

Лабораторная работа-11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Арсоева Залина НБИбд-01-21

Содержание

Цель работы

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

Задания

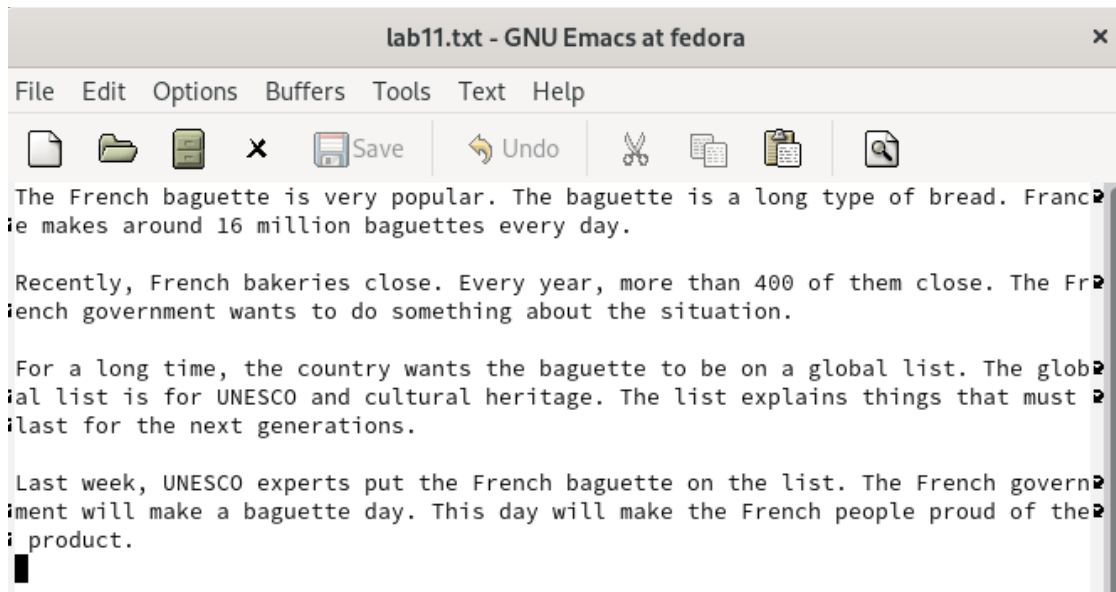
1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `-i` inputfile — прочитать данные из указанного файла; `-o` outputfile — вывести данные в указанный файл; `-r` шаблон — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-r`.
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.
3. Написать 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, передается 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. Используя команды `getopts` `grep`, написала 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`. (рис. [-@fig:001])(рис. [-@fig:002])(рис. [-@fig:003])



Вставила в файл любой текст из интернета

```
report.md • $ lab11.sh x C prog2.c $ prog2.sh $ prog3.sh □ ...
home > zalinaarsoeva > $ lab11.sh
1  #!/bin/bash
2  iflag=0; oflag=0; pflag=0; cflag=0; nflag=0;
3  while getopts i:o:p:C:n optletter
4  do case $optletter in
5      i) iflag=1; ival=$OPTARG;;
6      O) oflag=1; oval=$OPTARG;;
7      p) pflag=1; pval=$OPTARG;;
8      C) cflag=1;;
9      n) nflag=1;;
10     *) echo illegal option $optletter
11     esac
12 done
13 if (($pflag==0))
14 then echo "Шаблон найден"
15 else
16     if (($iflag==0))
17     then echo "Файл не найден"
18     else
19         if (($oflag==0))
20         then if (($cflag==0))
21             then if (($nflag==0))
22                 then grep $pval $ival
23                 else grep -n $pval $ival
24                 fi
25             else if (($nflag==0))
26                 then grep -i $pval $ival
27                 else grep -i -n $pval $ival
28                 fi
29             fi
30         else if (($cflag==0))
31             then if (($nflag==0))
32                 then grep $pval $ival > $oval
33                 else grep -n $pval $ival > $oval
34                 fi
35             else if (($nflag==0))
36                 then grep -i $pval $ival > $oval
37                 else grep -i -n $pval $ival > $oval
38                 fi
39             fi
40         fi
41         fi
```

Пишу первый скрипт

```

[zalinaarsoeva@fedora ~]$ cat ~/lab11.txt
The French baguette is very popular. The baguette is a long type of bread. France makes around 16 million baguettes e
very day.

Recently, French bakeries close. Every year, more than 400 of them close. The French government wants to do something
about the situation.

For a long time, the country wants the baguette to be on a global list. The global list is for UNESCO and cultural he
ritage. The list explains things that must last for the next generations.

Last week, UNESCO experts put the French baguette on the list. The French government will make a baguette day. This d
ay will make the French people proud of the product.
[zalinaarsoeva@fedora ~]$ ./lab11.sh -i ~/lab11.txt -o ~/lab11-1.txt -p it -C -n
illegal option o
Recently, French bakeries close. Every year, more than 400 of them close. The French government wants to do something
about the situation.

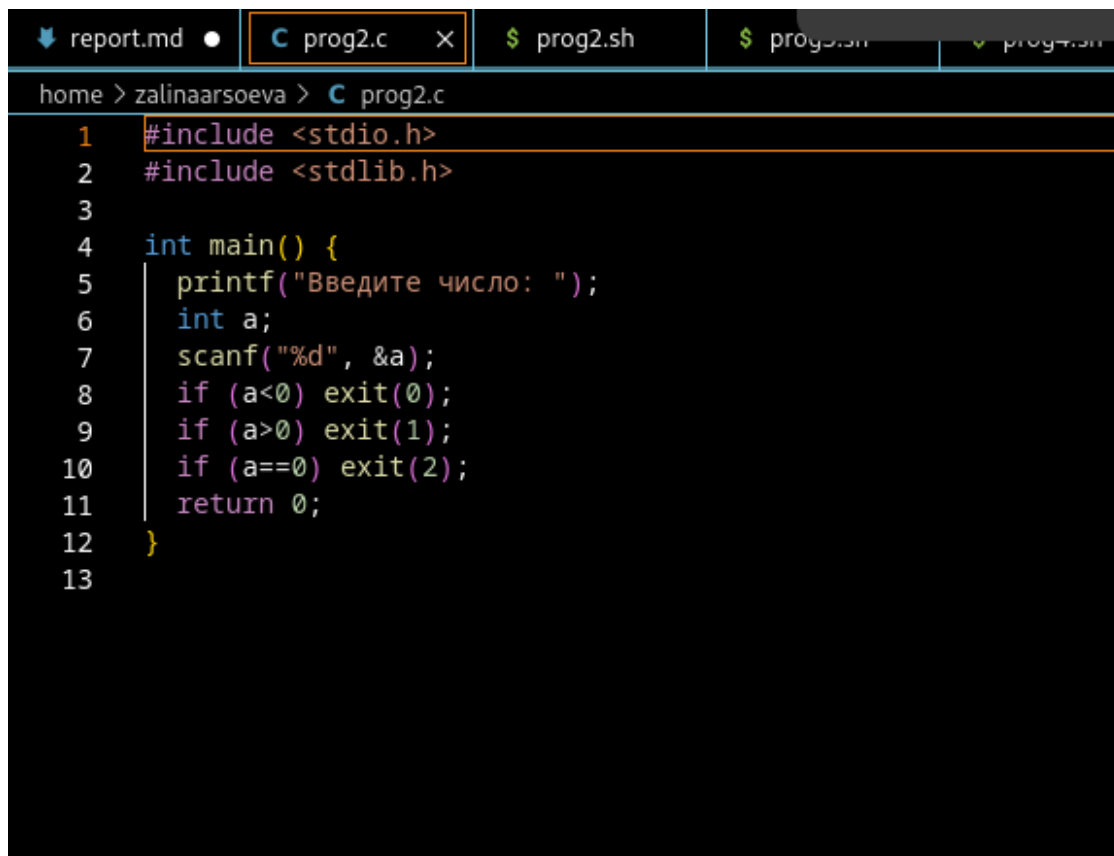
For a long time, the country wants the baguette to be on a global list. The global list is for UNESCO and cultural he
ritage. The list explains things that must last for the next generations.
[zalinaarsoeva@fedora ~]$ ./lab11.sh -i ~/lab11.txt -o ~/lab11-1.txt -p it -C -n
bash: ./lab11.sh: /bin/dash: неверный интерпретатор: Нет такого файла или каталога
[zalinaarsoeva@fedora ~]$ ./lab11.sh -i ~/lab11.txt -o ~/lab11-1.txt -p it -C -n
illegal option o
Recently, French bakeries close. Every year, more than 400 of them close. The French government wants to do something
about the situation.

For a long time, the country wants the baguette to be on a global list. The global list is for UNESCO and cultural he
ritage. The list explains things that must last for the next generations.
[zalinaarsoeva@fedora ~]$ cat ~/lab11-1.txt
cat: /home/zalinaarsoeva/lab11-1.txt: Нет такого файла или каталога
[zalinaarsoeva@fedora ~]$ cat ~/lab11-1.txt
cat: /home/zalinaarsoeva/lab11-1.txt: Нет такого файла или каталога
[zalinaarsoeva@fedora ~]$ ./lab11.sh -i ~/lab11.txt -o ~/lab11-1.txt -p it -n
illegal option o
3:Recently, French bakeries close. Every year, more than 400 of them close. The French government wants to do somethi
ng about the situation.
5:For a long time, the country wants the baguette to be on a global list. The global list is for UNESCO and cultural
heritage. The list explains things that must last for the next generations.
[zalinaarsoeva@fedora ~]$ ./lab11.sh -i ~/lab11.txt -C -n
Шаблон найден
[zalinaarsoeva@fedora ~]$ ./lab11.sh -o ~/lab11.txt -p it -n
illegal option o
Файл не найден

```

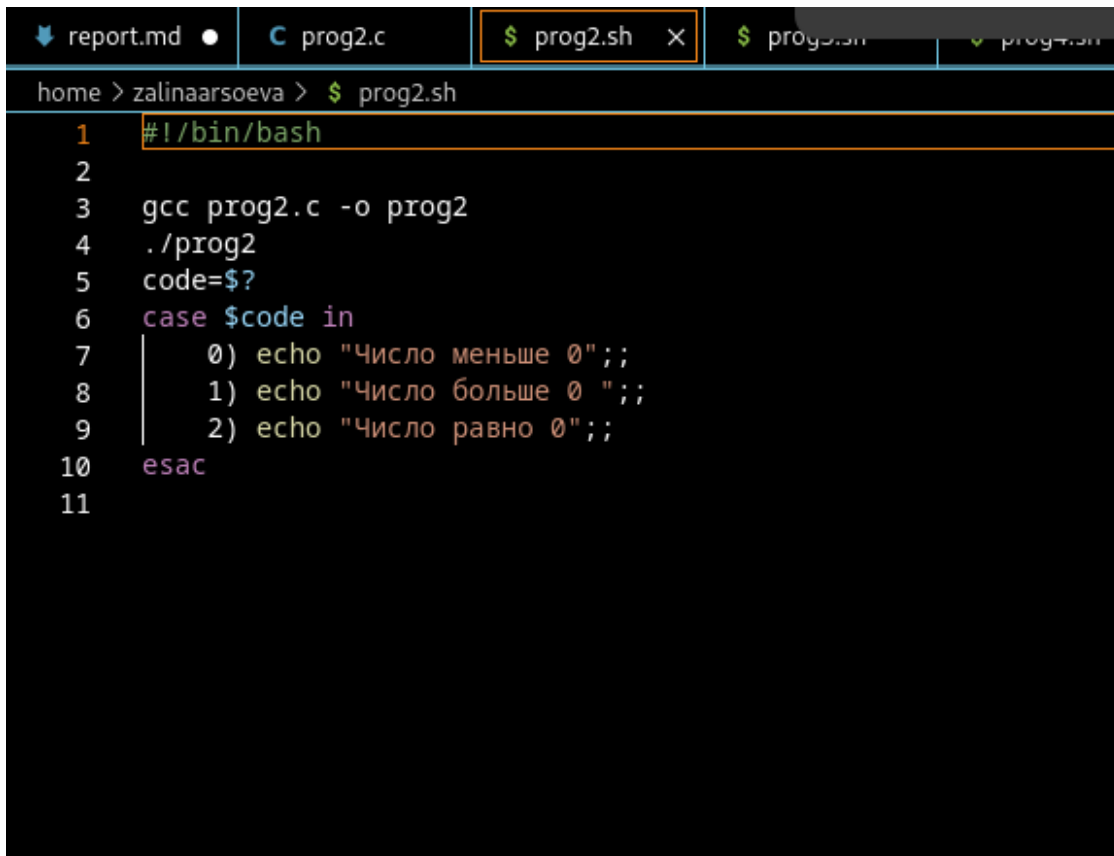
Проверяю в терминале

- 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])



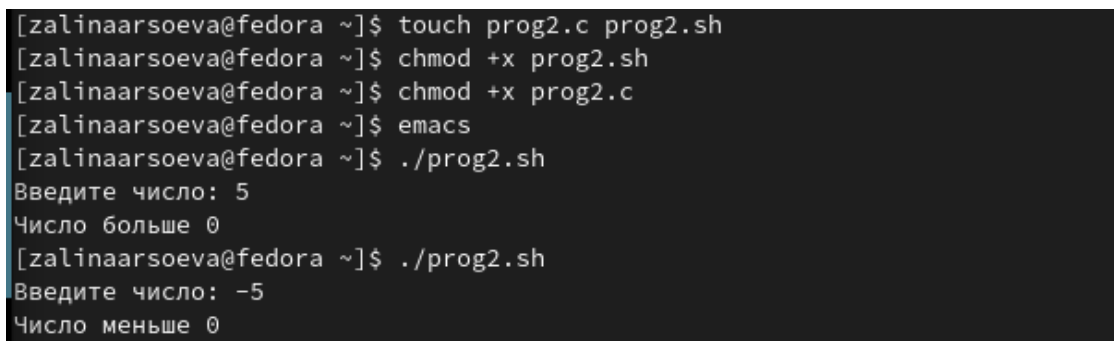
```
home > zalinaarsoeva > C prog2.c
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      printf("Введите число: ");
6      int a;
7      scanf("%d", &a);
8      if (a<0) exit(0);
9      if (a>0) exit(1);
10     if (a==0) exit(2);
11     return 0;
12 }
13
```

Пишу новый скрипт-на языке Си



```
report.md ● prog2.c $ prog2.sh × $ prog2.sh
home > zalinaarsoeva > $ prog2.sh
1  #!/bin/bash
2
3  gcc prog2.c -o prog2
4  ./prog2
5  code=$?
6  case $code in
7      0) echo "Число меньше 0";;
8      1) echo "Число больше 0";;
9      2) echo "Число равно 0";;
10 esac
11
```

Пишу еще один скрипт



```
[zalinaarsoeva@fedora ~]$ touch prog2.c prog2.sh
[zalinaarsoeva@fedora ~]$ chmod +x prog2.sh
[zalinaarsoeva@fedora ~]$ chmod +x prog2.c
[zalinaarsoeva@fedora ~]$ emacs
[zalinaarsoeva@fedora ~]$ ./prog2.sh
Введите число: 5
Число больше 0
[zalinaarsoeva@fedora ~]$ ./prog2.sh
Введите число: -5
Число меньше 0
```

Проверяю все в терминале

3. 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])

```
report.md • prog2.c $ prog2.sh $ prog3.sh x $ prog4.sh
home > zalinaarsoeva > $ prog3.sh
1  #!/bin/bash
2
3  opt=$1;
4  form=$2;
5  num=$3;
6  function Files() {
7      for ((i=1; i<=$num; i++)) do
8          file=$(echo $form | tr '#' "$i")
9          if [ $opt == "-r" ]
10         then
11             rm -f $file
12         elif [ $opt == "-c" ]
13         then
14             touch $file
15         fi
16     done
17 }
18 Files
19
```

Пишу новый скрипт

```

[zalinaarsoeva@fedora ~]$ touch prog3.sh
[zalinaarsoeva@fedora ~]$ chmod +x *.sh
[zalinaarsoeva@fedora ~]$ ls
backup      full.txt~   lab11.sh~  prog2.sh~  script3.sh~ temp      Загрузки
conf.txt    lab        lab11.txt  prog3.sh   script4.sh~ text.txt  Изображения
file.txt    '#lab07.sh~' prog2      reports    script4.sh~ ttt       Музыка
full        lab07.sh   prog2.c~   script2.sh~ script.sh~  work     Общедоступные
'#full.txt#' lab07.sh~  prog2.c~   script2.sh~ script.sh~  Видео    'Рабочий стол'
full.txt    lab11.sh   prog2.sh   script3.sh~ tap        Документы  Шаблоны
[zalinaarsoeva@fedora ~]$ emacs
[zalinaarsoeva@fedora ~]$ ./prog3.sh
./prog3.sh: строка 6: синтаксическая ошибка рядом с неожиданным маркером «(»
./prog3.sh: строка 6: `function Files() {'
[zalinaarsoeva@fedora ~]$ ./prog3.sh
./prog3.sh: строка 7: ((: i<=: синтаксическая ошибка: ожидается операнд (неверный маркер «<=»)
[zalinaarsoeva@fedora ~]$ ./prog3.sh
./prog3.sh: строка 7: ((: i<=: синтаксическая ошибка: ожидается операнд (неверный маркер «<=»)
[zalinaarsoeva@fedora ~]$ ./prog3.sh
[zalinaarsoeva@fedora ~]$ ./prog3.sh -c a#.txt 3
[zalinaarsoeva@fedora ~]$ ls
a1.txt      full        lab07.sh   prog2.c~   reports    script4.sh~ ttt       Музыка
a2.txt      '#full.txt#' lab07.sh~  prog2.c~   script2.sh~ script.sh~  work     Общедоступные
a3.txt      full.txt    lab11.sh   prog2.sh   script2.sh~ script.sh~  Видео    'Рабочий стол'
backup      full.txt~   lab11.sh~  prog2.sh~  script3.sh~ tap        Документы  Шаблоны
conf.txt    lab        lab11.txt  prog3.sh   script3.sh~ temp      Загрузки
file.txt    '#lab07.sh~' prog2      prog3.sh~  script4.sh~ text.txt  Изображения
[zalinaarsoeva@fedora ~]$ ./prog3.sh -r a#.txt 3
[zalinaarsoeva@fedora ~]$ ls
backup      full.txt~   lab11.sh~  prog2.sh~  script3.sh~ tap        Документы  Шаблоны
conf.txt    lab        lab11.txt  prog3.sh   script3.sh~ temp      Загрузки
file.txt    '#lab07.sh~' prog2      prog3.sh~  script4.sh~ text.txt  Изображения
full        lab07.sh   prog2.c~   reports    script4.sh~ ttt       Музыка
'#full.txt#' lab07.sh~  prog2.c~   script2.sh~ script.sh~  work     Общедоступные
full.txt    lab11.sh   prog2.sh   script2.sh~ script.sh~  Видео    'Рабочий стол'

```

Проверяю его в терминале

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 (использовал команду find).(рис. [-@fig:009])(рис. [-@fig:010])

```

report.md ● C prog2.c $ prog2.sh $ prog3.sh $ prog4.sh >
home > zalinaarsoeva > $ prog4.sh
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 $file"
8  done
9  dir=$(basename $(pwd))
10 tar -cvf $dir.tar $listing

```

Пишу новый скрипт


```
zalinaarsoeva@fedora:~$ sudo ~/prog4.sh
sudo] пароль для zalinaarsoeva:
config/
config/gnome-initial-setup-done
config/dconf/
config/dconf/user
config/goa-1.0/
config/user-dirs.dirs
config/user-dirs.locale
config/ibus/
config/ibus/bus/
config/ibus/bus/ca4fecca252c4c9a9da3891c7652378b-unix-wayland-0
config/evolution/
config/evolution/sources/
config/evolution/sources/system-proxy.source
config/pulse/
config/pulse/cookie
config/gtk-3.0/
config/gtk-3.0/bookmarks
config/.gsd-keyboard.settings-ported
config/abrt/
config/yelp/
config/yelp/yelp.cfg
config/gnome-session/
config/gnome-session/saved-session/
config/nautilus/
config/nautilus/search-metadata
config/enchant/
config/enchant/ru_RU.dic
config/enchant/ru_RU.exc
config/gtk-4.0/
config/mc/
config/mc/mcedit/
config/mc/ini
config/mc/panels.ini
config/mc/filehighlight.ini
config/mc/menu
config/mc/mc.ext
config/libreoffice/
config/libreoffice/4/
config/libreoffice/4/user/
config/libreoffice/4/user/config/
config/libreoffice/4/user/config/javasettings_Linux_X86_64.xml
config/libreoffice/4/user/config/autotbl.fmt
config/libreoffice/4/user/config/soffice.cfg/
config/libreoffice/4/user/config/soffice.cfg/modules/
config/libreoffice/4/user/config/soffice.cfg/modules/swriter/
config/libreoffice/4/user/config/soffice.cfg/modules/swriter/menubar/
```

Проверил его в терминале

Вывод:

Изучила основы программирования в оболочке ОС UNIX, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Ответы на контрольные вопросы:

1. Команда `getopts` является встроенной командой командной оболочки `bash`, предназначенной для разбора параметров сценариев. Она обрабатывает исключительно однобуквенные параметры как с аргументами, так и без них и этого вполне достаточно для передачи сценариям любых входных данных.

2. При генерации имен используют метасимволы:
 - произвольная (возможно пустая) последовательность символов; ? один произвольный символ; [...] любой из символов, указанных в скобках перечислением и/или с указанием диапазона; cat f* выдаст все файлы каталога, начинающиеся с "f"; cat f выдает все файлы, содержащие "f"; cat program.? выдаст файлы данного каталога с однобуквенными расширениями, скажем "program.c" и "program.o", но не выдаст "program.com"; cat [a-d]* выдаст файлы, которые начинаются с "a", "b", "c", "d". Аналогичный эффект дадут и команды "cat [abcd]" и "cat [bdac]".
3. Операторы && и || являются управляющими операторами. Если в командной строке стоит command1 && command2, то command2 выполняется в том, и только в том случае, если статус выхода из команды command1 равен нулю, что говорит об успешном ее завершении. Аналогично, если командная строка имеет вид command1 || command2, то команда command2 выполняется тогда, и только тогда, когда статус выхода из команды command1 отличен от нуля.
4. Оператор break завершает выполнение ближайшего включающего цикла или условного оператора, в котором он отображается.
5. Команда true всегда возвращает ноль в качестве выходного статуса для индикации успеха. Команда false всегда возвращает не-ноль в качестве выходного статуса для индикации неудачи. Во всех управляющих конструкциях в качестве логического значения используется код возврата из программы, указанной в качестве условия. Код возврата 0 – истина, любое другое значение – ложь. Программа true – всегда завершается с кодом 0, false – всегда завершается с кодом 1.
6. Введенная строка означает условие существования файла mans/i.\$s
7. Цикл While выполняется до тех пор, пока указанное в нем условие истинно. Когда указанное условие становится ложным - цикл завершается. Цикл Until выполняется до тех пор, пока указанное в нем условие ложно.