

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Дисциплина: Архитектура компьютера и операционные системы

Ванюшкина Т.В.

08 May 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Ванюшкина Татьяна Валерьевна
- группа НКАбд-01-24
- Российский университет дружбы народов

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

- Курс: Архитектура компьютеров и операционные системы. Раздел “Операционные системы” (02.03.00, УГСН) (rudn.ru)

Создание презентации

Используя команды `getopts` `grep`, пишу 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

```
#!/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;;
```

Пишу 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

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

int main () {
    int n;
    printf ("enter number");
    scanf ("%d", &n);
    if (n > 0) {
        exit(1);
    }
}
```


Пишу 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

```
#!/bin/bash
for((i=1; i<=$*; i++))
do
if test -f "$i".tmp
then rm "$i".tmp
else touch "$i.tmp"
fi
done
```

Пишу 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).

```
#!/bin/bash
find $* -mtime -7 -mtime +0 -type f > FILES.txt
tar -cf archive.tar -T FILES.txt
```

(рис.4 (fig:004?))

{#fig:004}

Резулитаты

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

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

- Тема задаётся в файле **Makefile**

```
“make REVEALJS_THEME = beige
```