

Отчет по лабораторной работе № 13

Операционные системы

Абронина Алиса Кирилловна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	12

Список иллюстраций

3.1	файл 1	7
3.2	файл 1.2	8
3.3	файл 1.3	8
3.4	файл 1.4	8
3.5	файл 2	9
3.6	файл 2.1	9
3.7	файл 2.2	9
3.8	файл 3	10
3.9	файл 3.1	10
3.10	файл 4	11
3.11	файл 4.1	11

Список таблиц

1 Цель работы

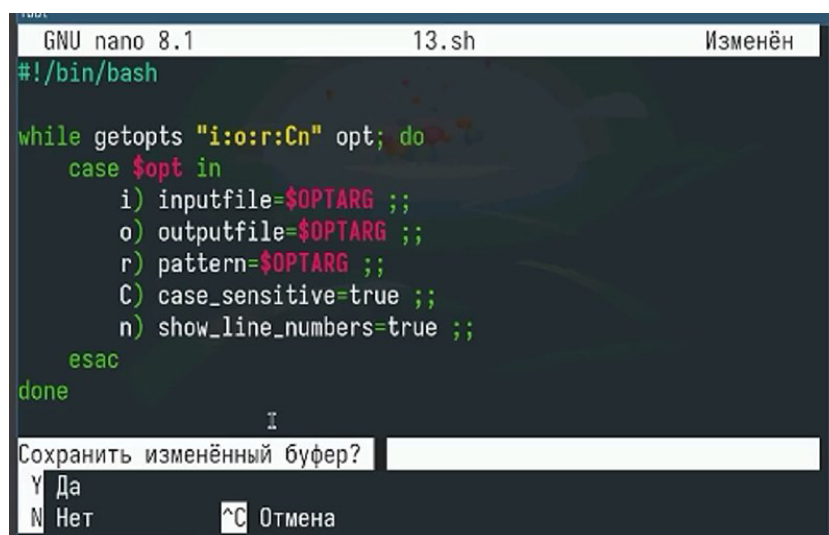
Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкции и циклов.

2 Задание

1. 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; – -п — выдавать номера строк.
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.
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.)
4. Написать командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории.

3 Выполнение лабораторной работы

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`. (рис. 3.1).(рис. 3.2).(рис. 3.3).(рис. 3.4).



```
GNU nano 8.1      13.sh      Изменён
#!/bin/bash

while getopts "i:o:r:Cn" opt; do
  case $opt in
    i) inputfile=$OPTARG ;;
    o) outputfile=$OPTARG ;;
    r) pattern=$OPTARG ;;
    C) case_sensitive=true ;;
    n) show_line_numbers=true ;;
  esac
done

I
Сохранить изменённый буфер?
Y Да
N Нет      ^C Отмена
```

Рис. 3.1: файл 1

```
/home/akabronina@lab13/result.txt
3:Есть ошибка
```

Рис. 3.2: файл 1.2

```
/home/akabronina@lab13/input.txt
Вот эта строка содержит ошибку
А вот тут нет ошибки
Есть ошибка
А вот это строка
```

Рис. 3.3: файл 1.3

```
[akabronina@vbox lab13]$ ./13.sh -i input.txt -o result.txt -r
"ошибка" -n
[akabronina@vbox lab13]$ ls
13.sh  input.txt  presentation  report  result.txt
```

Рис. 3.4: файл 1.4

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. (рис. 3.5)(рис. 3.6)(рис. 3.7).


```

vim nano 0.1 13.0 ИЗМЕНЕН
#include <stdio.h>
#include <stdlib.h>

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

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

```

Рис. 3.5: файл 2

```

./check_number
code=$?

if [ $code -eq 0 ]; then
    echo "Число равно нулю"
elif [ $code -eq 1 ]; then
    echo "Число больше нуля"
else
    echo "Число меньше нуля"
fi

```

Рис. 3.6: файл 2.1

```

[akabronina@vbox lab13]$ chmod +x 131.sh
[akabronina@vbox lab13]$ ./131.sh
./131.sh: строка 3: ./check_number: Нет такого файла или каталога
Число меньше нуля

```

Рис. 3.7: файл 2.2

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)(рис. 3.8).(рис. 3.9).

```
GNU nano 8.1 132.sh Изме
#!/bin/bash

N=$1

for ((i = 1; i <= N; i++)); do
    filename="${i}.tmp"
    if [ -f "$filename" ]; then
        echo "Удаляю существующий файл $filename"
        rm "$filename"
    fi
    echo "Создаю файл $filename"
    touch "$filename"
done
```

Рис. 3.8: файл 3

```
[akabronina@vbox lab13]$ nano 132.sh
[akabronina@vbox lab13]$ chmod +x 132.sh
[akabronina@vbox lab13]$ ./132.sh 4
Создаю файл 1.tmp
Создаю файл 2.tmp
Создаю файл 3.tmp
Создаю файл 4.tmp
[akabronina@vbox lab13]$ ls
13      13.c  2.tmp  input.txt  result.txt
131.sh  13.sh  3.tmp  presentation
132.sh  1.tmp  4.tmp  report
[akabronina@vbox lab13]$
```

Рис. 3.9: файл 3.1

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`). (рис. 3.10). (рис. 3.11).

```
GNU nano 8.1      133.sh      Изменён
#!/bin/bash

dir=$1
archive_name="recent_files_$(date +%Y%m%d).tar.gz"

find "$dir" -type f -mtime -7 | tar -czf "$archive_name" -T -
echo "Создан архив: $archive_name"
```

Рис. 3.10: файл 4

```
[akabronina@vbox lab13]$ ./133.sh ./report
Создан архив: recent_files_20250502.tar.gz
```

Рис. 3.11: файл 4.1

4 Выводы

Изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкции и циклов.