

Работа с Unix

Работу выполнил:
Рыбалко Тимофей Александрович
НБИбд-02-24

Российский университет дружбы народов, Москва,
Россия

Введение

- Освоить работу с файловой системой Linux необходимо.
- Умение работать с git и GitHub важно для IT-специалистов.

Цель работы

- Изучить основы программирования в оболочке bash.
- Научиться создавать и выполнять командные файлы.

Техническое обеспечение

- Виртуальная машина: VirtualBox или QEMU.
- Операционная система: Linux (дистрибутив Fedora).
- Требования: Intel Core i3-550, 4 ГБ ОЗУ, 80 ГБ свободного места.

Создание скрипта резервного копирования

- Создаем скрипт для автоматического бэкапа:

```
GNU nano 8.3                               /home/tarihbalko/backup_script.sh      Изменён
# Проверяем существование директории backup
if [ ! -d ~/backup ]; then
    mkdir -p ~/backup
fi

# Архивируем скрипт с помощью tar
backup_file="backup_script_$(date +%Y%m%d_%H%M%S).tar.gz"
tar -czf ~/backup/$backup_file $0

echo "Резервная копия создана: ~/backup/$backup_file"
```

Обработка аргументов командной строки

- Пишем обработчик аргументов командной строки

```
GNU nano 8.3 /home/tarihbalko/args_processor.sh
# Проверяем существование директории backup
if [ ! -d ~/backup ]; then
    mkdir -p ~/backup
fi

# Архивируем скрипт с помощью tar
backup_file="backup_script_$(date +%Y%m%d_%H%M%S).tar.gz"
tar -czf ~/backup/$backup_file $0

echo "Резервная копия создана: ~/backup/$backup_file"
```

Реализация аналога команды ls

- Разрабатываем упрощенный аналог команды ls

```
GNU nano 8.3 /home/tarihbalko/my_ls.sh Изменён
#!/bin/bash

dir=${1:-.} # Используем текущий каталог, если аргумент не указан

echo "Содержимое каталога $dir:"
echo "-----"

for file in "$dir"/*; do
    if [ -d "$file" ]; then
        type="Каталог"
    elif [ -f "$file" ]; then
        type="Файл"
    else
        type="Другое"
    fi

    permissions=""
    if [ -r "$file" ]; then permissions="${permissions}r"; else permissions="${permissions}"
    if [ -w "$file" ]; then permissions="${permissions}w"; else permissions="${permissions}"
    if [ -x "$file" ]; then permissions="${permissions}x"; else permissions="${permissions}"

    echo "type: ${file##*/} | Права: $permissions"
done
```

Подсчет файлов по расширению

- Создаем счетчик файлов по расширению

```
GNU nano 8.3 /home/tarihbalko/count_files.sh
#!/bin/bash

if [ $# -ne 2 ]; then
    echo "Использование: $0 <расширение> <директория>"
    exit 1
fi

extension=$1
directory=$2

if [ ! -d "$directory" ]; then
    echo "Ошибка: директория $directory не существует"
    exit 1
fi

count=$(find "$directory" -type f -name ".*$extension" | wc -l)

echo "Количество файлов с расширением .$extension в $directory: $count"
```


Отображение созданных скриптов

- Убеждаемся в корректном создании всех скриптов

```
[tarihbalko@vbox ~]$ ls -l ~/.sh
-rw-r--r--. 1 tarihbalko tarihbalko 355 мая  2 17:30 /home/tarihbalko/args_procesor.sh
-rw-r--r--. 1 tarihbalko tarihbalko 355 мая  2 17:31 /home/tarihbalko/args_processor.sh
-rw-r--r--. 1 tarihbalko tarihbalko 460 мая  2 17:33 /home/tarihbalko/count_files.sh
-rw-r--r--. 1 tarihbalko tarihbalko  98 апр 26 17:20 /home/tarihbalko/lab07.sh
-rw-r--r--. 1 tarihbalko tarihbalko 783 мая  2 17:32 /home/tarihbalko/my_ls.sh
```

Ответы на вопросы:

- 1. Командная оболочка - интерпретатор команд ОС (bash, zsh, ksh), отличаются функционалом и синтаксисом.
- 2. POSIX - стандарт совместимости UNIX-систем, обеспечивающий переносимость.
- 3. Переменные - var=значение, массивы - arr=(знач1 знач2).
- 4. let - арифметические вычисления, read - чтение ввода пользователя.
- 5. Арифметические операции - + - * / % ** и битовые операции.
- 6. (()) - конструкция для арифметических вычислений и сравнений.
- 7. Стандартные переменные - PATH, HOME, USER, PS1, PWD.
- 8. Метасимволы - спецсимволы (* ? > < |), имеющие особое значение.
- 9. Экранирование - через \, ' ' или " " (частичное).
- 10. Создание - текстовый файл с shebang (#!/bin/bash), запуск - chmod +x и ./файл.
- 11. Функции - func_name() { команды } или function func_name { команды }.
- 12. Проверка - test -f (файл), test -d (каталог) или [-f файл].
- 13. set - управление параметрами оболочки, typeset - объявление переменных, unset - удаление.
- 14. Передача параметров - через аргументы командной строки (\$1, \$2, ...\$n).
- 15. Специальные переменные: \$# - количество, \$* - все, \$? - код, \$\$ - PID, \$0 - имя

Заключение

- Освоены базовые команды bash.
- Созданы полезные скрипты для работы.