

Лабораторная работа № 13

Программирование в командном процессоре ОС Unix. Ветвления и циклы

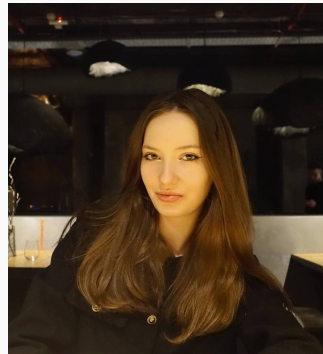
Скобеева А.А.

07 марта 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Скобеева Алиса Алексеевна
- студентка 1-го курса направления “Прикладная информатика”
- Российский университет дружбы народов
- 1132246836@pfur.ru



Вводная часть

- Данная презентация актуальна для студентов технических направлений, изучающих предмет “Архитектура компьютеров: операционные системы”, а также для всех, кто интересуется программированием в командном процессоре ОС UNIX.

- Командный процессор ОС UNIX
- Fedora Sway

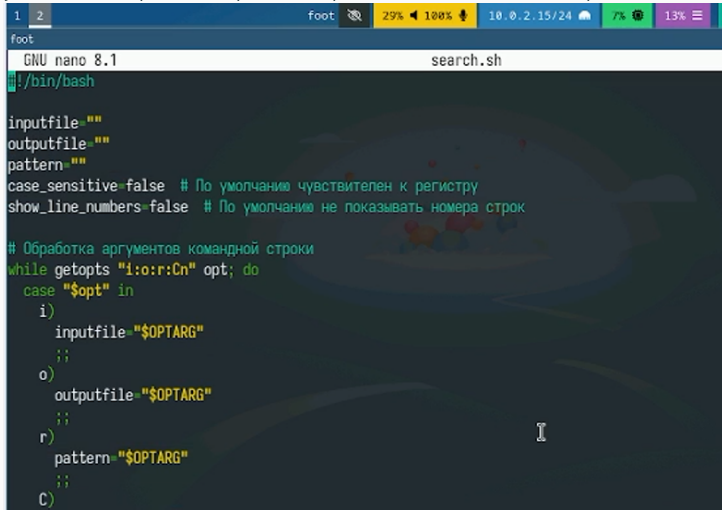
- Изучить ОП в оболочке ОС UNIX
- Научиться писать более сложные команды с использованием логических управляющих конструкций и циклов

- Оболочка ОС UNIX
- Материалы ТУИС

Основная часть

1-ый командный файл

- Пишем файл, который анализирует командную строку с ключами, а затем ищет в указанном файле строки, определяемые ключом -p.



```
1 2
foot 29% 100% 10.0.2.15/24 7% 13%
GNU nano 8.1 search.sh
#!/bin/bash

inputfile=""
outputfile=""
pattern=""
case_sensitive=false # По умолчанию чувствителен к регистру
show_line_numbers=false # По умолчанию не показывать номера строк

# Обработка аргументов командной строки
while getopts "i:o:r:Cn" opt; do
  case "$opt" in
    i)
      inputfile="$OPTARG"
      ;;
    o)
      outputfile="$OPTARG"
      ;;
    r)
      pattern="$OPTARG"
      ;;
    C)

```

```
foot
GNU nano 8.1                                output.txt
1:example
2:example
5:example example example
6:example
7:flower example cake
8:space space space example
```

2-ой командный файл

- Пишем следующую программу, которая определяет, что число больше, меньше или равно нулю

```
GNU nano 8.1                                check_number.sh
#!/bin/bash

read -p "Введите число: " num

# Проверка, является ли ввод числом. Ограничение: допускаются только целые числа
if ! [[ "$num" =~ ^?[0-9]+$ ]]; then
    echo "Ошибка: Неверный ввод. Введите целое число."
    exit 2
fi

# Сравнение числа и вывод сообщения
if (( num > 0 )); then
    echo "Число больше нуля."
    exit 0
elif (( num < 0 )); then
    echo "Число меньше нуля."
    exit 1
else
    echo "Число равно нулю."
    exit 3
fi
```

```
[aaskobeeva@fedora ~]$ ./check_number.sh  
Введите число: 4  
Число больше нуля.  
[aaskobeeva@fedora ~]$ ./check_number.sh  
Введите число: 0  
Число равно нулю.
```

3-ий командный файл

- Пишем командный файл, создающий указанное число файлов, а также умеющий их удалять

```
GNU nano 8.1                                     filemake1.sh
exit 1
fi

# Создание файлов
N="$1"
for i in $(seq 1 $N); do
    touch "$i.tmp"
done

echo "Создано $N файлов."

# Функция для удаления файлов
cleanup() {
    echo "Удаление временных файлов..."
    for i in $(seq 1 $N); do
        rm -f "$i.tmp"
    done
    echo "Временные файлы удалены."
}
```

```
[aaskobeeva@fedora ~]$ ./filemake1.sh 5  
Создано 5 файлов.  
Удаление временных файлов...  
Временные файлы удалены.  
[aaskobeeva@fedora ~]$ ./filemake1.sh 14  
Создано 14 файлов.  
Удаление временных файлов...  
Временные файлы удалены.
```

4-ый командный файл

- Пишем командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории

```
foot
GNU nano 8.1                                lasttask.sh
#!/bin/bash

# Скрипт для архивирования файлов, измененных менее недели назад

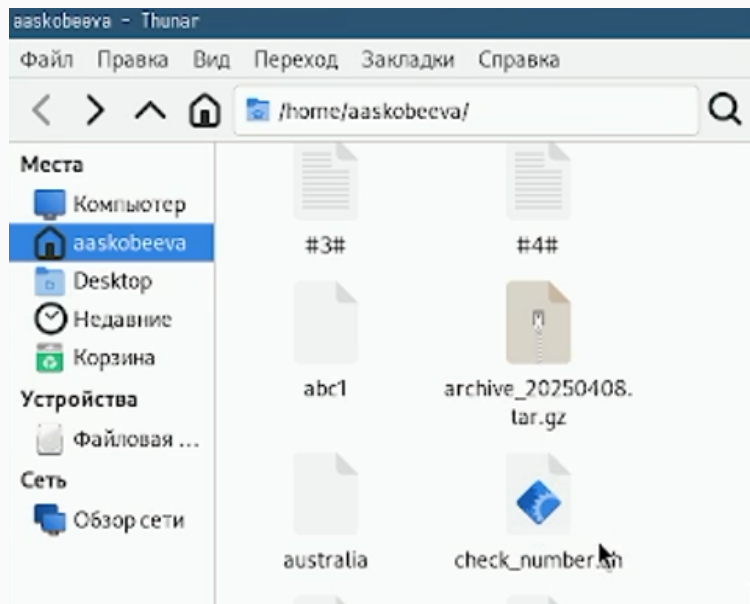
# Проверка наличия аргумента (каталога)
if [ -z "$1" ]; then
    echo "Использование: $0 <каталог>"
    exit 1
fi

directory="$1"

# Проверка, что каталог существует
if [ ! -d "$directory" ]; then
    echo "Ошибка: Каталог '$directory' не существует."
    exit 1
fi

# Определение имени архива
archive_name="archive_$(date +%Y%m%d).tar.gz"
```


Результат работы 4-го файла



- Мы успешно написали 4 программы и проверили корректность их работы