

Лабораторная работа №11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Ежова А. М.

19 апреля 2023

Российский университет дружбы народов, Москва, Россия

Факультет физико-математических и естественных наук

Информация

- Ежова Алиса Михайловна
- студентка НБИбд-01-22
- Российский университет дружбы народов
- <https://github.com/AlisaEzhova>

Вводная часть

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

- Процессор **pandoc** для входного формата Markdown
- Результирующие форматы
 - pdf
 - html
- Автоматизация процесса создания: **Makefile**

Создание презентации

- Pandoc: преобразователь текстовых файлов
- Сайт: <https://pandoc.org/>
- Репозиторий: <https://github.com/jgm/pandoc>

- Использование LaTeX
- Пакет для презентации: beamer
- Тема оформления: **metropolis**

```
slide_level: 2  
aspectratio: 169  
section-titles: true  
theme: metropolis
```

- Используется фреймворк `reveal.js`
- Используется тема `beige`

- Тема задаётся в файле `Makefile`

```
REVEALJS_THEME = beige
```

Результаты

- Полученный **pdf**-файл можно демонстрировать в любой программе просмотра **pdf**
- Полученный **html**-файл содержит в себе все ресурсы: изображения, **css**, скрипты

- 1) 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 `-r`:

```
#!/bin/bash cflag=0; nflag=0; while getopts i:o:p:C:n opt do case opt in i) ival =OPTARG;; o)
oval=OPTARG;; p) pval =OPTARG;; C) cflag=1;; n) nflag=1;; esac done if [ $cflag -a $nflag ] then
grep -n $pval ival >oval elif test $cflag then grep $pval ival >oval elif test $nflag then grep -n
-i $pval ival >oval else grep -i $pval ival >oval fi
```



```

amezhova@dk2n24 ~ $ touch lab11_1.sh
amezhova@dk2n24 ~ $ chmod +x lab11_1.sh
amezhova@dk2n24 ~ $ ./lab11_1.sh
./lab11_1.sh: строка 16: $oval: неоднозначное перенаправление
amezhova@dk2n24 ~ $ ./lab11_1.sh -i text.txt -o fout.txt -p файлы -C -n
amezhova@dk2n24 ~ $ cat fout.txt
amezhova@dk2n24 ~ $

```

Рис. 1: Код

```

#!/bin/bash
cflag=0;
nflag=0;
while getopts i:o:p:C:n opt
do
case $opt in
i) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
grep -p $pval $ival>$oval

```

2) Написала сначала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем завершила программу при помощи функции `exit(n)`, передавая информацию о коде завершения в оболочку. Командный файл вызовет эту программу и, проанализировав с помощью команды `$?`, выдаст сообщение о том, какое число было введено:

```
#!/bin/bash gcc -c script2.c gcc -o script2 script2.c ./script2 case $? in 1) echo отрицательное;; 2) echo равно нулю;; 3) echo положительное;; esac
```

```
amezhova@dk2n24 ~ $ touch lab11_2.sh
amezhova@dk2n24 ~ $ chmod +x lab11_2.sh
amezhova@dk2n24 ~ $ ./lab11_2.sh
amezhova@dk2n24 ~ $
```

Рис. 3: Код 3

```
#!/bin/bash
gcc -c script2.c
gcc -o script2 script2.c
./script2
case $? in
1) echo отрицательное;;
```

3) Написала командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N:

```
#!/bin/bash let i=$1+1 while (( i-=1 )) do touch $i.tmp done let j=$2+1; while (( j-=1 )) do rm $j.tmp done
```

4) 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.

```
#!/bin/bash (find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar
```

- В ходе выполнения Лабораторной работы №11, я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.