

Лабораторная работа №11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Шуплецов А. А.

22 апреля 2023

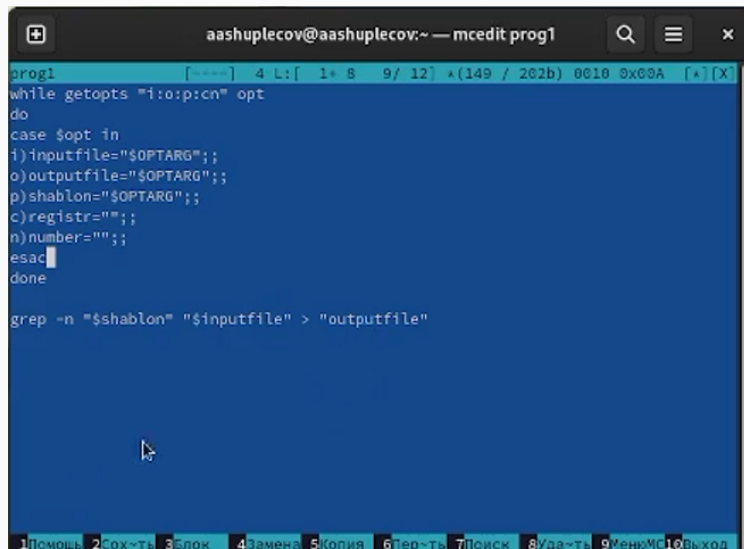
Российский университет дружбы народов, Москва, Россия

Информация

- Шуплецов Александр Андреевич
- студент ФФМиЕН
- Российский университет дружбы народов
- <https://github.com/winnralex>

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

Используя команды `getopts` `grep`, напомним командный файл, который анализирует командную строку с ключами.



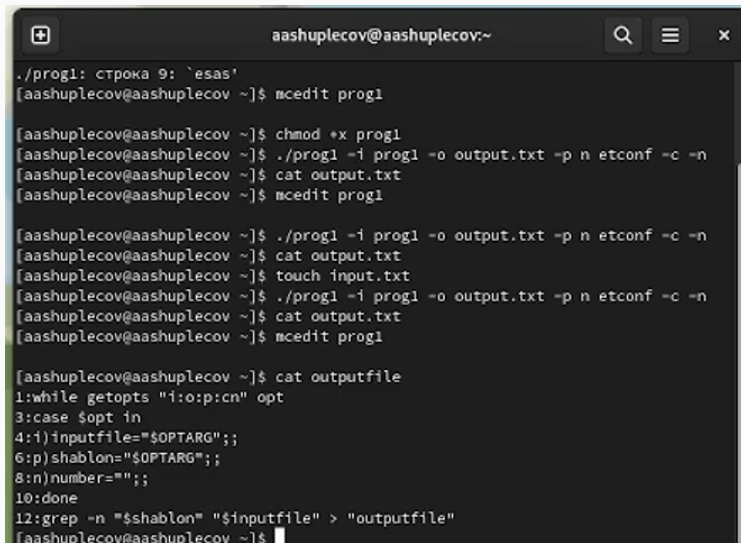
The screenshot shows a terminal window titled "aashuplecov@aashuplecov:~ — mcedit prog1". The script content is as follows:

```
prog1
[----] 4 L: [ 1+ 8 9/ 12] *(149 / 202b) 0010 0x00A [*] [X]
while getopts "i:o:p:cn" opt
do
case $opt in
i)inputfile="$OPTARG";;
o)outputfile="$OPTARG";;
p)shablon="$OPTARG";;
c)registr="";;
n)number="";;
esac
done

grep -n "$shablon" "$inputfile" > "outputfile"
```

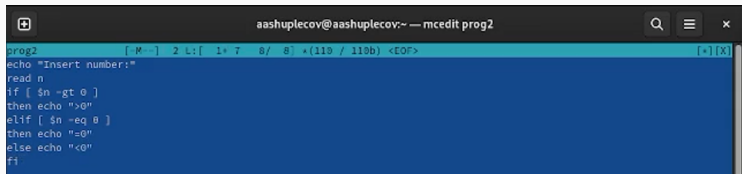
At the bottom of the terminal, there is a menu bar with the following items: 1Помощь, 2Сох-ть, 3Блок, 4Замена, 5Копия, 6Пер-ть, 7Поиск, 8Уда-ть, 9Меню, 10Выход.

Убедимся, что командный файл, анализирующий командную строку с ключами, работает.



```
aashuplecov@aashuplecov:~  
./prog1: строка 9: `esas'  
[aashuplecov@aashuplecov ~]$ mcedit prog1  
  
[aashuplecov@aashuplecov ~]$ chmod +x prog1  
[aashuplecov@aashuplecov ~]$ ./prog1 -i prog1 -o output.txt -p n etconf -c -n  
[aashuplecov@aashuplecov ~]$ cat output.txt  
[aashuplecov@aashuplecov ~]$ mcedit prog1  
  
[aashuplecov@aashuplecov ~]$ ./prog1 -i prog1 -o output.txt -p n etconf -c -n  
[aashuplecov@aashuplecov ~]$ cat output.txt  
[aashuplecov@aashuplecov ~]$ touch input.txt  
[aashuplecov@aashuplecov ~]$ ./prog1 -i prog1 -o output.txt -p n etconf -c -n  
[aashuplecov@aashuplecov ~]$ cat output.txt  
[aashuplecov@aashuplecov ~]$ mcedit prog1  
  
[aashuplecov@aashuplecov ~]$ cat outputfile  
1:while getopts "i:o:p:cn" opt  
3:case $opt in  
4:i)inputfile="$OPTARG";;  
6:p)shablon="$OPTARG";;  
8:n)number="";;  
10:done  
12:grep -n "$shablon" "$inputfile" > "outputfile"  
[aashuplecov@aashuplecov ~]$
```

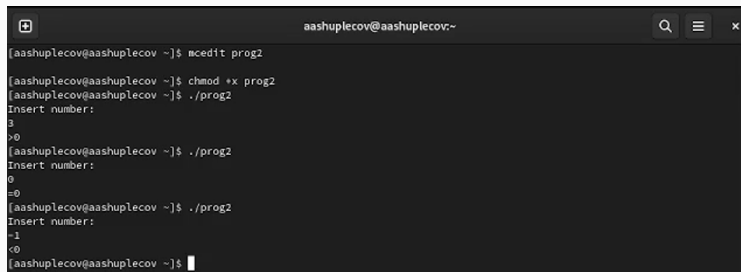
Напишем на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.



```
prog2 [~M--] 2 L:[ 1* 7 8/ 8] *(110 / 110b) <EOF> [*][X]  
echo "Insert number:"  
read n  
if [ $n -gt 0 ]  
then echo ">0"  
elif [ $n -eq 0 ]  
then echo "=0"  
else echo "<0"  
fi
```

Рис. 3: текст командного файла, выводящего число

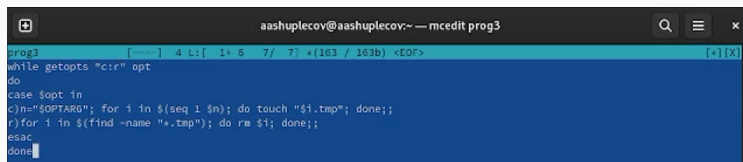
Убедимся, что программа, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю, работает.



```
aashuplecov@aashuplecov:~  
[aashuplecov@aashuplecov ~]$ mcedit prog2  
[aashuplecov@aashuplecov ~]$ chmod +x prog2  
[aashuplecov@aashuplecov ~]$ ./prog2  
Insert number:  
3  
>0  
[aashuplecov@aashuplecov ~]$ ./prog2  
Insert number:  
0  
=0  
[aashuplecov@aashuplecov ~]$ ./prog2  
Insert number:  
-1  
<0  
[aashuplecov@aashuplecov ~]$
```

Рис. 4: проверка командного файла, выводящего число

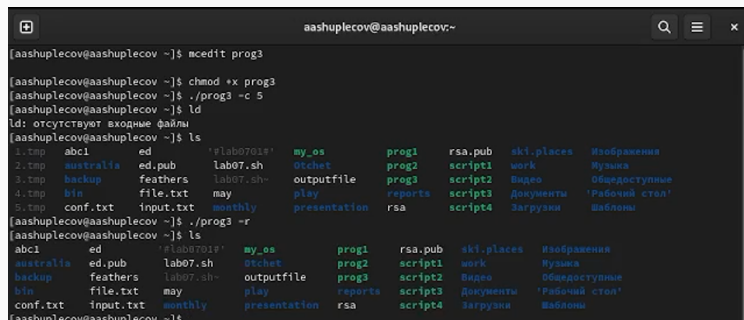
Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до [?] (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.).



```
prog3 [-----] 4 L: [ 1* 6 7/ 7] *(163 / 163b) <EOF> [ * ] [X]
while getopts "c:r" opt
do
case $opt in
c)n="$OPTARG"; for i in $(seq 1 $n); do touch "$i.tmp"; done;;
r)for i in $(find -name "*.tmp"); do rm $i; done;;
esac
done
```

Рис. 5: текст командного файла, создающего указанное число файлов

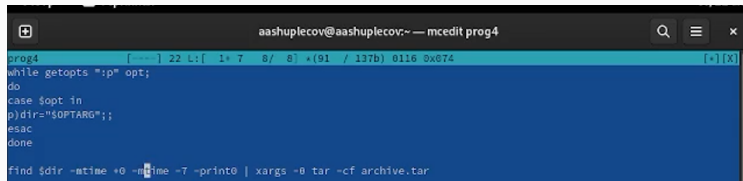
Убедимся, что командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до [?] (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.), работает.



```
aashuplecov@aashuplecov:~  
[aashuplecov@aashuplecov ~]$ mcedit prog3  
[aashuplecov@aashuplecov ~]$ chmod +x prog3  
[aashuplecov@aashuplecov ~]$ ./prog3 -c 5  
[aashuplecov@aashuplecov ~]$ ld  
ld: отсутствуют входные файлы  
[aashuplecov@aashuplecov ~]$ ls  
1.tmp  abc1      ed      'elab0701#'  my_os      prog1      rsa.pub  ski.places  Изображения  
2.tmp  australia ed.pub  lab07.sh     Otchet     prog2      script1  work        Музыка  
3.tmp  backup   feathers lab07.sh-    outputfile prog3      script2  Видео       Общедоступные  
4.tmp  bin      file.txt may        play       reports    script3   Документы  'Рабочий стол'  
5.tmp  conf.txt input.txt monthly    presentation rsa        script4   Загрузки   Шаблоны  
[aashuplecov@aashuplecov ~]$ ./prog3 -r  
[aashuplecov@aashuplecov ~]$ ls  
abc1      ed      'elab0701#'  my_os      prog1      rsa.pub  ski.places  Изображения  
australia ed.pub  lab07.sh     Otchet     prog2      script1  work        Музыка  
backup    feathers lab07.sh-    outputfile prog3      script2  Видео       Общедоступные  
bin       file.txt may        play       reports    script3   Документы  'Рабочий стол'  
conf.txt  input.txt monthly    presentation rsa        script4   Загрузки   Шаблоны  
[aashuplecov@aashuplecov ~]$
```

Рис. 6: проверка командного файла, создающего указанное число файлов

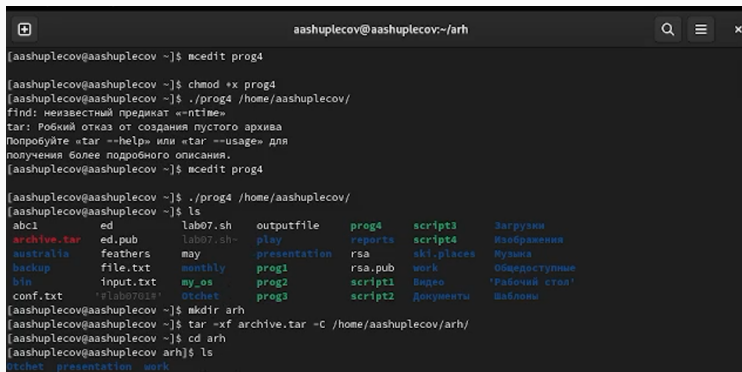
Напишем командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории, которые использовались менее недели тому назад.



```
prog4 [----] 22 L: [ 1* 7 8/ 8] *(91 / 137b) 0116 0x074 [*][X]
while getopts ":p" opt;
do
case $opt in
p)dir="$OPTARG";;
esac
done
find $dir -mtime +0 -mtime -7 -print0 | xargs -0 tar -cf archive.tar
```

Рис. 7: текст командного файла, который запаковывает в архив недавно использованные файлы

Убедимся, что командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории, которые использовались менее недели тому назад, работает.



```
aashuplecov@aashuplecov:~/arh
[aashuplecov@aashuplecov ~]$ mcedit prog4

[aashuplecov@aashuplecov ~]$ chmod +x prog4
[aashuplecov@aashuplecov ~]$ ./prog4 /home/aashuplecov/
find: неизвестный предикат «=mtime»
tar: Ровкий отказ от создания пустого архива
Попробуйте «tar --help» или «tar --usage» для
получения более подробного описания.
[aashuplecov@aashuplecov ~]$ mcedit prog4

[aashuplecov@aashuplecov ~]$ ./prog4 /home/aashuplecov/
[aashuplecov@aashuplecov ~]$ ls
abc1      ed      lab07.sh  outputfile  prog4      script3     Загрузки
archive.tar  ed.pub  lab07.sh~ play        reports    script4     Изображения
australia  feathers may      presentation  rsa      ski.places  Музыка
backup     file.txt monthly  prog1        rsa.pub  work        Общедоступные
bin        input.txt my_os    prog2        script1   Видео      'Рабочий стол'
conf.txt   'lab0701*' 0tchet  prog3        script2   Документы  Шаблоны

[aashuplecov@aashuplecov ~]$ mkdir arh
[aashuplecov@aashuplecov ~]$ tar -xf archive.tar -C /home/aashuplecov/arh/
[aashuplecov@aashuplecov ~]$ cd arh
[aashuplecov@aashuplecov arh]$ ls
0tchet presentation work
```

Рис. 8: проверка командного файла, который запаковывает в архив недавно использованные файлы

Я изучил основы программирования в оболочке ОС UNIX/Linux, научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Кулябов Д.С. “Материалы к лабораторным работам”