

# Лабораторная работа №13

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Титков Ярослав Максимович

10 мая 2025 г.

Российский университет дружбы народов, Москва, Россия

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `-i` — прочитать данные из указанного файла; `-o` — вывести данные в указанный файл; `-r` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-r`.
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в код завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до `N` (например `1.tmp`, `2.tmp`, `3.tmp`, `4.tmp` и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).

# 1. Работы с командами и файлами

```
/bin/bash

while getopts "i:op:cn" opt; do
  case $opt in
    i) inputfile="$OPTARG" ;;
    o) outfile="$OPTARG" ;;
    p) pattern="$OPTARG" ;;
    C) case_sensitive=1 ;;
    n) with_line_numbers=1 ;;
    *) echo "Usage: $0 -i inputfile -o outfile -p pattern [-C] [-n]" ; exit 1 ;;
  esac
done

if [ -z "$inputfile" ] || [ -z "$outfile" ] || [ -z "$pattern" ]; then
  echo "Missing required options"
  exit 1
fi

grep_opts=""
[ "$with_line_numbers" = 1 ] && grep_opts="$grep_opts -n"
[ "$case_sensitive" = 1 ] && grep_opts="$grep_opts -i"

grep $grep_opts "$pattern" "$inputfile" -> "$outfile"
```

Рис. 1: Работа с командами

## 2. Написания программ на языке Си

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    int number;
    printf("Введите число: ");
    scanf("%d", &number);

    if (number > 0)
        exit(1);
    else if (number < 0)
        exit(2);
    else
        exit(0);
}
```

Рис. 2: Написание программ на Си

### 3. Написания программ командных файлов

```
/bin/bash
./check_number
code=57

case $code in
  0) echo "Число равно нулю" ;;
  1) echo "Число больше нуля" ;;
  2) echo "Число меньше нуля" ;;
  *) echo "Неизвестный код: $code" ;;
esac
```

Рис. 3: Командные файлы

## 4. Написания программ командных файлов

```
/bin/bash

if [ -z "$1" ]; then
    echo "Использование: $0 N [--delete]"
    exit 1
fi

N=$1

if [ "$2" == "--delete" ]; then
    for ((i=1; i<=N; i++)); do
        file="$i.tmp"
        if [ -f "$file" ]; then
            rm "$file"
            echo "Удалён: $file"
        else
            echo "Файл не существует: $file"
        fi
    done
else
    for ((i=1; i<=N; i++)); do
        touch "$i.tmp"
        echo "Создан: $i.tmp"
    done
fi
```

Прочитано 25 строк

Справка	Записать	Поиск	Вырезать	Выполнить	Полная	Отмена	Установить м...	На скобку	Преддущий	Назад
Выход	ЧитФайл	Замена	Вставить	Выровнять	К строке	Повтор	Копировать	Обр. поиск	Следующий	Вперед

Рис. 4: Командные файлы

## 5. Написания программ командных файлов

```
/bin/bash
if [ -z "$1" ]; then
    echo "Использование: $0 <директория>"
    exit 1
fi

dir="$1"
timestamp=$(date +%Y%m%d_%H%M%S)
archive_name="archive_${timestamp}.tar.gz"

find "$dir" -type f -mtime -7 -print0 | tar -czvf "$archive_name" --null -T -
echo "Архив создан: $archive_name"
```

Рис. 5: Написания программ



## 6. Компиляция всего

```
/bin/bash
if [ -z "$1" ]; then
    echo "Использование: $0 <директория>"
    exit 1
fi

dir="$1"
timestamp=$(date +%Y%m%d_%H%M%S)
archive_name="archive_${timestamp}.tar.gz"

find "$dir" -type f -mtime -7 -print0 | tar -czvf "$archive_name" --null -T -
echo "Архив создан: $archive_name"
```

1. Прочитать 14 строк

Рис. 6: Компиляция файлов

Изучил основы программирования в оболочке ОС UNIX. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов