# Лабораторная работа №11

Архитектура ОС

Демидович. Н. М.

22 апреля 2023

Российский университет дружбы народов, Москва, Россия

Докладчик

#### Докладчик

- Демидович Никита Михайлович
- Студент группы НКАбд-01-22
- Кафедра прикладной информатики и теории вероятностей
- Российский университет дружбы народов
- · 1132221550@pfur.ru
- https://github.com/nikdem1



Цели и задачи

# Цель лабораторной работы

Цель данной лабораторной работы - изучить основы программирования в оболочке ОС UNIX, научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Задачи

#### Задачи

#### Последовательность выполнения работы:

- 1. Используя команды getopts grep, написать командный файл, который анализирует командную строку с ключами:
- · -iinputfile прочитать данные из указанного файла;
- · -ooutputfile вывести данные в указанный файл;
- -ршаблон указать шаблон для поиска;
- -С различать большие и малые буквы;
- · -n выдавать номера строк.

а затем ищет в указанном файле нужные строки, определяемые ключом -р. 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 о коде завершения в оболочку. Команд- ный файл должен

PRIZEDZIŁ ZIV BROFRZMANI W BROZUZBUZBUZBORZE C BOMOULIO KOMZUBLI \$2 PRIBZIŁ COOKULOUMO O

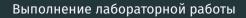
# Теоретическое введение

## Теоретическое введение

Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек:

- оболочка Борна (Bourne shell или sh) стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций;
- С-оболочка (или csh) надстройка на оболочкой Борна, использующая С-подобный синтаксис команд с возможностью сохранения истории выполнения команд;
- оболочка Корна (или ksh) напоминает оболочку С, но операторы управления программой совместимы с операторами оболочки Борна;
- BASH сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек С и Корна (разработка компании Free Software

Выполнение лабораторной работы

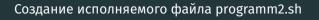


Далее описан ход выполнения данной лабораторной работы.

## Создание директории lab11, файлов input.txt, output.txt и исполняемого файла programm1.sh

На первом этапе выполнения работы я создал отдельную директорию для дальнейшей работы, командный файл programm1.sh, текстовый файлы input.txt и output.txt и приступил к написанию файла с командами getopts и grep, который анализирует командную строку с ключами: - -iinputfile — прочитать данные из указанного файла; - -ooutputfile — вывести данные в указанный файл; - -ршаблон — указать шаблон для поиска; - -С — различать большие и малые буквы; - -п — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом -р.

```
#! /bin/bash
while getopts i:o:p:cn optletter
do
case $optletter in
    i) iflag=1: ival=$OPTARG::
    o) oflag=1: oval=$OPTARG::
    p) pflag=1; pval=$OPTARG;;
    c) cflag=1;;
    n) nflag=1;;
    *) echo Illegal option $optletter::
    esac
done
```



После этого я создал исполняемый файл для второй программы.

```
#! /bin/bash
gcc -o cprog 12.c
./cprog
case $? in
0) echo "Число равно нулю";;
1) echo "Число больше нуля";;
2) echo "Число меньше нуля";;
esac
```

# Создание исполняемого файла 12.с

Также мною был создан файл 12.с для программы на языке Си.

#### Мсполняемый файл 12.с

```
#include <stdlib.h>
#include <stdio.h>
int main () {
    int n;
    printf ("Введите число: ");
    scanf ("%d", &n);
    if(n>0){
        exit(1);
    else if (n==0) {
        exit(0);
    else {
```

### Создание исполняемого файла programm3.sh

Далее я написал командный файл programm3.sh, 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, которое передается в аргументы командной строки.

```
#! /bin/bash
for((i=1; i<=$*; i++))
do
if test -f "$i".tmp
then rm "$i".tmp
else touch "$i.tmp"
fi
done</pre>
```

### Создание исполняемого файла programm4.sh

И на финальном этапе выполнения работы я создал исполняемый файл для четвертой программы. Это командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории.

# Мсполняемый файл programm4.sh

```
#! /bin/bash
find $* -mtime -7 -mtime +0 -type f > FILES.txt
tar -cf archive.tar -T FILES.txt
```

# Результаты

#### Результаты

В результате выполнения данной лабораторной работы я изучил основы программирования в оболочке ОС UNIX и научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Список источников

#### Список источников

Лабораторная работа №11 (Архитектура ОС).