЕЛКА ВНИЗ	Вместе с клавишей СТРЕЛКА ВВЕРХ позволяет пользователю передвигаться вверх и вниз по списку команд, хранимых в буфере RECALL.

3.2.5. Редактирование командных строк

Для редактирования командной строки, которую Вы вводите или вызвали повторно:

Используйте клавиши со стрелками влево или вправо для движения курсора по текущей командной строке.

Используйте клавиши CTRL/H или F12 для передвигания курсора в левый конец командной строки.

Используйте CTRL/E для движения курсора в правый конец командной строки.

Используйте CTRL/A или F14 для переключения режимов вставки и замены.

- В режиме замены (overstrike) каждый вводимый символ заменяет собой символ, находящийся в текущей позиции курсора. Режим замены является режимом по умолчанию.
- В режиме вставки (insert) символы справа от курсора сдвигаются вправо, чтобы дать место новым символам.

- Чтобы установить режим редактирования Вашего терминала, введите одну из следующих команд:

```
SET TERMINAL/INSERT
SET TERMINAL/OVERSTRIKE
```

Для изменения информации в командной строке:

Нажмите клавишу DELETE для удаления символа, находящегося слева от курсора.

Нажмите LF, CTRL/J, или F13 для удаления предыдущего слова.

Нажмите CTRL/U для удаления всех символов слева от курсора.

3.2.5.1. Пример повторного вызова и редактирования командной строки

Следующая последовательность иллюстрирует использование повторного вызова и редактирования команд. Терминал установлен для редактирования в режиме ВСТАВКИ. Заметьте, что клавиша СТРЕЛКА ВВЕРХ может быть использована вместо команды RECALL.

1. Вы неправильно ввели команду, но не заметили этого и нажали RETURN.

```
$ SNOW DEFAULT
%DCL-W-IVVERB, unrecognized command verb - check validity and spelling
\SNOW\
```

2. Вы ввели другую команду, прежде чем обнаружили ошибку.

```
$ SHOW TIME
22-OCT-1990 13:26.21
```

3. Вы увидели предыдущую ошибку и ввели RECALL/ALL, чтобы получить список записей в буфере повторного вызова.

```
$ RECALL/ALL
1 SHOW TIME
2 SNOW DEFAULT
```

4. Вы ввели RECALL 2, чтобы вывести на экран неправильно введенную команду. Система показала командную строку.

```
$ RECALL 2
$ SNOW DEFAULT
```

5. Теперь используйте управляющие символы для редактирования командной строки. Нажмите, ← чтобы переместить курсор на N.

```
$ SNOW DEFAULT
```

6. Нажмите клавишу "H", чтобы заменить "N".

```
$ SHOW DEFAULT
```

7. Нажмите клавишу RETURN. Даже если курсор не находится в конце строки, вся командная строка посылается интерпретатору командных строк.

```
$ SHOW DEFAULT
WORK3: [PICCADACI]
```

3.2.6. Использование клавиши CTRL

Некоторые клавиши терминала имеют специальные значения, когда нажимаются с клавишей CTRL. Таблица 3-10 показывает широко используемые комбинации управляющих клавиш.

Таблица 3-10. Функции управляющих клавиш терминала

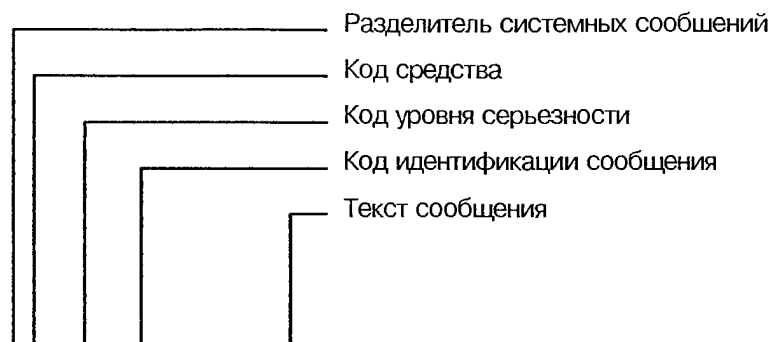
Функция	Комбинация клавиш	Выводимое сообщение
Переключает из режима замены в режим вставки (то же, что и <F14>).	CTRL/A	

<i>Функция</i>	<i>Комбинация клавиш</i>	<i>Выводимое сообщение</i>
Повторно вызывает самую последнюю команду (то же, что и СТРЕЛКА ВВЕРХ).	CTRL/B	
Прерывает выполнение команды или программы.	CTRL/Y	Interrupt
Прерывает или отменяет команду или программу.	CTRL/C	Interrupt или Cancel
Временно останавливает поток информации, выводимой на терминал.	CTRL/S	
Возобновляет поток информации, выводимой на терминал.	CTRL/Q	
Останавливает поток информации, выводимой на терминал. Информация продолжает идти, но терминал не выводит ее на экран или бумагу.	CTRL/O	Output Off
Возобновляет поток информации на экран или бумагу.	CTRL/O	Output On
Показывает однострочное сообщение о вашем процессе.	CTRL/T	
Передвигает курсор к началу текущей строки.	CTRL/H	
Передвигает курсор к концу текущей строки.	CTRL/E	
Регенерирует текущую строку.	CTRL/R	
Удаляет слово слева от курсора (то же, что и <F13>).	CTRL/J	
Удаляет все символы слева от курсора.	CTRL/U	

3.2.7. Интерпретация системных сообщений

Операционная система VMS посылает сообщения в ответ на многие команды, которые Вы вводите с терминала. Эти сообщения могут рассказать Вам, что определенная операция была неудачной, и подсказать возможную причину ошибки. Система обычно не показывает сообщений, которые подтверждают успешное выполнение команды. Рис. 3-4 дает пример системного сообщения, показанного после того, как пользователь ввел "THINK".

Рис. 3-4. Системное сообщение



%DCL-W-IVVERB, unrecognized command verb - check validity and spelling
 \THINK\

3.2.7.1. Элементы системного сообщения

Таблица 3-11 описывает элементы системного сообщения, используя Рис. 3-4 в качестве примера.

Таблица 3-11. Элементы системного сообщения

Элемент сообщения	Пример	Комментарии
Разделитель системного сообщения	%, -	Все системные сообщения начинаются либо с знака процента (%), либо с дефиса (-). Знак процента предшествует первому полученному системному сообщению, тогда как дефис предшествует дополнительным сообщениям.
Код средства	DCL	Указывает часть операционной системы, которая обнаружила ошибку.
Код уровня серьезности	W	Описывает серьезность ошибки.
Код идентификации сообщения	IVVERB	Используется для нахождения дальнейшей информации о сообщении.
Текст сообщения	>unrecognized command verb - check validity and spelling	Дает более детальную информацию о сообщении и в случае ошибки подсказывает действие по исправлению.

3.2.7.2. Уровни серьезности системных сообщений

Таблица 3-12 описывает коды серьезности, которые могут появляться в системных сообщениях.

* Если Вам нужно сообщить об ошибке администратору системы или сопровождающему специалисту, обязательно запишите сообщение. Если сообщение состоит более, чем из одной строки, запишите все строки.

Таблица 3-12. Уровни серьезности в системных сообщениях

Уровень серьезности	Сокращение	Комментарий и пример
Успех	S	Команда выполнена успешно. (Операционная система VMS обычно не показывает сообщений об успешном выполнении.) \$ COPY/LOG A.TXT B.TXT %COPY-S-COPIED, WORK3:[USER]A.TXT;1 copied to ...
Информация	I	Операционная система VMS иногда показывает дополнительную информацию об успехе операции. \$ PURGE/LOG %PURGE-I-FILPURG, WORK3:[USER]TEST.LIS;2 %PURGE-I-FILPURG, WORK3:[USER]TEST.LIS;1 %PURGE-I-TOTAL, 2 files deleted (6 blocks)
Предупреждение	W	Некоторые, но не все, Ваши запросы могут быть выполнены, и Вам рекомендуется проверить выходную информацию команды или программы. \$ DIRECTORY NONEXISTENT.FILE %DIRECT-W-NOFILES, no files found
Ошибка	E	Выходная информация или результат программы неверен, но система может попытаться продолжить выполнение. \$ DELETE A.TXT %DELETE-E-DELVER, explicit version number
Серьезная (или фатальная) ошибка	F	Операция потерпела неудачу. Система не может продолжить выполнение запроса. \$ DEASSIGN Y %SYSTEM-F-NOLOGNAM, no logical name match

3.2.8. Выводы

Командный язык Digital (DCL) - это средство общения пользователя с операционной системой.

DCL предоставляет множество полезных возможностей, включая продолжение команды на новых строках, сокращение команд и подсказки о недостающих параметрах.

Команда RECALL или клавиша СТРЕЛКА ВВЕРХ может быть использована для повторного вызова ранее введенных команд, как для модификации их, так и для повторного ввода.

Командная строка DCL может редактироваться с использованием различных клавиш терминала.

Клавиша CTRL изменяет значение многих клавиш терминала.

Системные сообщения VMS говорят Вам, успешна ли определенная операция, и подсказывают возможную причину ошибки.

3.2.9. ПРИЛОЖЕНИЕ - ПРОСМОТР И УСТАНОВКА ХАРАКТЕРИСТИК ТЕРМИНАЛА

Вы можете просматривать и модифицировать характеристики Вашего терминала, используя клавишу терминала Set-Up или команды DCL *SHOW TERMINAL* и *SET TERMINAL*.

3.2.9.1. Клавиша Set-Up

Режим установки - это последовательность экранов дисплея, которые позволяют Вам проверять и изменять рабочие свойства терминала с помощью клавиатуры.

Каждый экран перечисляет набор рабочих свойств терминала.

Некоторые свойства предназначены для Вашего удобства, другие определяются требованиями Вашей компьютерной системы.

Каждое свойство режима установки имеет заводское значение по умолчанию. Если Вы изменили значения, Вы можете вернуть заводские значения в режиме установки.

Терминал сохраняет Ваши значения установок, а также заводские значения по умолчанию. Когда Вы выключаете питание, Вы не теряете свои сохраненные установки.

3.2.9.1.1. Вход и выход из режима установки

Чтобы войти в режим установки, нажмите клавишу Set-Up (F3).

Информация, отображаемая в данный момент на экране, пропадает.

Экран Set-Up Directory (Каталог установок) изображается на Вашем терминале.

Каталог установок перечисляет все другие экраны режима установки. Вы можете выбрать любой другой экран режима установки из Каталога установок.

Чтобы выйти из режима установки, снова нажмите Set-Up.

Вы можете покинуть режим установки из любого экрана этого режима.

Информация, которая была ранее на экране, появится вновь, когда Вы покинете режим установки.

3.2.9.1.2. Поля действий в режиме установки

Большинство экранов режима установки имеют несколько *полей действий*.

Когда Вы выбираете поле действия, терминал немедленно выполняет это действие.

Используйте клавиши со стрелками для перемещения курсора между полями действий.

Нажмите клавишу ENTER, чтобы выбрать действие, выделенное курсором.

3.2.9.2. Установка и просмотр характеристик терминала из DCL

3.2.9.2.1. Команда SHOW TERMINAL

Используйте команду **SHOW TERMINAL** для просмотра текущих характеристик Вашего терминала.

Используйте эту команду для получения информации, относящейся к установкам Вашего терминала.

Каждая характеристика соответствует квалификатору команды SET TERMINAL

Пример 3-18 показывает выходную информацию команды SHOW TERMINAL

Пример 3-18. Выходная информация команды SHOW TERMINAL

```
$ SHOW TERMINAL
Terminal: _VTA145:      Device_Type: PRO_Series      Owner: SMITH
Physical terminal: _LTA88:      Username: SMITH
Input:  9600      Lffill: 0      Width:  80      Parity: None
Output: 9600      Crfill: 0      Page:   24
Terminal Characteristics:
Interactive      Echo              Type_ahead      No Escape
No Hostsync     Ttsync             Lowercase       Tab
Wrap            Scope              No Remote       No Eightbit
Broadcast       No Readsynchron   No Form         Fulldup
No Modem        No Local_echo     No Autobaud     Hangup
No Brcstmbx     No DMA             No Altypeahd    Set_speed
Line Editing    Overstrikeediting No Fallback     No Dialup
No Secure server Disconnect     No Psthru       No Syspassword
No SIXEL Graphics No Soft Characters Printer port    Numeric keypad
ANSI_CRT        Regis              No Block_mode   Advanced_video
Edit_mode       DEC_CRT            No DEC_CRT2     No DEC_CRT3
```

3.2.9.2.2. Команда SET TERMINAL

Команда **SET TERMINAL** изменяет интерпретацию системой характеристик терминала.

Эта команда позволяет Вам изменить одну или более характеристик Вашего терминала для особых приложений, либо для отмены системных характеристик по умолчанию.

Формат:

\$ SET TERMINAL /квалификатор[=значение]

Пример команды:

```
$ SET TERMINAL/WIDTH=132
```

Эта команда изменяет ширину экрана терминала и делает ее равной 132 символам.

Таблица 3-13 показывает еще несколько квалификаторов команды SET TERMINAL.

Таблица 3-13. Другие квалификаторы команды SET TERMINAL

Функция	Квалификатор
Выбрать тип терминала.	/DEVICE_TYPE=тип
Заставить сам терминал установить соответствующий тип терминала. (Используйте только с терминалами производства Digital.)	/INQUIRE
Установить число строк, отображаемых на странице.	/PAGE[=n]
Произвести изменения, которые остаются в силе, когда пользователь выходит из системы и входит в нее снова.	/PERMANENT
Определяет скорость, с которой терминал передает и принимает данные.	/SPEED=скорость
Включает/выключает реакцию системы на нажатие клавиши терминала HOLD SCREEN.	/TTSYNC /NOTTSYNC
Включает/выключает получение терминалом сообщений оператора.	/BROADCAST /NOBROADCAST

3.3. Рабочая среда пользователя

3.3.1. ВВЕДЕНИЕ

ОС VMS дает ряд возможностей которые позволяют настроить рабочую среду в соответствии с требованиями удобства и эффективности.

Имена часто (совместно) используемых файлов и каталогов могут быть представлены достаточно кратко, т.е. логические имена проще в использовании и запоминании, чем полные спецификации имен файлов.

Логические символы могут использоваться для сокращения наиболее часто используемых команд.

Многие клавиши видеотерминала могут быть переопределены для того, чтобы вводить команды нажатием одной или двух клавиш.

Для того, чтобы постоянно использовать эти возможности при входе в систему можно создать специальный командный файл LOGIN.COM, который обеспечит автоматическое исполнение необходимых вам функций.

Эта часть дает понятие о логических именах и символах, объясняет как создавать описание функциональных клавиш и создавать простые командные процедуры исполняемые при входе в систему.

3.3.2. Цели

Для адаптации рабочего окружения и решения исполняемых им специфических задач оператор VMS должен:

Создавать и использовать логические имена для спецификаций файлов.

Создавать и использовать символы для определения синонимов команд.

Определять и использовать клавиши дополнительной (функциональной) клавиатуры.

Создавать простые командные процедуры LOGIN.COM.

3.3.3. Литература

VMS DCL Dictionary

User's Guide для вашего терминала.

3.3.4. Использование логических имён

Логическое имя - это имя, которое можно использовать как любую часть спецификации имени файла.

Оно используется для:

- Уменьшения объема набивки и улучшения читаемости (используется для замены длинных спецификаций файлов)
- Достижения независимости от имен устройств и файлов в программах и командных процедурах
- Передачи данных в программу или командную процедуру

Логические имена и эквивалентные им строки могут иметь длины до 255 символов. (в т.ч. арифметические знаки, знаки доллара и подчеркивания).

Эквивалентные строки могут быть:

- Именем устройства
- Спецификацией файла
- Другим логическим именем
- Произвольной строкой символов

Логические имена хранятся в таблицах логических имен

Вам может понадобиться:

Добавить или изменить назначение логических имен для Вашего процесса

Определить эквивалентность логических имен

Просмотреть содержимое доступных Вам таблиц логических имен

Удалить логическое имя

Нижеследующие разделы рассматривают каждую из этих операций.

3.3.4.1. Создание логических имен

Существуют две команды DCL, которые позволяют создать логическое имя. Они решают одну задачу, но имеют разный синтаксис. Таблица 3-14 описывает эти команды.

Таблица 3-14. Команды используемые для создания логических имен

Команда	Формат и пример
ASSIGN	\$ ASSIGN эквивалентное имя[,...] логическое имя[:]
DEFINE	\$ ASSIGN DUA1:[CHARLES] CHARLIE >\$ DEFINE логическое имя эквивалентное имя[,...] >\$ DEFINE CHARLIE DUA1:[CHARLES]

3.3.4.1.1. Использование логических имен для представления имени каталога

Пример 3-19 показывает как использовать логическое имя для представления спецификации каталога.

Пример 3-19. Использование логического имени для представления спецификации каталога

```
❶ $ ASSIGN WORK1:[ROUNDS.TRAVEL] TRAVEL
❷ $ ASSIGN WORK1:[ROUNDS.EQUIPMENT] EQUIP
❸ $ SET DEFAULT TRAVEL
❹ $ SHOW DEFAULT
WORK1:[ROUNDS.TRAVEL]
❺ $ DIRECTORY EQUIP
```

Directory WORK1:[ROUNDS.EQUIPMENT]

```
EQUIPMENT_REQUEST.07MAR91;1          INVENTORY.DATA;1
REQUISITION.FORM;1          SUPPLIES.MEMO;1 SUPPLIES.MEMO1;1
SUPPLIES.MEMO2;1
```

Total of 6 files.

```

❹ $ TYPE EQUIP:REQUISITION.FORM
EQUIPMENT REQUEST FORM
.....
Name:_____ Cost Center:_____ Badge:_____
.
.
.

```

Пример 3-19, замечания:

- ❶ Присвоение логического имени подкаталогу.
- ❷ Присвоение логического имени другому подкаталогу.
- ❸ Установка каталога по умолчанию с использованием логического имени вместо непосредственного указания спецификации каталога.
- ❹ Каталог по умолчанию изменен.
- ❺ Обращение к подкаталогу по логическому имени.
- ❻ Когда логическое имя является частью спецификации файла, оно отделяется от остальной части двоеточием (:).

3.3.4.1.2. Использование логического имени для представления спецификации файла

Пример 3-20 показывает как создать логическое имя для обращения к часто используемому файлу.

Пример 3-20. Использование логического имени для представления спецификации файла.

```

❶ $ SHOW DEFAULT
WORK9:[ROUNDS]
$
❷$ DIRECTORY STAT
%DIRECT-W-NOFILES, no files found
$
❸ >$ ASSIGN WORK1:[ROUNDS.PROJECT]PROJECT_STATUS.TEMPLATE STAT
$
❹ $ EDIT/TPU STAT
.
.
.
WORK1:[ROUNDS.PROJECT]PROJECT_STATUS.TEMPLATE;1 38 lines
$
❺ $ DIRECTORY STAT
Directory WORK1:[ROUNDS.PROJECT]
PROJECT_STATUS.TEMPLATE;1
Total of 1 file.
$
❻ $ PRINT STAT
Job PROJECT_STATUS (queue SYS$PRINT, entry 635) started on LPA0
$

```

Пример 3-20, замечания:

- ❶ Выводим каталог по умолчанию. Это не тот каталог и устройство где создан файл.
- ❷ Проверяем есть ли файл с именем STAT.
- ❸ Присваиваем логическое имя STAT строке которая включает устройство, каталог, имя и тип файла.

- ④ Редактируем файл, используя для обращения логическое имя вместо прямого указания спецификации файла.
- ⑤ Просматриваем информацию о новом файле, обращаясь к нему по логическому имени STAT.
- ⑥ Используем логическое имя в команде PRINT для его вывода на системную печать.

Используя логическое имя STAT нет необходимости указывать полную спецификацию файла для доступа к нему. Кроме того оно не зависит от устройства и каталога принятого по умолчанию.

3.3.4.1.3. Использование логических имен в утилите MAIL

Для облегчения работы с утилитой MAIL можно использовать логические имена.

Присвоение логических имен часто употребляемым длинным именам узлов или именам пользователей.

- Уменьшает объем набивки необходимой при обмене сообщениями.
- При изменении имени узла пользователя можно изменить строку назначения и продолжить посылку сообщений используя то же логическое имя.

Присвоение логических имен спецификациям файлов списков рассылки.

- Можно указать список рассылки его логическим именем и не заботится о его положении на устройстве/каталоге
- Если существует много списков рассылки их можно хранить в одном каталоге.

Пример 3-21 показывает несколько способов использования логических имен в утилите MAIL.

Пример 3-21. Использование логических имен в утилите MAIL.

```
① $ ASSIGN SLOPPY::PICCADACI CP
② $ ASSIGN WORK3:[ROUNDS.LISTS]USER_PROJECT.DIS USER
③ $ ASSIGN "BUNNELL,DUFFY,MARSH,REGNELL,ROUNDS" FRIENDS
$
④ $ MAIL
MAIL> SEND
⑤ To:      CP
    .
    .
MAIL> SEND
⑥ To:      @USER
    .
    .
MAIL> SEND
⑦ To:      FRIENDS
    .
    .
```

Пример 3-21, замечания:

- ① Присвоение логического имени "CP" строке имени узла/пользователя
- ② Присвоение логического имени "USER" спецификации файла списка рассылки.
Обратите внимание на то, что одно логическое имя является частью другого.
- ③ Присвоение логического имени "FRIENDS" списку имен пользователей.

Это позволяет использовать значение логического имени, как список рассылки. Для того, чтобы включить в эквивалентную логическому имени строку запятое (,), надо заключить ее в кавычки при присвоении логического имени.

④ Обращение к утилите MAIL.

⑤ Посылка почтового сообщения пользователю SLOPPY::PICCADACI.

Использование логического имени уменьшает объем набивки.

⑥ Посылка почтового сообщения пользователям указанным в списке рассылки с именем WORK3:[ROUNDS.LISTS]USER_PROJECT.DIS.

Использование логического имени определяющего список рассылки дает возможность не вводить полную спецификацию файла.

⑦ Пыслаем сообщение пяти пользователям.

3.3.4.2. Трансляция логических имен

После того как логическое имя присвоено, может возникнуть необходимость посмотреть его значение. Таблица 3-15 описывает две команды которые могут быть использованы для определения соответствия логических имен.

Таблица 3-15. Команды используемые для определения соответствия логических имен

Команда	Коментарий и формат
SHOW LOGICAL	Итерационно раскрывает логическое имя до 10 уровней вложенности, или пока не будет исчерпана возможность раскрытия.
SHOW TRANSLATION	\$ SHOW LOGICAL логическое имя Выводит первое найденное значение эквивалентной строки. Итерации не производятся. \$ SHOW TRANSLATION логическое имя

Пример 3-22 показывает различие между командами SHOW LOGICAL и SHOW TRANSLATION.

Пример 3-22. Определение соответствия логических имен

```

① $ ASSIGN DJA0: DISK1
   $
② $ ASSIGN DISK1: PAYROLL
   $
③ $ SHOW TRANSLATION PAYROLL
   PAYROLL = "DISK1:" (LNM$PROCESS_TABLE)
   $
④ $ SHOW LOGICAL PAYROLL
   "PAYROLL" = "DISK1:" (LNM$PROCESS_TABLE)
   1 "DISK1" = "DJA0:" (LNM$PROCESS_TABLE)
   $
```

Пример 3-22, замечания:

- ① Создается логическое имя DISK1, которое соответствует строке "DJA0:".
- ② Создается другое логическое имя PAYROLL, эквивалентное строке "DISK1:".
- ③ При использовании команды SHOW TRANSLATION для определения соответствия имени производится только одно раскрытие.
- ④ При использовании команды SHOW LOGICAL для определения соответствия имени PAYROLL, производится итеративное раскрытие, пока будет возможность его провести.

3.3.4.3. Удаление логических имен

Значение логического имени процесса существует:

До выхода из системы

До присвоения логическому имени нового значения

До удаления логического имени командой DEASSIGN

Для того чтобы постоянно (на время работы в системе) присвоить логическое имя, присвоение надо указать в специальном командном файле, исполняемом при входе в систему и, как правило, имеющим имя LOGIN.COM.

Для отмены значения логического имени заданного командами

ASSIGN или DEFINE используется команда DEASSIGN.

Syntax:

\$ DEASSIGN логическое имя

Можно использовать квалификатор /ALL для отмены всех логических имен процесса:

\$ DEASSIGN/ALL

Пример 3-23. Отмена логических имен

```
❶ $ ASSIGN WORK3:[ROUNDS.COMMANDS] COMMANDS
$
❷ $ SHOW LOGICAL COMMANDS
"COMMANDS" = "WORK3:[ROUNDS.COMMANDS]" (LNM$PROCESS_TABLE)
$
❸ $ DEASSIGN COMMANDS
$
❹ $ SHOW LOGICAL COMMANDS
%SHOW-S-NOTRAN, no translation for logical name COMMANDS
$
```

Пример 3-23, замечания:

- ❶ Присвоение логического имени COMMANDS для определения подкаталога.
- ❷ Просмотр значения логического имени COMMANDS.
- ❸ Отмена логического имени.
- ❹ Логическое имя COMMANDS больше не существует.

3.3.4.4. Логические имена создаваемые системой

Постоянные для процесса логические имена создаются командным интерпретатором при входе в систему и действуют на протяжении всего процесса.

Эти логические имена нельзя отменить.

Их можно переопределить указав то же логическое имя в командах ASSIGN или DEFINE, но если это имя потом отменяется, то восстанавливается его значение, определенное при входе.

Таблица 3-16. Логические имена постоянные для процесса

SYS\$INPUT	Принятый по умолчанию файл из которого DCL осуществляет ввод информации.
SYS\$OUTPUT	Принятый по умолчанию файл (как правило терминал пользователя) на который DCL выводит информацию.

SY\$\$ERROR	Принятый по умолчанию файл в который DCL выводит сообщения генерируемые при ошибках и предупреждениях.
SY\$\$COMMAND	Начальный файл (как правило терминал) из которого DCL осуществляет ввод команд. (DCL использует это логическое имя для запоминания начального входного потока.)

Используя команду SHOW LOGICAL/PROCESS можно вывести информацию о назначениях постоянных для процесса логических имен.

Пример 3-24. Логические имена процесса

```
$ SHOW LOGICAL/PROCESS
```

```
(LNM$PROCESS_TABLE)
```

```
"SY$$COMMAND" = "_BROWNY$RTA2:"
"SY$$DISK" = "$1$DUA1:"
"SY$$ERROR" = "_BROWNY$RTA2:"
"SY$$INPUT" = "_BROWNY$RTA2:"
"SY$$OUTPUT" [super] = "_BROWNY$RTA2:"
"SY$$OUTPUT" [exec] = "_BROWNY$RTA2:" "TT" = "RTA2:"
```

3.3.4.4.1. Использование постоянных логических имен процесса для переназначения выходного потока

Иногда полезно сохранить выводимую информацию для последующего просмотра. Можно переназначить вывод на терминал или в файл используя команды ASSIGN и DEFINE, задав новое значение логическому имени соответствующему терминалу.

Формат:

```
$ ASSIGN output_file SY$$OUTPUT
```

```
$ DEFINE SY$$OUTPUT output_file
```

Примеры команд:

```
$ ASSIGN STATS.DAT SY$$OUTPUT
```

```
$ DEFINE SY$$OUTPUT STATS.DAT
```

Эти присвоения будут действовать до выхода из системы или использования команды DEASSIGN.

Можно проделать временное назначение (на исполнение одной команды) добавив квалификатор /USER в командную строку:

```
$ ASSIGN/USER STATS.DAT SY$$OUTPUT
```

```
$ DEFINE/USER SY$$OUTPUT STATS.DAT
```

Это присвоение будет действовать только на время исполнения одного исполняемого образа (программы)

Пример 3-25 показывает как назначить поток вывода в файл. Этот файл может быть позднее отредактирован, отправлен на печать, отправлен по почте другому пользователю, и т.д.

Пример 3-25. Переназначение вывода в файл

```

❶ $ SHOW PROCESS
10-DEC-1990 09:26:17.18 User: ROUNDS      Process ID:      23E00245
                        Node: SLOPPY      Process name:    "ROUNDS"

Terminal:              RTA2: (SUPER::ROUNDS)
User Identifier:       [VMS,ROUNDS]
Base priority:        4
Default file spec:    $1$DUA1:[ROUNDS]

Devices allocated:    SLOPPY$RTA2:
$
❷ $ ASSIGN PROCESS.TXT SYS$OUTPUT
❸ $ SHOW PROCESS
❹ $ DEASSIGN SYS$OUTPUT
❺ $ TYPE PROCESS.TXT
10-DEC-1990 09:26:50.98 User: ROUNDS      Process ID:      23E00245
                        Node: SLOPPY      Process name:    "ROUNDS"

Terminal:              RTA2: (SUPER::ROUNDS)
User Identifier:       [VMS,ROUNDS]
Base priority:        4
Default file spec:    $1$DUA1:[ROUNDS]

Devices allocated:    SLOPPY$RTA2:
$

```

Пример 3-25, замечания:

- ❶ Вывод команды SHOW PROCESS на терминал.
- ❷ Переназначение логического имени на файл с именем PROCESS.TXT.
В этом примере использована команда ASSIGN
Для использования команды DEFINE необходимо параметры поменять местами:
\$ DEFINE SYS\$OUTPUT PROCESS.TXT
Теперь при вводе команды SHOW PROCESS вывод будет осуществляться в файл, а не на терминал.
- ❸ Отменяем назначение SYS\$OUTPUT и далее весь вывод идет на терминал
- ❹ PROCESS.TXT содержит информацию выведенную командой SHOW PROCESS.

3.3.4.4.2. Другие логические имена назначаемые системой

При входе в систему ОС VMS создает также и другие логические имена, которые могут оказаться полезными в работе. Таблица 3-17 показывает два таких логических имени.

Таблица 3-17. Другие логические имена создаваемые системой

Логическое имя	Описание
SYS\$LOGIN	Устройство и каталог, устанавливаемые при входе в систему, как корневого каталог пользователя.
SYS\$LOGIN_DEVICE	Часть SYS\$LOGIN определяющая имя устройства.

Если при создании собственных логических имен использовать имена создаваемые системой, то не придется указывать необходимое устройство или каталог.

Если менеджер системы переместит Вашу структуру каталогов на другой диск, то логические имена можно будет использовать без изменений.

Если пользователь работает более чем на одной ЭВМ, то ему не надо запоминать его принятое по умолчанию устройство для каждой системы.

Вне зависимости от текущего каталога и устройства легко вернуться в исходную точку установив текущим каталог SYS\$LOGIN.

Пример 3-26 показывает использование логических имен SYS\$LOGIN и SYS\$LOGIN_DEVICE.

Пример 3-26. Использование создаваемых системой логических имен

```
❶ $ SET DEFAULT $1$DUA9:[VMS$COM.VMS_CURRICULUM.ART]
$ SHOW DEFAULT
    $1$DUA9:[VMS$COM.VMS_CURRICULUM.ART]
$
❷ $ DIRECTORY SYS$LOGIN

Directory WORK3:[ROUNDS]

ADD.BAS;3          ADD.EXE;3          AETNA.DIR;1          ANNOUNCE.TXT;3
.
.
.
Total of 125 files.
$
❸ $ ASSIGN SYS$LOGIN_DEVICE:[ROUNDS.COMMANDS] COMMANDS
$
❹ $ DIRECTORY COMMANDS

Directory WORK3:[ROUNDS.COMMANDS]

241COLORS.COM;2 BATV10.COM;4          CHECK_CMS.COM;1 COLOR.COM;2
.
.
.
Total of 43 files.
$
❺ $ SET DEFAULT SYS$LOGIN
$      SHOW DEFAULT
      WORK3:[ROUNDS]
$
```

Пример 3-26, замечания:

- ❶ Изменяем принятые по умолчанию каталог и устройство и просматриваем их.
- ❷ Указываем начальные каталог и устройство используя логическое имя SYS\$LOGIN.
- ❸ Используем логическое имя SYS\$LOGIN_DEVICE для представления принятого по умолчанию устройства и присваиваем логическое имя подкаталогу.
- ❹ Используем новое логическое имя для ссылки на подкаталог.
- ❺ Устанавливаем логическое имя SYS\$LOGIN для установки принятых по умолчанию каталога и устройства и просматриваем их.

3.3.5. Использование логических символов

Командные процедуры могут хранить данные и манипулировать ими используя *логические символы*.

Логический символ это идентификатор которому ставится в соответствие числовая, текстовая или логическая величина.

При использовании идентификатора логического символа в командной строке DCL заменяет его на присвоенное значение.

Логические символы могут использоваться как:

- Синонимы команд DCL
- Параметры передаваемые в командные процедуры

- Переменные внутри командных процедур

3.3.5.1. Типы логических символов

Доступность значения присваиваемого логическому символу можно сделать локальной или глобальной для командного интерпретатора:

Локальный логический символ:

- Существует только во время исполнения командной процедуры которая его создает, а также во время исполнения любой командной процедуры вызываемой из нее.
- Прекращает существовать при выходе из создавшей его командной процедуры.

Глобальный логический символ:

- Существует на протяжении всего процесса, в т.ч. и после того как создавшая его командная процедура завершилась
- Используется внутри командной процедуры и продолжает использоваться после ее завершения

3.3.5.2. Непосредственное создание логических символов

Можно создать логический символ, используя один или два знака равенства (=) для присвоения значения идентификатору логического символа. Используемый синтаксис зависит от типа данных (текстовый или числовой) и области значимости символа (локальный или глобальный).

Таблица 3-18 показывает как создавать такие типы логических символов.

Таблица 3-18. Присвоение значений логическим символам

Операция	Оператор	Пример
Присвоение текстовой строки локальному логическому символу	= или :=	USER = "SMITH" USER := SMITH
Присвоение текстовой строки локальному глобальному символу	== или :==	GROUP == "Project Group" GROUP :== Project Group
Присвоение целого числа локальному логическому символу	=	COUNT = 1
Присвоение целого числа глобальному логическому символу	==	NEWCOUNT == 100

3.3.5.3. Использование логических символов как синонимов команд

Приравнивание логическому символу командной строки позволяет выполнить эту команду набрав на клавиатуре имя логического символа. Например, можно создать логический символ для часто используемых и длинных команд:

```
$ HOME == "SET DEFAULT SYS$LOGIN:"
$ PN == "PRINT/NOTIFY"
$ UP == "SET DEFAULT [-]"
```

3.3.5.4. Сокращение имен логических символов

Допускается использование сокращенных форм логических символов, если при их определении использовать символ звездочка (*). Следующий пример показывает как создать локальный логический символ который можно использовать в сокращенной форме:

```
$ CL*EANUP = "PURGE/LOG"
```

После того, как этот символ создан следующие обращения вызовут исполнение команды PURGE/LOG:

```
$ CL
```

```
$ CLE
```

```
$ CLEA
```

```
$ CLEAN
```

```
$ CLEANU
```

```
$ CLEANUP
```

3.3.5.5. Просмотр значений и удаление логических символов

В процессе работы может возникнуть необходимость просмотреть значения одного или нескольких логических символов или удалить их.

Таблица 3-19 показывает команды DCL используемые для этого.

Таблица 3-19. Просмотр значений и удаление логических символов

Операция	Командная строка/Пример	Пояснение
Просмотр значения логического символа	\$ SHOW SYMBOL[/GLOBAL] логический символ	
Просмотр значение одного глобального логического символа	\$ SHOW SYMBOL/GLOBAL SD	Выводит значение глобального логического символа SD
Просмотр значение одного локального логического символа	\$ SHOW SYMBOL X	Выводит значение локального логического символа X
Просмотр значений всех глобальных логических символов	\$ SHOW SYMBOL/GLOBAL/ALL	Выводит значения всех логических символов определенных в таблице глобальных логических символов
Просмотр значений всех локальных логических символов	\$ SHOW SYMBOL/ALL	Выводит значения всех логических символов определенных в таблице локальных логических символов
Просмотр значений всех глобальных логических символов с использованием	\$ SHOW SYMBOL/GLOBAL S* C	Выводит значения всех логических символов определенных в таблице глобальных

Операция	Командная строка/Пример	Пояснение
шаблона		символов начинающихся с буквы "S"
Просмотр значений всех локальных логических символов с использованием шаблона	\$ SHOW SYMBOL S*	Выводит значения всех логических символов определенных в таблице локальных символов начинающихся с буквы "S"
Удаление логических символов	\$ DELETE/SYMBOL[/GLOBAL] логический символ	
Удаление глобального логического символа	\$ DELETE/SYMBOL/GLOBAL SD	Удаляет логический символ SD из таблицы глобальных символов
Удаление локального логического символа	\$ DELETE/SYMBOL X	Удаляет логический символ X из таблицы локальных символов
Удаление глобальных логических символов	всех \$ DELETE/SYMBOL/GLOBAL/ALL	Удаляет все логические символы из таблицы глобальных символов
Удаление локальных логических символов	всех \$ DELETE/SYMBOL/ALL	Удаляет все логические символы из таблицы локальных символов

3.3.6. Создание определений для дополнительной клавиатуры

В процессе работы с ОС VMS у пользователя появляется набор наиболее употребимых команд. Для того, чтобы сократить время затрачиваемое на их набивку можно воспользоваться командой **DEFINE/KEY**, которая позволяет задать определения клавишам дополнительной клавиатуры. Эти определения как правило содержат команды DCL или их часть.

Клавиши дополнительной клавиатуры PF1 - PF4 всегда доступны для программирования.

Перечисленные клавиши должны быть специально разрешены для программирования:

- от KP0 до KP9
- от F6 до F20 (если они есть на терминале)
- ЗАПЯТАЯ (COMMA)
- ТОЧКА (PERIOD)
- ВВОД (ENTER)

- МИНУС (MINUS)

Использование этих клавиш для программирования разрешается командами:

```
$ SET TERMINAL/APPLICATION_KEYPAD
```

```
$ SET TERMINAL/NUMERIC
```

Эти определения клавиш доступны только на уровне командного интерпретатора DCL. Они не действуют в текстовых редакторах, утилите Mail и т.д.

3.3.6.1. Использование команды **DEFINE/KEY**

Команда **DEFINE/KEY** используется для настройки определений дополнительной клавиатуры на уровне командного интерпретатора DCL.

Формат:

\$ DEFINE/KEY имя-клавиши эквивалентная-строка/квалификаторы

При программировании дополнительной клавиатуры можно использовать следующие квалификаторы команды:

/TERMINATE	Выполняет автоматический возврат каретки.
/NOECHO	Подавляет вывод текста исполняемой команды на экран.
/ERASE	Удаляет символы из текущей строки перед выводом и исполнением команды вызываемой нажатой клавишей дополнительной клавиатуры.
/NOLOG	Подавляет вывод информационного сообщения системы при определении клавиши.

Пример 3-27. Команда **DEFINE/KEY**

- ❶ \$ SET TERMINAL/NUMERIC
- ❷ \$ DEFINE/KEY COMMA "RECALL/ALL"/TERMINATE
- ❸ %DCL-I-DEFKEY, DEFAULT key COMMA has been defined

Пример 3-27 , замечания:

- ❶ Разрешение программирования клавиш.
SET TERMINAL/APPLICATION_KEYPAD выполняет тоже самое
- ❷ Определения клавиши <box>(COMMA) так, что при каждом нажатии в ответ на подсказку DCL выполняется команда RECALL/ALL
Квалификатор /TERMINATE означает, что символ возврат каретки присоединяется к заданной строке и не надо нажимать вручную клавишу RETURN.
- ❸ Обратите внимание на информационное сообщение системы при определении клавиши.
Оно подавляется квалификатором /NOLOG.

Пример 3-28. Пример кода команды **DEFINE/KEY**

- ❶ \$ DEFINE/KEY PF1 "SET TERMINAL/NUMERIC"/TERMINATE/NOLOG/ECHO
- ❷ \$ DEFINE/KEY PF2 "SHOW KEY/FULL/ALL"/TERMINATE/NOLOG/ECHO
- ❸ \$ DEFINE/KEY COMMA "RECALL/ALL"/TERMINATE/NOLOG/NOECHO

Пример 3-28, замечания:

- ❶ Определение клавиши <PF1>. При нажатии на нее дополнительная клавиатура будет разрешена для программирования.
При нажатии на <PF1> автоматически прибавляется <RETURN>.
Информационное сообщение системы не выводится при определении.

При нажатии <PF1>, заданная командная строка выводится на экран терминала.

- ❷ При нажатии на <PF2> выводится текущий список назначений клавиш.
- ❸ Нажатие клавиши <COMMA> вызывает вывод содержимого буфера команд на экран. Сама команда RECALL/ALL на экран не выводится.

3.3.6.2. Просмотр определений клавиш

Команда SHOW KEY используется для вывода на экран определений клавиш дополнительной клавиатуры.

Пример 3-29 иллюстрирует использование этой команды.

Пример 3-29. Использование команды SHOW KEY

- ❶ >\$ SHOW KEY
_Key name: PF1
DEFAULT keypad definitions:
PF1 = "SET TERMINAL/NONUMERIC"
\$
- ❷ \$ SHOW KEY PF1
DEFAULT keypad definitions:
PF1 = "SET TERMINAL/NONUMERIC"
\$
- ❸ \$ SHOW KEY/ALL
DEFAULT keypad definitions:
COMMA = "RECALL/ALL"
PF1 = "SET TERMINAL/NONUMERIC"
PF2 = "SHOW KEY/FULL/ALL"
\$
- ❹ \$ SHOW KEY/FULL/ALL
DEFAULT keypad definitions:
COMMA = "RECALL/ALL" (noecho,terminate,noerase,nolock)
PF1 = "SET TERMINAL/NONUMERIC" (echo,terminate,noerase,nolock)
PF2 = "SHOW KEY/FULL/ALL" (echo,terminate,noerase,nolock)

Пример 3-29, замечания:

- ❶ Запрос вывода определения клавиши. Если имя клавиши не задано, то выводится подсказка для ввода.
- ❷ Запрос вывода определения клавиши. Имя клавиши указывается.
- ❸ Запрос вывода текущего списка определений клавиш.
- ❹ Запрос вывода текущего списка определений клавиш, включая квалификаторы, соответствующие каждому определению.

3.3.6.3. Использование определений клавиш

Пример 3-30. Использование определений клавиш

- ❶ \$ DEFINE/KEY PF1 "SET TERMINAL/NONUMERIC"/TERMINATE/NOLOG/ECHO
\$ DEFINE/KEY PF2 "SHOW KEY/FULL/ALL"/TERMINATE/NOLOG/ECHO
\$ DEFINE/KEY COMMA "RECALL/ALL"/TERMINATE/NOLOG/NOECHO
\$
- ❷ \$ SET TERMINAL/NONUMERIC
\$
- ❸ \$ SHOW KEY/FULL/ALL
DEFAULT keypad definitions:
PF1 = "SET TERMINAL/NONUMERIC" (echo,terminate,noerase,nolock)
PF2 = "SHOW KEY/FULL/ALL" (echo,terminate,noerase,nolock)
COMMA = "RECALL/ALL" (noecho,terminate,noerase,nolock)
\$

```

❶ $
    1 SHOW KEY/FULL/ALL
    2 SET TERMINAL/NUMERIC
    3 DEFINE/KEY COMMA "RECALL/ALL"/TERMINATE/NOLOG/NOECHO
    4 DEFINE/KEY PF2 "SHOW KEY/FULL/ALL"/TERMINATE/NOLOG/ECHO
    5 DEFINE/KEY PF1 "SET TERMINAL/NUMERIC"/TERMINATE/NOLOG/ECHO
$

```

Пример 3-30, замечания:

- ❶ Создаем определения клавиш.
- ❷ Нажимаем <PF1> для разрешения использования определений клавиш (См. ранее.)
- ❸ Нажимаем <PF2> для вывода определений клавиш.
- ❹ Нажимаем <COMMA> для просмотра командного буфера. Эта команда не выводится на экран.

3.3.6.4. Удаление определений клавиш

Определения клавиш будут существовать до выхода из системы или вызова команды DELETE/KEY.

Формат:

\$ DELETE/KEY [имя-клавиши]

Если имя клавиши не указано, будет выведена подсказка.

Пример 3-31. Использование команды DELETE/KEY

```

❶ $ DEFINE/KEY KP7 "SHOW USERS"
    %DCL-I-DEFKEY, DEFAULT key KP7 has been defined
$
❷ $ SHOW KEY KP7
    DEFAULT keypad definitions:
        KP7 = "SHOW USERS"
$
❸ $ DELETE/KEY
    _Key name: KP7
    %DCL-I-DELKEY, DEFAULT key KP7 has been deleted
$
❹ $ SHOW KEY KP7
    %DCL-W-UNDKEY, DEFAULT key KP7 is undefined

```

Пример 3-31, замечания:

- ❶ Определение KP7 (клавиша 7 на дополнительной клавиатуре).
- ❷ Просматриваем определение KP7.
- ❸ Удаляем определение KP7.
Если имя клавиши не указано будет выведена подсказка.
- ❹ Определения клавиши KP7 больше нет.

3.3.6.5. Основные ошибки при определении клавиш

Пример 3-32 показывает основные ошибки при определении клавиш.

Пример 3-32. Основные ошибки при определении клавиш

```
❶ $ DEFINE/KEY PF5 "DIRECTORY"
%DCL-W-IVKEYNAM, unrecognized key name - check validity and spelling
$
❷ $ DEFINE/KEY KP7 "SHOW USERS"
%DCL-I-DEFKEY, DEFAULT key KP7 has been defined
$ 7
%DCL-W-IVVERB, unrecognized command verb - check validity and spelling
\7\
$
```

Пример 3-32, замечания:

- ❶ Попытка определить несуществующую клавишу.
- ❷ Клавиша определена успешно, но определение не работает. Введите команду SET TERMINAL/NUMERIC или SET TERMINAL/APPLICATION_KEYPAD для того, чтобы разрешить определения.

3.3.7. Командные процедуры

Командная процедура это созданный обычным образом текстовый файл, содержащий команды DCL.

Его можно создать, используя текстовый редактор.

Командный интерпретатор DCL считывает из него команды аналогично терминалу.

Большинство из того, что можно сделать с терминала можно сделать и из командной процедуры.

Командная процедура исполняемая при входе в систему может быть использована для установки характеристик программной среды процесса при каждом входе в систему:

Может использоваться для:

- Назначения логических имен
- Установки характеристик терминала
- Создания определений клавиш

Обычно имеет имя LOGIN.COM.

Обычно находится в корневом каталоге пользователя.

3.3.7.1. Создание командных процедур

Для создания командной процедуры надо выполнить следующие действия:

1. Спроектировать командную процедуру.

Определить решаемую задачу.

Определить результат исполнения процедуры.

2. Создать файл командной процедуры.

Воспользоваться текстовым редактором.

Присвоить файлу имя и тип COM.

3. Выполнить и протестировать командную процедуру.

Для запуска используется символ "коммерческое эт" (@).

Для вывода на экран каждой исполняемой строки процедуры используется команда DCL SET VERIFY, это поможет определить место возможной ошибки.

Существует два способа разрешить верификацию (контроль):

- Ввести команду SET VERIFY перед выполнением командной процедуры.
- Добавить команду SET VERIFY в начало файла командной процедуры.

Верификация остается до выхода из системы или ввода команды.

4. Если необходимо внесите исправления в командную процедуру и повторно оттестируйте повторяя шаги 2 и 3.
5. Добавьте комментарии для упрощения дальнейшего использования и обслуживания командной процедуры. Комментарии могут включать:

Имя процедуры и ее автора.

Описание действий выполняемых процедурой, включая типы вводимых данных.

Описания специфических команд включенных в процедуру.

3.3.7.2. Правила

При написании командных процедур выполняйте следующие правила:

Начинайте каждую команду DCL с новой строки.

Каждая команда DCL должна начинаться со знака доллара (\$).

Для продолжения команды на новой строке используйте перенос (-) (знак доллара в строке продолжения не ставится).

Комментарии начинаются со знака доллара и восклицательного знака (\$!).

3.3.7.3. Рекомендации

В дополнение к основным правилам построения командных процедур согласованный (типовой) формат делает их более удобными для чтения, тестирования и обслуживания.

Присваивайте тип файлов COM для всех командных процедур.

Используйте полную запись команд DCL, а не сокращения.

Для удобочитаемости используйте "пустые" командные строки для отделения блоков команд.

Заканчивайте выход из командной процедуры командой EXIT.

Пример 3-33 показывает использование этих рекомендаций.

Пример 3-33. Пример командной процедуры

```
$                                ! L O G I N . C O M
$
$ ASSIGN SYS$LOGIN_DEVICE:[ROUNDS.TRAVEL] TRAVEL
$ ASSIGN SYS$LOGIN_DEVICE:[ROUNDS.EQUIPMENT] EQUIP
$
$ DA == "DIRECTORY/DATE"          ! List creation dates of files.
$ DEL*ETE == "DELETE/LOG"         ! Change action of the DELETE command.
$ HOME == "SET DEFAULT SYS$LOGIN:" ! Set default to top-level directory. $
$ DEFINE/KEY PF1 "SET TERMINAL/NONUMERIC"/TERMINATE/NOLOG/ECHO
$ DEFINE/KEY PF2 "SHOW KEY/FULL/ALL"/TERMINATE/NOLOG/ECHO
$ DEFINE/KEY COMMA "RECALL/ALL"/TERMINATE/NOLOG/NOECHO
$
$ EXIT
```

3.3.7.4. Отладка командных процедур

Для тестирования командной процедуры необходимо выполнить следующие действия:

1. Исполнить командную процедуру:

Введите команду исполнения (@) и имя процедуры.

```
$ @LOGIN.COM
```

или

```
$ @LOGIN
```

Интерпретатор DCL читает строки из командной процедуры.

Операционная система не реагирует на нажатия клавиш во время исполнения процедуры.

2. Определите как работает процедура.

3. Найдите и устраните ошибки в процедуре:

Введите команду SET VERIFY.

Выполните командную процедуру.

Исполняемые командные строки будут выводиться на экран.

Внесите исправления.

Если процедура работает без ошибок, то введите команду SET NOVERIFY.

3.3.7.5. Прерывание и продолжение исполнения командной процедуры

Пользователь может прервать исполнение командной процедуры, используя клавиатуру терминала, а затем продолжить ее исполнение.

Нажатие CTRL/Y или CTRL/C приводит к прерыванию исполнения процедуры.

- Система отображает на экран [Interrupt]
- Некоторые программы выполняют специальные действия при нажатии CTRL/C.

Ввод команды CONTINUE продолжает исполнение процедуры.

3.3.8. Командная процедура, исполняемая при входе в систему

Командная процедура LOGIN.COM выполняется во время входа пользователя в систему.

Находится в корневом каталоге пользователя, если системный менеджер не указал иначе при регистрации пользователя.

LOGIN.COM это типовое имя файла, но системный менеджер может указать и другое.

Этот файл должен содержать команды, которые настраивают рабочую среду пользователя, включая установку характеристик терминала, определение клавиш и логических имен.

При создании или модификации файла LOGIN.COM протестируйте его до выхода из системы:

```
@LOGIN
```

Если LOGIN.COM содержит ошибки, которые не дают войти в системы, нужно воспользоваться квалификатором /NOCOMMAND при входе в систему:

```
Username: SHONE/NOCOMMAND  
Password:
```

3.3.8.1. Пример LOGIN.COM

Пример 3-34 показывает как командная процедура LOGIN.COM используется для настройки рабочей среды пользователя.

Пример 3-34. Пример файла

```

$                                ! LOGIN.COM❶
$❷
$                                ! Logical name assignments.❸
$ ASSIGN SYS$LOGIN_DEVICE:[ROUNDS.TRAVEL] TRAVEL❹
$ ASSIGN SYS$LOGIN_DEVICE:[ROUNDS.EQUIPMENT] EQUIP
$ ASSIGN SLOPPY::PICCADACI CP
$ ASSIGN TCHR::MOREAU MEG
$ ASSIGN SYS$LOGIN_DEVICE:[ROUNDS.LISTS]USER_PROJECT.DIS USER
$ ASSIGN "BUNNELL,DUFFY,MARSH,REGNELL,ROUNDS" FRIENDS
$
$                                ! Symbol definitions.
$ DA == "DIRECTORY/DATE"        ! List creation dates of files.❺
$ HOME == "SET DEFAULT SYS$LOGIN:" ! Set default to top-level directory.
$ WHE*RE == "SHOW DEFAULT"      ! Display current default.
$ TO == "SET DEFAULT"
$ UP == "SET DEFAULT [-]"
$ DEL*ETE == "DELETE/LOG"       ! Change action of the DELETE command.❻
$
$                                ! Key definitions.
>$ DEFINE/KEY PF1 "SET TERMINAL/NONUMERIC"/TERMINATE/NOLOG/ECHO❷
$ DEFINE/KEY PF2 "SHOW KEY/FULL/ALL"/TERMINATE/NOLOG/ECHO ! Show key
definitions.
$ DEFINE/KEY PF3 "MAIL/EDIT=(SEND,REPLY,FORWARD)"/TERMINATE/NOLOG ! Edit
messages.
$ DEFINE/KEY PF4 "TYPE/PAGE NL:"/TERMINATE/NOLOG/NOECHO!Clear the screen.
$ DEFINE/KEY COMMA "RECALL/ALL"/TERMINATE/NOLOG/NOECHO ! Show recent
commands.
$ DEFINE/KEY MINUS "SET DEFAULT [-]"/TERMINATE/NOLOG/NOECHO! Up one level.
$
$ SET TERMINAL/NONUMERIC
$ SET TERMINAL/LINE_EDITING/INSERT
$
$ SET PROMPT="Ready> "❸
$
$ EXIT❾

```

Пример 3-34, замечания:

- ❶ Полезно включать в файл имя командной процедуры.
- ❷ "Пустые" строки (только со знаком доллара) используются для разбивки процедуры на секции.
- ❸ Комментарии используются для пояснения назначения секций.
- ❹ Логические имена присваиваются часто используемым каталогам и файлам.
- ❺ Логические символы создаются для часто используемых команд DCL.
- ❻ Переопределение способа исполнения команды DCL "*" определяет точку сокращения.
- ❼ Определение клавиш терминала и установка его характеристик.
- ❸ Замена подсказки DCL на отличную от "\$".
- ❾ "Правильный" выход из процедуры.

3.3.8.2. Основные ошибки в командной процедуре LOGIN.COM

Таблица 3-20 описывает основные ошибки при создании и использовании командной процедуры LOGIN.COM.

Таблица 3-20. Основные ошибки в файлах LOGIN.COM

Проблема	Причины
Пользователь не может войти в	Не содержит ли LOGIN.COM команду LOGOUT?

Проблема	Причины
систему	<code>\$ LOGOUT</code> Войдите в систему без исполнения <code>LOGIN.COM</code> и удалите команду <code>LOGOUT</code> из файла.
Введенные настройки не действуют	<code>Username: SHONE/NOCOMMAND</code> Настройки не доступны системе до исполнения <code>LOGIN.COM</code> . Выйдите и повторно войдите в систему или исполните <code>LOGIN.COM</code> введя команду:
Некоторые логические символы не работают	<code>\$ @LOGIN</code> Логические символы используемые вне <code>LOGIN.COM</code> должны быть глобальными и создаваться с использованием двух знаков равенства (<code>==</code>).
Некоторые назначения клавиш не работают	<code>\$ HOME == "SET DEFAULT SYS\$LOGIN:"</code> <code>\$ PU*RGE := PURGE/LOG</code> Надо выполнить одну из следующих команд для того, чтобы разрешить определения клавиш дополнительной клавиатуры:
Некоторые команды DCL работают не так, как должны	<code>\$ SET TERMINAL/NUMERIC</code> <code>\$SET TERMINAL/APPLICATION_KEYPAD</code> Не созданы ли логические символы похожие на команды DCL? Логический символ подавляет действие команды DCL:
Некоторые настройки работают, а некоторые нет	<code>\$ DEL*ETE == "LOGOUT"</code> <code>\$ DELETE</code> ROUNDS logged out at 21-JUN-1991 12:46:31.20 Исполнение файла <code>LOGIN.COM</code> было прервано по причине ошибки и часть команд не исполнилась. Введите команду <code>SET VERIFY</code> и протестируйте файл <code>LOGIN.COM</code> , проверив каждую команду и определив место где исполнение прекращается.

3.3.9. ВЫВОДЫ

Логические имена используются вместо спецификации файла или ее части.

- Для создания логических имен используются команды `ASSIGN` или `DEFINE`.
- Для просмотра значений логических имен используются команды `SHOW LOGICAL` или `SHOW TRANSLATION`.
- Команда `DEASSIGN` используется для удаления (отмены) логических имен.

Логические символы создаются как синонимы часто используемых команд.

- Для присвоения логическому символу значения используется один или два знака равенства.
- Команда `SHOW SYMBOL` выводит значение логического символа.
- Команда `DELETE/SYMBOL` удаляет символ.

Можно уменьшить объем набивки, задавая с помощью команды `DEFINE/KEY` определения клавишам дополнительной клавиатуры.

Для установки логических имен, символов, определения клавиш дополнительной клавиатуры и установки других характеристик терминального сеанса при каждом входе в систему используется командная процедура LOGIN.COM.

3.4. Письменные упражнения

3.4.1. Терминология входа в систему

Поставьте в соответствие определению слева выражение справа. Выражения могут быть использованы более одного раза, и некоторые определения могут соответствовать более, чем одному выражению.

Определения	Выражения
___ Обязательная информация, которая должна быть введена пользователем во время процедуры входа в систему.	a. BREAK b. CONNECT
___ Требуется, если Вы хотите иметь болееодной терминальной сессии в действии одновременно.	c. Подсказка DCL d. Пароль
___ Команда, используемая, когда доступ к системе осуществляется через терминальный сервер	e. Сессия f. SET FORWARD
___ \$	g. SET HOST
___ Команда, используемая после подсказки терминального сервера для вывода на экран числа сессий	h. SET PASSWORD
___ Команда, используемая для изменения пароля	i. SHOW SESSIONS
___ Команда, используемая для доступа к другой системе в компьютерной сети	j. Терминальный сервер k. Имя пользователя
___ Команда для инициализации доступа к системе через терминальный сервер	
___ Используется для возврата к подсказке терминального сервера без выхода из системы	
___ Слово или предложение, не отображаемое на терминале во время набора, требующееся для доступа к системе	

3.4.2. Компоненты команды DCL

Поставьте в соответствие компонентам командной строки слева определения справа.

Командная строка:

\$ TYPE MYFILE.001/PAGE

Компонент	Определение
___ \$	a. Разделитель команды
___ TYPE	b. Параметр команды
___ (пробел)	c. Команда DCL
___ MYFILE.001	d. Подсказка DCL
___ /	e. Квалификатор

Компонент	Определение
— PAGE	f. Разделитель квалификатора

3.4.3. Клавиши терминала

Подберите букву, обозначающую клавишу или комбинацию клавиш, которая лучше выполняет каждую из операций в левой колонке. Вы можете не использовать все буквы из списка.

Операции	Специальные функциональные клавиши
— Вы вошли в систему. Длинная череда сообщений, все из которых вы уже раньше видели, движется по Вашему экрану. Подавите эти сообщения, не останавливая и не сбрасывая программу, которая их производит.	a. CTRL/B b. CTRL/O c. CTRL/Q
— Вы только что набрали символы TYPE FILE&. Курсор находится сразу за символом &. Удалите этот символ.	d. CTRL/R e. CTRL/S
— Вы ввели команду SHOW SYSTEM. Список процессов в Вашей системе ползет по Вашему экрану. Прекратите дальнейшее выполнение команды и верните управление своему терминалу.	f. CTRL/U g. CTRL/Y h. DEL или RUBOUT
— Вы ввели следующую строку на Вашем терминале: \$ SHOW PROCESS/ALL	i. RETURN
Строки информации двигаются по экрану Вашего терминала. Остановите вывод и остановите, но не сбрасывайте программу, которая его генерирует.	
— Возобновите генерацию выходной информации, которую вы остановили в предыдущей операции.	
— Вы сделали обширные исправления в командной строке на печатающем терминале. В результате получилось следующее: \$ PRYNT \ TNY\INT9\9\ FILIN\NI\	
Выведите эту строку без отображения исправлений.	
— Вы только что ввели командную строку. Вызовите эту команду повторно.	

3.4.4. Системные сообщения

Поставьте в соответствие компонентам системного сообщения об ошибке слева их определения справа.

Сообщение об ошибке:

```
$ MAKE
%DCL-W-IVVERB, unrecognized command verb - check validity and spelling
\MAKE\
```

Компоненты	Определения
— %	a. Код средства
— DCL	b. Код идентификации сообщения

Компоненты	Определения
— W	c. Текст сообщения
— IVERB	d. Код уровня серьезности
— unrecognized...	e. Разделитель системного сообщения

3.5. Лабораторные упражнения

3.5.1. Вход и выход из системы OpenVMS

3.5.1.1. Пароли

Выполните следующие действия за интерактивным терминалом.

1. Попробуйте войти в систему, используя имя, отличное от Вашего, и Ваш пароль.
Наблюдайте, что происходит.
2. Теперь попробуйте войти в систему с использованием Вашего имени и неверного пароля.
Продолжайте попытки с неправильным паролем.
Что происходит?
3. Войдите в систему, используя правильное имя пользователя и пароль.
4. Измените свой пароль, используя команду SET PASSWORD/GENERATE.
5. Измените свой пароль, используя команду SET PASSWORD.
6. Выйдите из системы.

3.5.1.2. Терминальный сервер

Замечание

Следующие упражнения могут быть выполнены только в том случае, если Вы выполняете лабораторные упражнения на системе, которая поддерживается программным обеспечением терминального сервера. Проверьте со своим инструктором.

1. Войдите в систему с использованием правильных имени пользователя и пароля.
2. Попрактикуйтесь в использовании клавиши BREAK и различных команд терминального сервера:
 - Используйте клавишу BREAK для установления новой сессии с Вашего терминала.
 - Используйте команду SHOW SESSIONS, чтобы увидеть две Ваши сессии.
 - Используйте команды терминального сервера для перемещения вперед-назад между сессиями.
 - Используйте команду RESUME для возобновления каждой сессии.
 - Используйте команду LOCK для блокирования Вашей сессии.
 - Введите пароль для разблокирования Вашего терминала.
3. Выйдите из системы в обеих сессиях.

3.5.2. Ввод команд

3.5.2.1. Повторный вызов и редактирование командных строк

Это упражнение использует некоторые команды, которые еще не рассмотрены в лекциях. Цель этого упражнения - дать Вам практический опыт ввода команд, повторного вызова и редактирования командных строк и использования квалификаторов.

1. Войдите в систему.
2. Введите команду *SHOW TME*.
Что происходит?
3. Введите команду *DCL SHOW USERS*.
4. Введите команду *DCL SHOW USERS/FULL/OUTPUT=USERS.LIST*
Появилось ли что-нибудь на Вашем экране?
5. Введите команду *DCL TYPE USERS.LIST*.
6. Используйте клавишу СТРЕЛКА ВВЕРХ для повторного вызова предыдущей команды и отредактируйте командную строку, добавив квалификатор */PAGE* к команде *TYPE*:

TYPE/PAGE USERS.LIST

Обратите внимание на разницу в результате.

7. Введите команду *DCL RECALL/ALL*.
Что выведено на экран?
8. Используйте команду *RECALL [номер]* для повторного вызова команды *SHOW TME*. Используйте редактирование командной строки для исправления ошибки набора:

SHOW TIME

Нажмите RETURN, чтобы ввести исправленную команду.

3.5.2.2. Команды *SET TERMINAL* и *SHOW TERMINAL*

1. Введите команду *SET TERMINAL/WIDTH=132*.
Что происходит?
2. Введите команду *SHOW TERMINAL* и найдите значение ширины.
3. Используйте команду *SET TERMINAL/WIDTH=80*.

3.6. Решения к письменным упражнениям

3.6.1. Терминология входа в систему

Поставьте в соответствие определению слева выражение справа. Выражения могут быть использованы более одного раза, и некоторые определения могут соответствовать более, чем одному выражению.

Определения	Выражения
<u>d k</u> Обязательная информация, которая должна быть введена пользователем во время процедуры входа в систему.	a. BREAK b. CONNECT
<u>j</u> Требуется, если Вы хотите иметь более одной	c. Подсказка DCL

Определения	Выражения
терминальной сессии в действии одновременно.	d. Пароль
<u>bfi</u> Команда, используемая, когда доступ к системе осуществляется через терминальный сервер	e. Сессия
<u>c</u> \$	f. SET FORWARD
<u>i</u> Команда, используемая после подсказки терминального сервера для вывода на экран числа сессий	g. SET HOST
<u>h</u> Команда, используемая для изменения пароля	h. SET PASSWORD
<u>g</u> Команда, используемая для доступа к другой системе в компьютерной сети	i. SHOW SESSIONS
<u>b</u> Команда для инициализации доступа к системе через терминальный сервер	j. Терминальный сервер
<u>a</u> Используется для возврата к подсказке терминального сервера без выхода из системы	k. Имя пользователя
<u>d</u> Слово или предложение, не отображаемое на терминале во время набора, требующееся для доступа к системе	

3.6.2. Компоненты команды DCL

Поставьте в соответствие компонентам командной строки слева определения справа.

Командная строка:

\$ TYPE MYFILE.001/PAGE

Компонент	Определение
<u>d</u> \$	a. Разделитель команды
<u>c</u> TYPE	b. Параметр команды
<u>a</u> (пробел)	c. Команда DCL
<u>b</u> MYFILE.001	d. Подсказка DCL
<u>f</u> /	e. Квалификатор
<u>e</u> PAGE	f. Разделитель квалификатора

3.6.3. Клавиши терминала

Подберите букву, обозначающую клавишу или комбинацию клавиш, которая лучше выполняет каждую из операций в левой колонке. Вы можете не использовать все буквы из списка.

Операции	Специальные функциональные клавиши
<u>b</u> Вы вошли в систему. Длинная череда сообщений, все из которых вы уже раньше видели, движется по Вашему экрану. Подавите эти сообщения, не останавливая и не сбрасывая программу, которая их производит.	a. CTRL/B b. CTRL/O c. CTRL/Q
<u>h</u> Вы только что набрали символы TYPE FILE&. Курсор находится сразу за символом &. Удалите этот символ.	d. CTRL/R e. CTRL/S

Операции	Специальные функциональные клавиши
<u>g</u> Вы ввели команду SHOW SYSTEM. Список процессов в Вашей системе ползет по Вашему экрану. Прекратите дальнейшее выполнение команды и верните управление своему терминалу. <u>e</u> Вы ввели следующую строку на Вашем терминале: \$ SHOW PROCESS/ALL Строки информации ползут по экрану Вашего терминала. Остановите вывод и остановите, но не сбрасывайте программу, которая его генерирует. <u>c</u> Возобновите генерацию выходной информации, которую вы остановили в предыдущей операции. <u>d</u> Вы сделали обширные исправления в командной строке на печатающем терминале. В результате получилось следующее: \$ PRYNT \ TNY\INT9\9\ FILIN\NI\ Выведите эту строку без отображения исправлений. <u>a</u> Вы только что ввели командную строку. Вызовите эту команду повторно.	f. CTRL/U g. CTRL/Y h. DEL или RUBOUT i. RETURN

3.6.4. Системные сообщения

Поставьте в соответствии компонентам системного сообщения об ошибке слева их определения справа.

Сообщение об ошибке:

```
$ MAKE
%DCL-W-IVVERB, unrecognized command verb - check validity and spelling
\MAKE\
```

Компоненты	Определения
<u>e</u> %	a. Код средства
<u>a</u> DCL	b. Код идентификации сообщения
<u>d</u> W	c. Текст сообщения
<u>b</u> IVVERB	d. Код уровня серьезности
<u>c</u> unrecognized...	e. Разделитель системного сообщения