

Домашнее задание № 1

Задание № 1

1. В каком файле хранится основная информация о пользователях? С помощью какой команды он может быть просмотрен?

Основная информация о пользователях на большинстве систем Linux хранится в файле `/etc/passwd`.

Для просмотра содержимого этого файла можно использовать команду:

```
cat /etc/passwd
```

Также можно воспользоваться командой `less` для удобного просмотра содержимого:

```
less /etc/passwd
```

2. Подробно описать информацию о текущем пользователе, которая хранится в файле, указанном в ответе на предыдущий вопрос.

3. Где и в каком виде хранятся пароли пользователей? С помощью какой команды они могут быть просмотрены?

Пароли пользователей в системах Linux хранятся в зашифрованном виде в файле `/etc/shadow`. Содержимое файла `/etc/shadow` можно просмотреть с помощью команды:

```
cat /etc/shadow
```

* Отображение паролей в чистом виде запрещено в современных системах из соображений безопасности

4. В каком файле хранится информация о группах? С помощью какой команды он может быть просмотрен?

Информация о группах пользователей хранится в файле `/etc/group`.

Для просмотра содержимого этого файла можно использовать команду:

```
cat /etc/group
```

Также можно воспользоваться командой `less` для удобного просмотра:

```
less /etc/group
```

5. Информация из каких файлов позволяет узнать полный перечень групп, в которых состоит пользователь? На примере текущего пользователя и реальных системных файлов провести данный анализ

Для получения полного перечня групп, в которых состоит пользователь, используются следующие файлы: 1. `/etc/group` — основной файл, содержащий информацию о группах и их членах 2. `/etc/gshadow` — содержит зашифрованные

пароли для групповых учетных записей и их дополнительные параметры 3. etc/passwd — этот файл хранит информацию о пользователях, но он также может содержать информацию о группах, к которым относится пользователь, через их идентификатор группы (GID).

6. Привести перечень команд (желательно, не менее 2 шт.), которые позволяют проверить информацию о текущем пользователе, которая была получена в результате поиска ответа на предыдущий вопрос.

Для проверки информации о текущем пользователе, включая группы, к которым он принадлежит, можно использовать следующие команды: 1. Команда `id` - Эта команда выводит идентификатор пользователя, группы и группы, в которых он состоит. 2. Команда `groups` - Эта команда отображает группы, к которым принадлежит текущий пользователь.

Задание № 2

1. Что такое переменная оболочки (локальная переменная)? Какими свойствами она обладает? Привести пример команды, создающей переменную оболочки, и пример команды, выводящей её значение

Переменная оболочки (локальная переменная) — это переменная, которая доступна только в пределах текущей сессии или конкретной оболочки (сессии терминала). Такие переменные сохраняются только временно и исчезают после завершения сессии.

Свойства переменной оболочки: - Только временное использование — значение переменной сохраняется только в текущей сессии оболочки. - Изменяемость — переменные можно изменять и удалять вручную или через команды. - Область действия — доступ ограничен текущей сессией или определенным процессом.

Пример команды, создающей переменную оболочки:

```
MYVAR="Hello, world!"
```

Пример команды, выводящей значение переменной:

```
echo $MYVAR
```

2. Что такое переменная окружения (глобальная переменная)?

Переменная окружения (глобальная переменная) — это переменная, которая доступна для всех процессов, запущенных в системе, включая оболочку и дочерние процессы. Такие переменные обычно используются для хранения информации, которая должна быть доступна системным утилитам и другим программам.

Свойства переменной окружения: - Глобальное использование — доступна всем процессам и приложениям в системе. - Изменяемость — значения можно

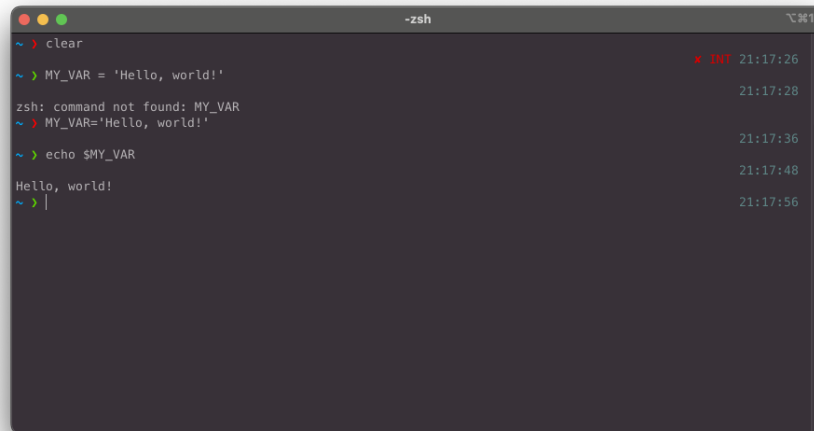
A screenshot of a terminal window titled "-zsh" with a temperature indicator of 11°C in the top right corner. The terminal shows a sequence of commands and their outputs:
1. Command: `clear`
2. Command: `MY_VAR = 'Hello, world!'`
3. Output: `zsh: command not found: MY_VAR`
4. Command: `MY_VAR='Hello, world!'`
5. Command: `echo $MY_VAR`
6. Output: `Hello, world!`
The terminal also displays a vertical timestamp column on the right side with times: 21:17:26, 21:17:28, 21:17:36, 21:17:48, and 21:17:56.

Figure 1: Установка и вывод на экран переменной оболочки

изменять с помощью команд. - Взаимодействие с другими программами — позволяет программам обмениваться данными между собой.

Пример команды, создающей переменную окружения:

```
export MYENV="                "
```

Пример команды, выводящей значение переменной окружения:

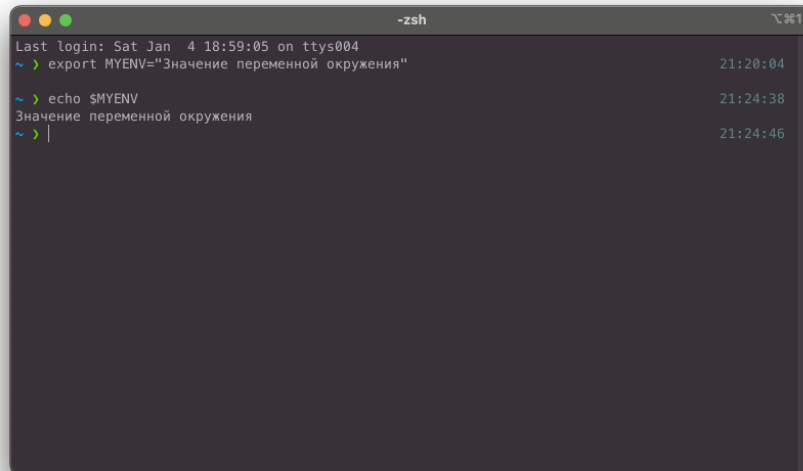
```
echo $MYENV
```

3. На примере двух переменных, созданных в процессе подготовки ответов на предыдущие вопросы, показать на практике основное различие между локальной и глобальной переменными

Различие между глобальной и локальной переменными можно продемонстрировать, создав дочерний `bash`-процесс и попробовав вывести в нем значения данных переменных. Глобальная переменная будет так же доступна в дочернем процессе, в то время как локальная переменная - нет, так как она доступна только в рамках текущей сессии терминала.

4. Каким образом возможно задать значение переменной, которое будет устанавливаться автоматически при входе в систему для текущего пользователя?

Чтобы задать значение переменной, которое будет автоматически устанавливаться при входе в систему для текущего пользователя, необходимо добавить её в файл конфигурации оболочки пользователя.

A screenshot of a terminal window titled '~zsh'. The window shows the following commands and their outputs: 'Last login: Sat Jan 4 18:59:05 on ttys004', '~ > export MYENV="Значение переменной окружения"' with a timestamp of 21:20:04, '~ > echo \$MYENV' with a timestamp of 21:24:38, and the output 'Значение переменной окружения' with a timestamp of 21:24:46. The prompt '~ >' is visible at the end of the last line.

```
Last login: Sat Jan 4 18:59:05 on ttys004
~ > export MYENV="Значение переменной окружения"
~ > echo $MYENV
Значение переменной окружения
~ > |
```

Figure 2: Установка и вывод на экран переменной окружения

Файлы конфигурации оболочки:

~/ .bashrc (или ~/ .zshrc) - выполняется для интерактивных оболочек при входе в терминал (чаще всего используется).

~/ .bash_profile или (~/ .profile, ~/ .zsh_profile): Выполняется только при входе в систему (например, через SSH или консоль).

Чтобы добавить переменную в конфигурационный файл Bash одной строкой, можно использовать команду echo с перенаправлением в файл. Например:

```
echo 'export MYENV="                "' >> ~/.zshrc
```

Применение изменений:

Чтобы изменения сразу вступили в силу, необходимо выполнить команду:

```
source ~/.zshrc
```

Теперь переменная MYENV будет автоматически устанавливаться при входе в систему.

5. Каким образом возможно задать значение переменной, которое будет устанавливаться автоматически при входе в систему для всех пользователей?

Чтобы задать значение переменной, которая будет устанавливаться автоматически при входе в систему для всех пользователей, её нужно прописать в системных

```
bash
Last login: Sat Jan  4 21:20:04 on ttys001
~ > echo $MY_VAR                                00:54:32

~ > echo $MY_VAR                                00:58:57

~ > MY_VAR='Hello, world!'                       00:59:00
~ > echo $MY_VAR                                00:59:03
Hello, world!
~ > bash                                         00:59:05

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
bash-3.2$ echo $MY_VAR

bash-3.2$ exit
exit
~ >                                              13s 00:59:20
~ > echo $MY_VAR                                00:59:20
Hello, world!
~ > export MYENV="Значение переменной окружения" 00:59:22
~ > echo $MYENV                                00:59:53
Значение переменной окружения
~ > bash                                         01:00:02

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
bash-3.2$ echo $MYENV
Значение переменной окружения
bash-3.2$ |
```

Figure 3: Вывод глобальной и локальной переменных в докерном процессе

конфигурационных файлах, которые используются для настройки окружения.

Системные файлы для задания переменных:

1. **/etc/profile** - Этот файл выполняется при входе в систему всеми пользователями.
2. **Файлы в директории /etc/profile.d/** - Можно создать отдельный скрипт в этой директории для установки переменной.
3. **/etc/environment** - Используется для задания переменных окружения, но в отличие от других файлов, он не поддерживает сложные команды (только формат **VARIABLE=value**).

6. Что такое переменная среды окружения PATH? Вывести значение переменной среды окружения PATH

Переменная среды окружения **PATH** содержит в себе строку со списком директорий (разделенных символом ":"), в которых по-умолчанию будет осуществляться поиск исполняемого файла команды.

7. Указать команду, которая позволяет добавить к значению переменной среды окружения PATH абсолютный путь домашнего каталога текущего пользователя

```
export PATH=$PATH:$HOME
```

Задание № 3

1. Чем отличаются команды, встроенные в командный интерпретатор, и внешние? С

Встроенные команды выполняются внутри самого командного интерпретатора (оболочки, например Bash). Они не требуют запуска отдельного процесса, что делает их выполнение быстрее.

Внешние команды - это отдельные исполняемые файлы, расположенные в директориях, указанных в переменной окружения PATH (например, /bin, /usr/bin). Выполнение внешней команды запускает новый процесс.

С помощью каких команд возможно определить, является команда внешней или внутренней?

1. **Команда type** - сообщает, встроена ли команда в оболочку или является внешней.
2. **Команда which**: Показывает путь к исполняемому файлу внешней команды. Если команда встроенная, which не даст результата
3. **Команда command -v**

2. Чем отличаются команды `man` и `help`, предоставляющие справочную информацию? Привести примеры их использования с учётом особенностей получения справочной информации для встроенных и внешних команд

Команда `help` предоставляет справку только для встроенных команд оболочки. Формат справки обычно менее подробный. **Команда `man`**, напротив - предоставляет подробные справочные страницы (man-pages) для внешних команд, конфигурационных файлов, системных вызовов, библиотек и т.д. Формат справки более полный и структурированный.

3. С помощью какой команды можно делать псевдонимы для существующих команд? Привести примеры использования данной команды

Для создания псевдонима используется команда `alias`.

```
alias ll='ls -aF'
```

4. С помощью какого вызова команды, которая используется в предыдущем вопросе, можно определить псевдоним `d` для некой существующей команды, позволяющий выводить текущую дату и время в формате Unix time (количество секунд, прошедших с 1 января 1970 года 00:00:00 UTC)?

```
alias d='date +%s'
```

5. Как сделать так, чтобы псевдоним, определение которого было необходимо для ответа на предыдущий вопрос, автоматически определялся при каждом входе текущего пользователя в систему?

Достаточно добавить строчку установки псевдонима в файл конфигурации оболочки командной строки пользователя (`.bashrc`, `.zshrc`)

```
alias d='date +%s'
```

Для того, чтобы изменения вступили в силу, без перезагрузки текущей сессии пользователя необходимо выполнить команду

```
source ~/.bashrc
```