Конспект к теме

ОСНОВЫ НАПИСАНИЯ СКРИПТОВ

Введение

> **Скрипты на Bash** — это мощный инструмент для автоматизации задач в Linux. Они позволяют выполнять команды, упрощать рутинные процессы и создавать собственные программы

В этой теме вы узнаете

- Что такое скрипты и как их писать
- Как работать с переменными и аргументами
- Как добавлять логику с помощью условных операторов и циклов
- Как работать с файлами и вести логи

Введение в скрипты Bash

> **Скрипт Bash** — это текстовый файл с командами, которые выполняются последовательно
>
> Используется для автоматизации рутинных задач, создания утилит и обработки данных

Создание скрипта

`nano script.sh` Создайте файл с расширением .sh

`#!/bin/bash` Начните файл с shebang (указания интерпретатора)

Запуск скрипта

`chmod +x script.sh` Сделайте файл исполняемым

`./script.sh` Запустите скрипт

Пример простого скрипта

```
#!/bin/bash
echo "Привет, мир!"
```

Скрипты Bash — это простой способ автоматизировать задачи и облегчить работу с системой

Переменные и аргументы

Переменные

`имя_переменной="значение"` Создание переменной

`echo $имя_переменной` Использование переменной

```
приветствие="Привет"
echo "$приветствие, мир!"
```

Пример

Аргументы

`./script.sh аргумент1 аргумент2` Передача аргументов при запуске скрипта

Чтение аргументов

`$1, $2, …` Аргументы

`$@` Все аргументы

`$#` Количество аргументов

`echo "Вы ввели: $1 и $2"` Пример

Чтение пользовательского ввода

Используйте команду read:

```
echo "Введите ваше имя:"
read имя
echo "Привет, $имя!"
```

Переменные и аргументы позволяют создавать универсальные и интерактивные скрипты

Условные операторы и циклы

Условные операторы

Синтаксис if

```
if [ условие ]; then
  команды
elif [ другое_условие ]; then
  команды
else
  команды
fi
```

Пример

```
if [ $1 -eq 0 ]; then
  echo "Ноль"
else
  echo "Не ноль"
fi
```

Циклы

Цикл for

```
for i in {1..5}; do
  echo "Счётчик: $i"
done
```

Цикл while

```
while [ $число -gt 0 ]; do
  echo $число
  число=$((число - 1))
done
```

Пример комбинирования

Перебор всех файлов в директории

```
for файл in *; do
  echo "Файл: $файл"
done
```

Условные операторы и циклы добавляют логику и повторяемость в скрипты, делая их более функциональными

Работа с файлами и логами

Работа с файлами

```
touch file.txt
```

Создание файлов

```
cp source.txt destination.txt
```

Копирование файлов

```
rm file.txt
```

Удаление файлов

Запись данных в файлы

```
echo "Текст" > file.txt
```

Перезапись файла

```
echo "Другой текст" >> file.txt
```

Добавление текста в файл

Ведение логов

```
echo "$(date): Сообщение" >> script.log
```

Логирование данных с указанием времени

Пример работы

Создайте скрипт, который проверяет наличие файла и записывает результат в лог:

```
#!/bin/bash
if [ -e "$1" ]; then
  echo "$(date): Файл $1 существует" >> script.log
else
  echo "$(date): Файл $1 не найден" >> script.log
fi
```

Работа с файлами и логами в скриптах позволяет сохранять данные и отслеживать выполнение задач

ИТОГ

На этом уроке вы узнали, как создавать и запускать скрипты Bash, работать с переменными, добавлять логику через условные операторы и циклы, а также управлять файлами и логами. Эти навыки позволят вам автоматизировать задачи и работать с системой более эффективно