

Программирование в командном процессоре ОС UNIX.

Ветвления и циклы

Световидова Полина НБИбд-04-22

Российский Университет Дружбы Народов


Цель работы

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

Выполнение лабораторной работы

Используя команды `getopts` `grep`, написала 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.

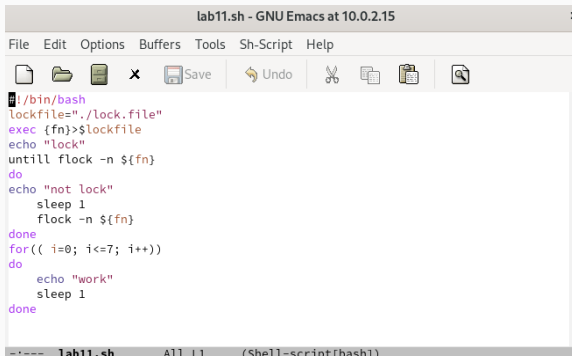
а затем ищет в указанном файле нужные строки, определяемые ключом -р.(рис. (fig:001?))(рис. (fig:002?))(рис. (fig:003?))



The image shows a code editor window with a tab labeled 'lab11.txt'. The editor displays four lines of text, each preceded by a line number (1-4). The text is: 'I get it cracking like a bad back', 'Bitch talking she the queen when she looking like a lab rat', 'I'm Angelina, you Jennifer', and 'Come on, bitch, you see where Brad at'. The text is highlighted with a yellow background. The editor's interface includes a top bar with 'Открыть' (Open), a plus icon, and the file name 'lab11.txt'. Below this is a tab bar with 'prog2.c', 'prog3.sh', 'prog4.sh' (selected), 'report.md', and 'lab11.txt'. The bottom of the editor shows the text being edited.

```
1 I get it cracking like a bad back
2 Bitch talking she the queen when she looking like a lab rat
3 I'm Angelina, you Jennifer
4 Come on, bitch, you see where Brad at
```

Рис. 1: Вставила в файл любой текст из интернета



```
lab11.sh - GNU Emacs at 10.0.2.15
File Edit Options Buffers Tools Sh-Script Help
[Icons: File, Folder, Save, Undo, Cut, Copy, Paste]

#!/bin/bash
lockfile="./lock.file"
exec {fn}>$lockfile
echo "lock"
until flock -n ${fn}
do
echo "not lock"
sleep 1
flock -n ${fn}
done
for(( i=0; i<=7; i++))
do
echo "work"
sleep 1
done

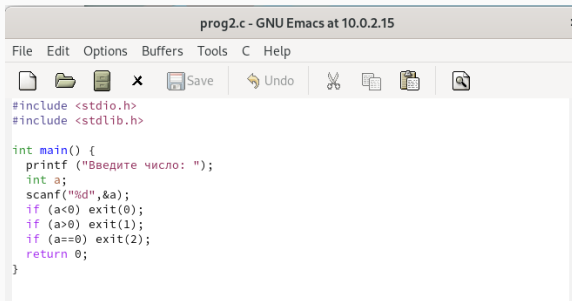
lab11.sh A11-11 (Shell-script fhash1)
```

Рис. 2: Пишу первый скрипт


```
[pmsvetovidova@10 ~]$ sudo -i
[sudo] пароль для pmsvetovidova:
[root@10 ~]# bash lab11.sh -i lab11.txt -o lab11-1.txt -p ice
[root@10 ~]# cat ~/lab11.txt
[root@10 ~]# dnf install cat
Последняя проверка окончания срока действия метаданных: 0:38:32 назад, Ср 12 апр 2023 18:38:16.
Нет соответствия аргументу: cat
Ошибка: Совпадений не найдено: cat
[root@10 ~]# ./lab11.sh -i ~/lab11.txt -o ~/lab11.txt -p it -C -n
[root@10 ~]# cat ~/lab11-1.txt
cat: /root/lab11-1.txt: Нет такого файла или каталога
[root@10 ~]# ./lab11.sh -i ~/lab11.txt -o ~/lab11.txt -p it -n
```

Рис. 3: Проверяю в терминале

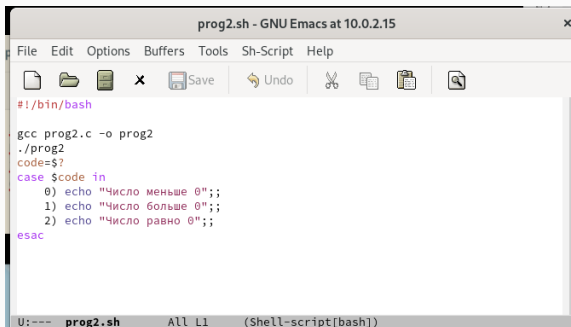
Написала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.(рис. (fig:004?))(рис. (fig:005?))(рис. (fig:006?))



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

int main() {
    printf ("Введите число: ");
    int a;
    scanf ("%d", &a);
    if (a<0) exit(0);
    if (a>0) exit(1);
    if (a==0) exit(2);
    return 0;
}
```

Рис. 4: Пишу новый скрипт-на языке Си



```
#!/bin/bash

gcc prog2.c -o prog2
./prog2
code=$?
case $code in
  0) echo "число меньше 0";;
  1) echo "число больше 0";;
  2) echo "число равно 0";;
esac
```

U:--- prog2.sh All L1 (Shell-script[bash])

Рис. 5: Пишу еще один скрипт

```
./prog2.sh: строка 4: ./prog2: Нет такого файла или каталога
[root@10 saeryh]# ./prog2.sh
Введите число: 3
Число больше 0
[root@10 saeryh]# ./prog2.sh
Введите число: 5
Число больше 0
[root@10 saeryh]# ./prog2.sh
Введите число: -2
Число меньше 0
[root@10 saeryh]# ./prog2.sh
Введите число: 0
Число равно 0
[root@10 saeryh]# chmod +x lab11.sh
```

Рис. 6: Проверяю все в терминале

Написала 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. (fig:007?))(рис. (fig:008?))

Рис. 7: Пишу новый скрипт

```
[root@10 saeryh]# chmod +x lab11.sh
chmod: невозможно получить доступ к 'lab11.sh': Нет такого файла или каталога
[root@10 saeryh]# chmod +x *.sh
[root@10 saeryh]# ./prog3.sh -c a#.txt 3
[root@10 saeryh]# ls
prog2 prog2.c prog2.c~ prog2.sh prog3.sh
[root@10 saeryh]# touch a1
[root@10 saeryh]# touch a1.txt
[root@10 saeryh]# touch a2.txt
[root@10 saeryh]# touch a3.txt
[root@10 saeryh]# ./prog3.sh -r a#.txt 3
[root@10 saeryh]#
```

Рис. 8: Проверяю его в терминале

Написала командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировала его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовал команду find).(рис. (fig:009?))(рис. (fig:010?))

```
1 #!/bin/bash
2
3 files=$(find ./ -maxdepth 1 -mtime -7)
4 listing=""
5 for file in "$files" ; do
6     file=$(echo "$file" | cut -c 3-)
7     listing+ "$listing $file"
8 done
9 dir=$|
```

Рис. 9: Пишу новый скрипт

```
tar: Error is not recoverable: exiting now
[root@10 saeryh]# tar -tf lab11.tar
tar: lab11.tar: Функция open завершилась с ошибкой: Нет такого файла
tar: Error is not recoverable: exiting now
[root@10 saeryh]# sudo /home/pmsvetovidova/saeryh/prog4.sh
[root@10 saeryh]# sudo /home/pmsvetovidova/saeryh/prog4.sh
[root@10 saeryh]# chmod +x prog4.sh
[root@10 saeryh]# chmod +x prog4.sh
[root@10 saeryh]# chmod +x *.sh
[root@10 saeryh]# mc

[root@10 saeryh]# ./prog4.sh

[root@10 saeryh]# sudo /home/pmsvetovidova/saeryh/prog4.sh
[root@10 saeryh]#
```

Рис. 10: Проверил его в терминале

Выводы

В ходе выполнения лабораторной работы №11 я изучила основы программирования в оболочке ОС Linux, а так же научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов