

Отчёт по лабораторной работе №13

операционные системы

Ведьмина А.С.

Российский университет дружбы народов, Москва, Россия

Информация

- Ведьмина Александра Сергеевна
- студентка
- ФФМиЕН
- Российский университет дружбы народов
- 1132236003@rudn.ru
- <https://asvedjmina.github.io/ru/>



Цель работы

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

Задание

1. Ознакомиться с теоретическим введением.
2. Последовательно выполнить задания.

Выполнение лабораторной работы

1. 1.1 Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `-i`inputfile — прочитать данные из указанного файла; `-o`outputfile — вывести данные в указанный файл; `-r`шаблон — 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`.

Выполнение лабораторной работы

```
filename.sh - GNU Emacs at fedora
File Edit Options Buffers Tools Sh-Script Help
!#/bin/bash

# Инициализация переменных
inputfile=""
outputfile=""
pattern=""
case_sensitive=0
show_line_numbers=0

# Анализ командной строки с помощью getopt
while getopt "i:o:p:Cn" opt; do
    case $opt in
        i) inputfile="$OPTARG";;
        o) outputfile="$OPTARG";;
        p) pattern="$OPTARG";;
        C) case_sensitive=1;;
        n) show_line_numbers=1;;
        *) echo "Использование: $0 [-i inputfile] [-o outputfile] [-p pattern] [-C] [-n]" >&2
           exit 1;;
    esac
done

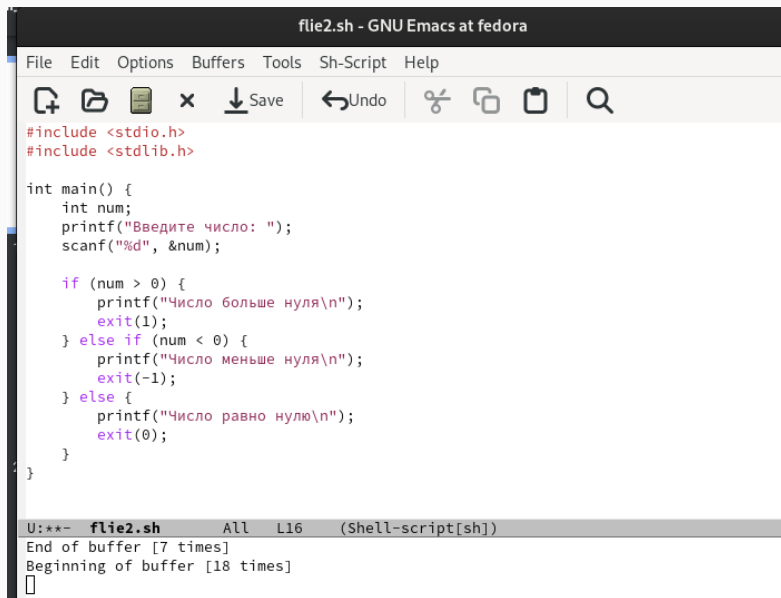
# Проверка наличия обязательных параметров
if [ -z "$inputfile" ] || [ -z "$pattern" ]; then
    echo "Ошибка: Не указаны обязательные параметры" >&2
    exit 1
fi

# Поиск строк, соответствующих шаблону
if [ $case_sensitive -eq 1 ]; then
    grep -E "$pattern" "$inputfile"
else
    grep -Ei "$pattern" "$inputfile"
fi | {
    if [ $show_line_numbers -eq 1 ]; then
        nl
    else
        cat
    fi
}
```

U:*** filename.sh Top L33 (Shell-script[sh])

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. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.

Выполнение лабораторной работы



```
flie2.sh - GNU Emacs at fedora
File Edit Options Buffers Tools Sh-Script Help
[Icons: New, Open, Save, Save As, Undo, Cut, Copy, Paste, Find]

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

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

    if (num > 0) {
        printf("Число больше нуля\n");
        exit(1);
    } else if (num < 0) {
        printf("Число меньше нуля\n");
        exit(-1);
    } else {
        printf("Число равно нулю\n");
        exit(0);
    }
}

U:*\*- flie2.sh    All    L16    (Shell-script[sh])
End of buffer [7 times]
Beginning of buffer [18 times]
█
```

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).

Выполнение лабораторной работы

```
file3.sh - GNU Emacs at fedora
File Edit Options Buffers Tools Sh-Script Help
[Icons: Save, Undo, Cut, Copy, Paste, Find]
#!/bin/bash

# Проверка наличия аргумента командной строки
if [ -z "$1" ]; then
    echo "Использование: $0 <количество_файлов>" >&2
    exit 1
fi

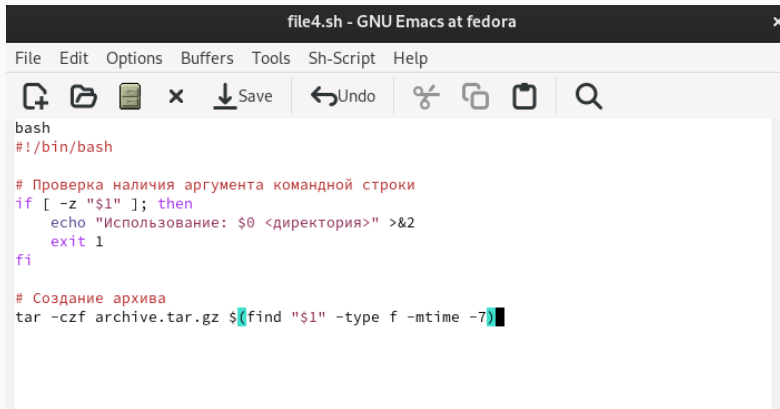
# Создание файлов
for i in $(seq 1 $1); do
    touch "$i.tmp"
done

# Удаление файлов
for i in *.tmp; do
    rm "$i"
done

U:*- file3.sh All L9 (Shell-script[sh])
Indentation variables are now local.
Indentation setup for shell type sh
Mark set
□
```

Выполнение лабораторной работы

4. Написать командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find).



```
bash
#!/bin/bash

# Проверка наличия аргумента командной строки
if [ -z "$1" ]; then
    echo "Использование: $0 <директория>" >&2
    exit 1
fi

# Создание архива
tar -czf archive.tar.gz $(find "$1" -type f -mtime -7)
```

Выводы

В ходе лабораторной работы я изучила основы программирования в оболочке ОС UNIX и научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.