

Лабораторная работа 11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Паласиос Ф.

20 апреля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Паласиос Фелипе
- студент группы НКАбд - 04 - 22
- Российский университет дружбы народов

Вводная часть

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: – `-iinputfile` — прочитать данные из указанного файла; – `-ooutputfile` — вывести данные в указанный файл; – `-rшаблон` — 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`.

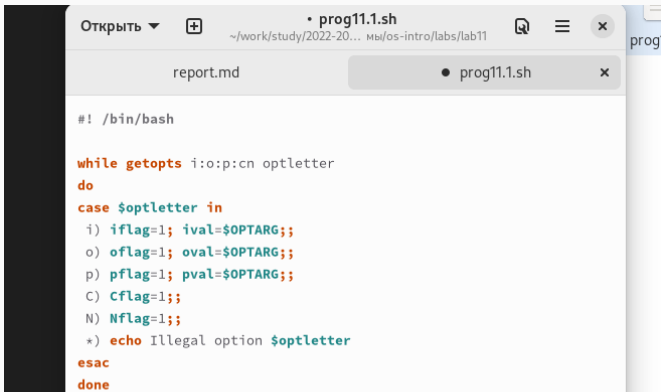
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Команд- ный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.

3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до `%` (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).

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`).

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`



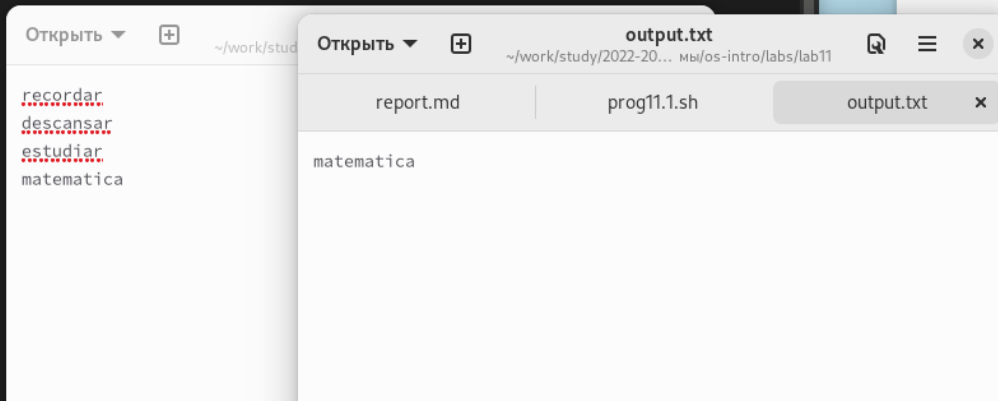
```
Открыть ▾ + • prog11.1.sh
~/work/study/2022-20... мы/os-intro/labs/lab11

report.md • prog11.1.sh x

#!/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
```

```
[palaciosfelipe@palaciosfelipe lab11]$ bash prog11.1.sh -p ma -i input.txt -o output.txt -c -n  
Illegal option c  
[palaciosfelipe@palaciosfelipe lab11]$
```



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

Результаты

The screenshot displays a development environment with three overlapping windows:

- prog12.c**: A C source file with the following code:

```
#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 {
        exit(2);
    }
}
```
- prog11.2.sh - GNU Emacs at**: A shell script with the following code:

```
#!/bin/bash

gcc -o cprog prog12.c
./cprog
case $? in
0) echo "Число равно 0";;
2) echo "Число меньше 0";;
1) echo "Число больше 0";;
esac
```
- Terminal**: A terminal window showing the execution of the script and the program:

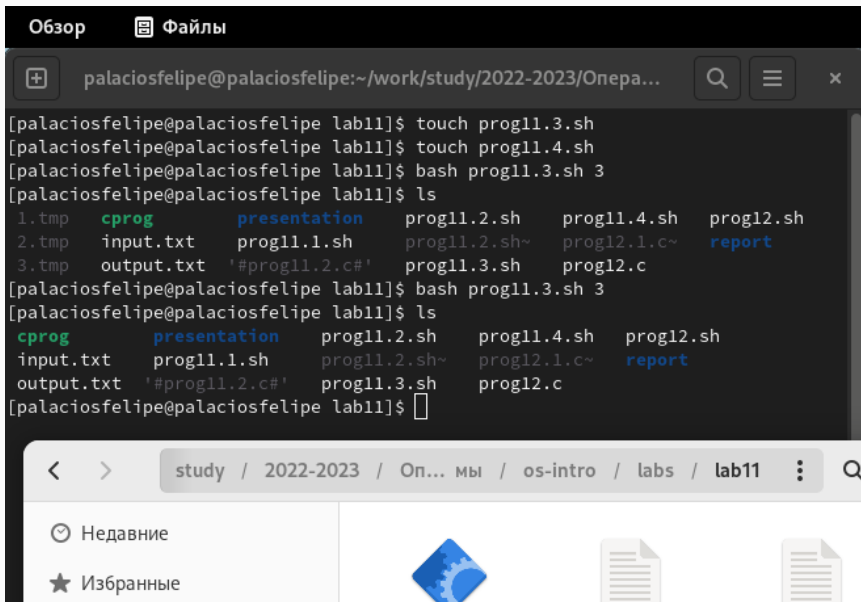
```
palaciosfelipe@palaciosfelipe:~/work/study/2022-2023/Онепа...
[palaciosfelipe@palaciosfelipe lab11]$ touch prog12.sh
[palaciosfelipe@palaciosfelipe lab11]$ touch prog12.c
[palaciosfelipe@palaciosfelipe lab11]$ bash prog11.2.sh
Введите число: 0
Число равно 0
[palaciosfelipe@palaciosfelipe lab11]$ 2
bash: 2: команда не найдена...
[palaciosfelipe@palaciosfelipe lab11]$ bash prog11.2.sh
Введите число: -2
Число меньше 0
[palaciosfelipe@palaciosfelipe lab11]$ bash prog11.2.sh
Введите число: 98
Число больше 0
```

3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до `%` (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)

The screenshot shows a terminal window and a file manager interface. The terminal window, titled "Обзор" and "Файлы", displays the following commands and output:

```
[palaciosfelipe@palaciosfelipe lab11]$ touch prog11.3.sh
[palaciosfelipe@palaciosfelipe lab11]$ touch prog11.4.sh
[palaciosfelipe@palaciosfelipe lab11]$ bash prog11.3.sh 3
[palaciosfelipe@palaciosfelipe lab11]$ ls
1.tmp      cprog      presentation  prog11.2.sh  prog11.4.sh  prog12.sh
2.tmp      input.txt   prog11.1.sh   prog11.2.sh~ prog12.1.c~  report
3.tmp      output.txt  '#prog11.2.c#' prog11.3.sh  prog12.c
```

The file manager shows the directory structure: `До...пка / work / study / 2022-2023 / Оп...мы / о...`. The left sidebar lists "Недавние", "Избранные", "Домашняя папка", and "Видео". The main area displays three files: `1.tmp`, `2.tmp`, and `3.tmp`.



The image shows a terminal window and a file manager interface. The terminal window has a title bar with "Обзор" and "Файлы" tabs. The terminal content shows a user at a shell prompt in a directory named "lab11". The user runs several commands: "touch prog11.3.sh", "touch prog11.4.sh", "bash prog11.3.sh 3", and "ls". The "ls" command output shows a directory listing with files like "1.tmp", "2.tmp", "3.tmp", "input.txt", "output.txt", "prog11.1.sh", "prog11.2.sh", "prog11.2.sh~", "prog11.3.sh", "prog11.4.sh", "prog12.1.c~", "prog12.c", and "report". The file manager interface below the terminal shows a breadcrumb path: "study / 2022-2023 / Он... мы / os-intro / labs / lab11". It also has a sidebar with "Недавние" and "Избранные" sections, and a main area with icons for a folder, a document, and a report.

```
Обзор  Файлы
palaciosfelipe@palaciosfelipe:~/work/study/2022-2023/Онепа...
[palaciosfelipe@palaciosfelipe lab11]$ touch prog11.3.sh
[palaciosfelipe@palaciosfelipe lab11]$ touch prog11.4.sh
[palaciosfelipe@palaciosfelipe lab11]$ bash prog11.3.sh 3
[palaciosfelipe@palaciosfelipe lab11]$ ls
1.tmp  cprog  presentation  prog11.2.sh  prog11.4.sh  prog12.sh
2.tmp  input.txt  prog11.1.sh  prog11.2.sh~  prog12.1.c~  report
3.tmp  output.txt  '#prog11.2.c#'  prog11.3.sh  prog12.c
[palaciosfelipe@palaciosfelipe lab11]$ bash prog11.3.sh 3
[palaciosfelipe@palaciosfelipe lab11]$ ls
cprog  presentation  prog11.2.sh  prog11.4.sh  prog12.sh
input.txt  prog11.1.sh  prog11.2.sh~  prog12.1.c~  report
output.txt  '#prog11.2.c#'  prog11.3.sh  prog12.c
[palaciosfelipe@palaciosfelipe lab11]$
```

< > study / 2022-2023 / Он... мы / os-intro / labs / lab11

Недавние

Избранные

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`)

```
Открыть ▾ + prog11.4.sh
~/work/study/2022-20... мы/os-intro/labs/lab11

report.md | prog12.c | prog11.3.sh | prog11.4.sh x

#!/bin/bash

find $* -mtime -7 -mtime +0 -type f >FILES.txt
tar -cf archive.tar -T FILES.txt

palaciosfelipe@palaciosfelipe:~/work/study/2022-2023/Онепа...
palaciosfelipe@palaciosfelipe:~/w... x palaciosfelipe@palaciosfelipe:~/w... x ▾

prog11.4.sh: строка 4: FILES.txt: команда не найдена
palaciosfelipe@palaciosfelipe lab11]$ bash prog11.4.sh home/palaciosfelipe/work
find: неверный аргумент «-7» у «-mtime»
prog11.4.sh: строка 4: FILES.txt: команда не найдена
tar: -: Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: Завершение работы с состоянием неисправности из-за возникших ошибок
palaciosfelipe@palaciosfelipe lab11]$ bash prog11.4.sh home/palaciosfelipe/work
find: 'home/palaciosfelipe/work': Нет такого файла или каталога
palaciosfelipe@palaciosfelipe lab11]$ bash prog11.4.sh home/work
find: 'home/work': Нет такого файла или каталога
palaciosfelipe@palaciosfelipe lab11]$ bash prog11.4.sh home
find: 'home': Нет такого файла или каталога
palaciosfelipe@palaciosfelipe lab11]$ bash prog11.4.sh /home/work
find: '/home/work': Нет такого файла или каталога
palaciosfelipe@palaciosfelipe lab11]$ bash prog11.4.sh /home/palaciosfelipe/work
palaciosfelipe@palaciosfelipe lab11]$ bash prog11.4.sh /home/palaciosfelipe/work/study
palaciosfelipe@palaciosfelipe lab11]$ ls
archive.tar  output.txt  prog11.2.sh  prog12.1.c~
cprog        presentation  prog11.2.sh~  prog12.c
FILES.txt    prog11.1.sh  prog11.3.sh  prog12.sh
input.txt    'prog11.2.c#' prog11.4.sh  report
```

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