

# Отчёт по лабораторной работе №12

Дугина Виктория Игоревна

**Цель работы:** изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## Выполнение работы

**1. Написать командный файл, реализующий упрощённый механизм семафоров.** Командный файл должен в течение некоторого времени  $t_1$  дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени  $t_2 < t_1$ , также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой (`> /dev/tty#`, где # — номер терминала куда перенаправляется вывод), в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов

**2. Реализовать команду `man` с помощью командного файла.** Изучите содержимое каталога `/usr/share/man/man1`. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой `less` сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге `man1`.

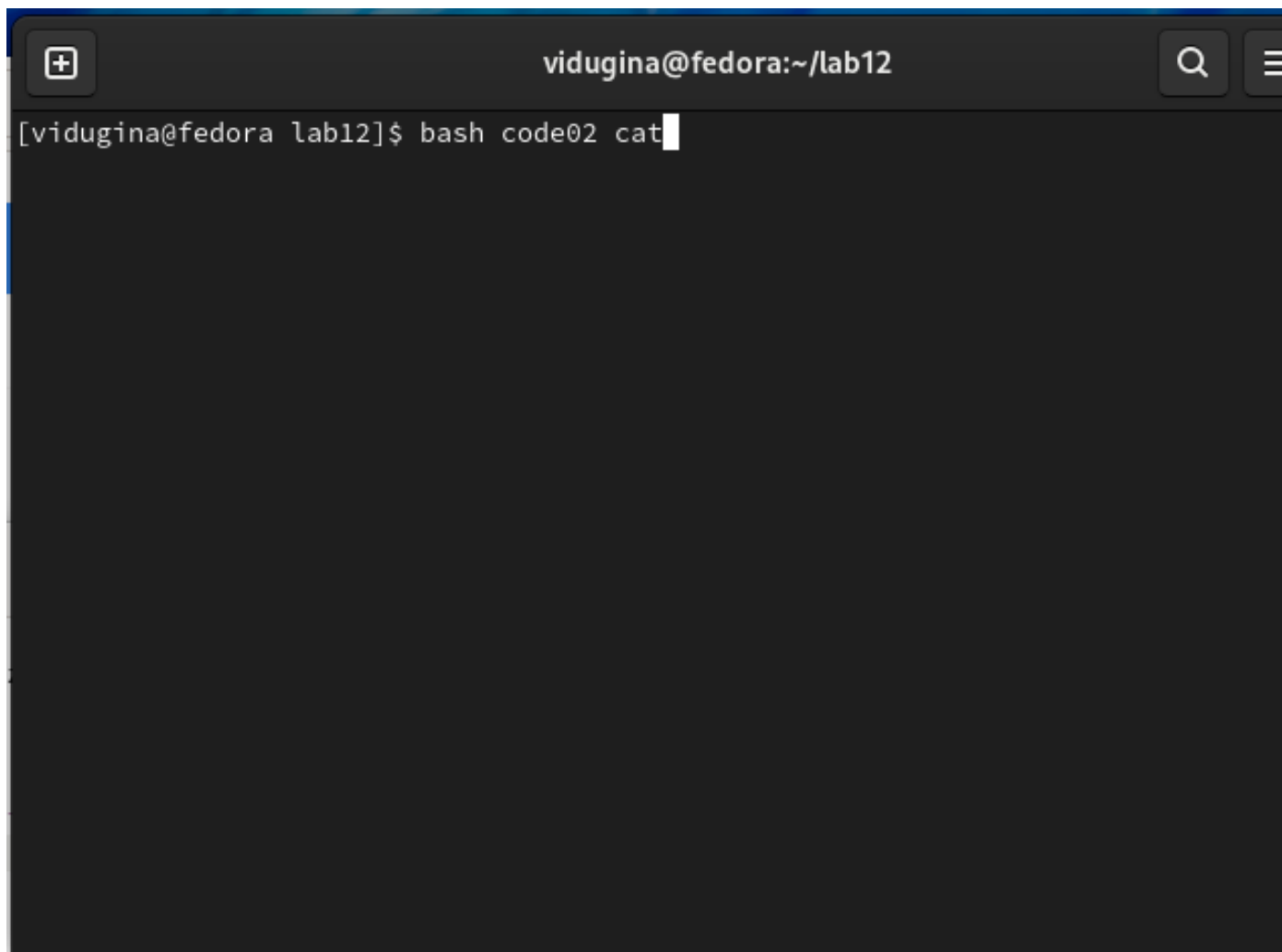
Задаём `M` полный путь к файлу из каталога `man1`, в зависимости от аргумента из командной строки (`$1`). Далее проверяем, существует ли данный файл (`test -f $M`). Если да - открываем с помощью команды `less`, нет - выдаём сообщение об этом.

```
1 #!/bin/bash
2 M=/usr/share/man/man1/$1.1.gz
3
4 if test -f $M
5 then less $M
6 else echo "Данной справки нет"
7 fi
```

sh    Ширина табуляции: 8    Стр 7, Стлб 3

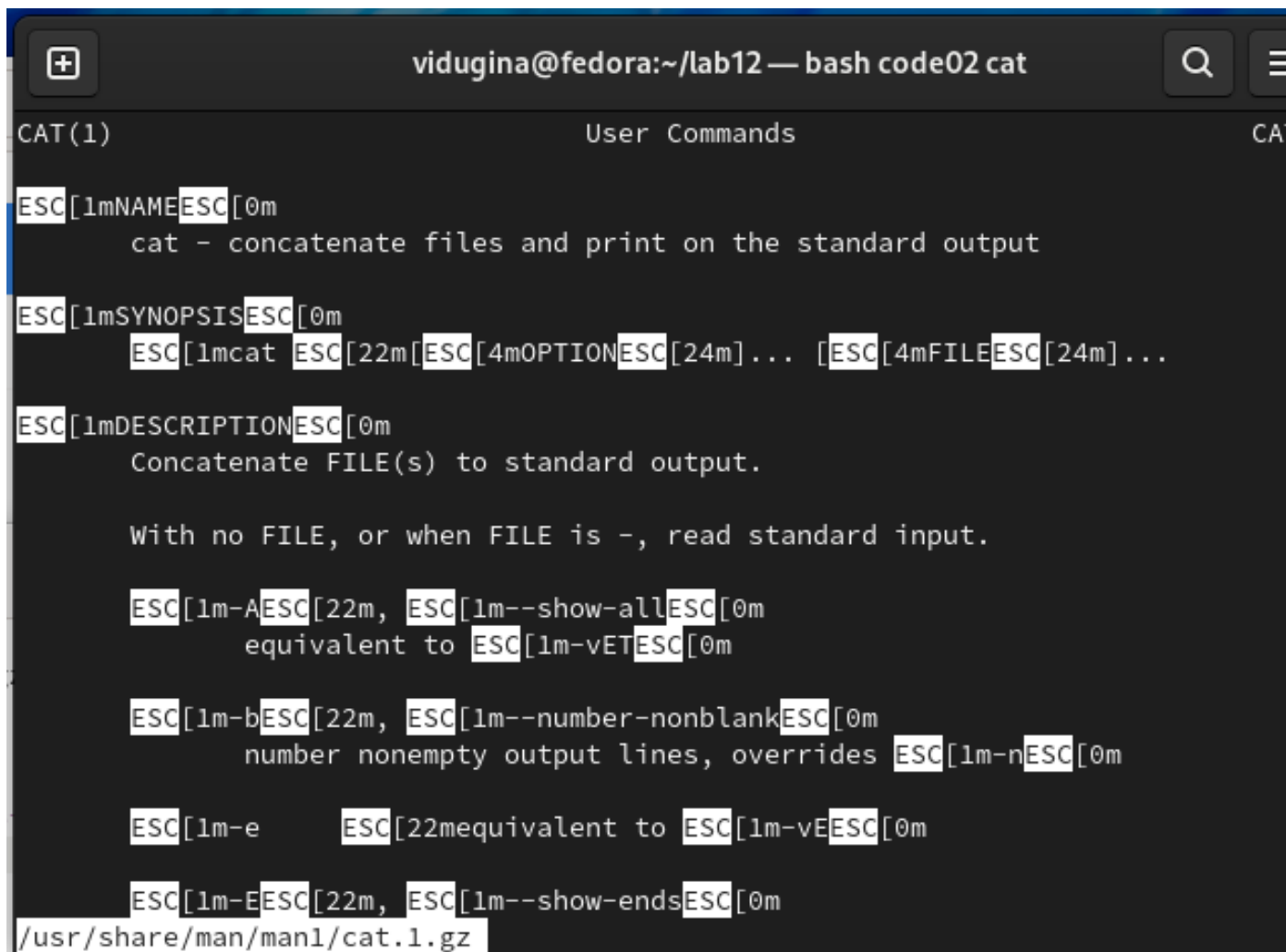
Рис 2.1 - Код

Работа программы.

A terminal window with a dark background. The title bar at the top shows a plus icon on the left, the text "vidugina@fedora:~/lab12" in the center, and a magnifying glass icon and a hamburger menu icon on the right. The terminal content shows a shell prompt "[vidugina@fedora lab12]" followed by the command "bash code02 cat" and a white cursor at the end of the line.

```
[vidugina@fedora lab12]$ bash code02 cat
```

*Puc 2.2*



A terminal window titled "vidugina@fedora:~/lab12 — bash code02 cat". The window shows the output of the command `man cat`. The output is formatted with bold headers for NAME, SYNOPSIS, DESCRIPTION, and various options. The terminal uses a dark background with light-colored text. The window has a title bar with a search icon and a menu icon.

```
CAT(1) User Commands CAT
ESC[1mNAMEESC[0m
    cat - concatenate files and print on the standard output

ESC[1mSYNOPSISESC[0m
    ESC[1mcat ESC[22m[ESC[4mOPTIONESC[24m]... [ESC[4mFILEESC[24m]...

ESC[1mDESCRIPTIONESC[0m
    Concatenate FILE(s) to standard output.

    With no FILE, or when FILE is -, read standard input.

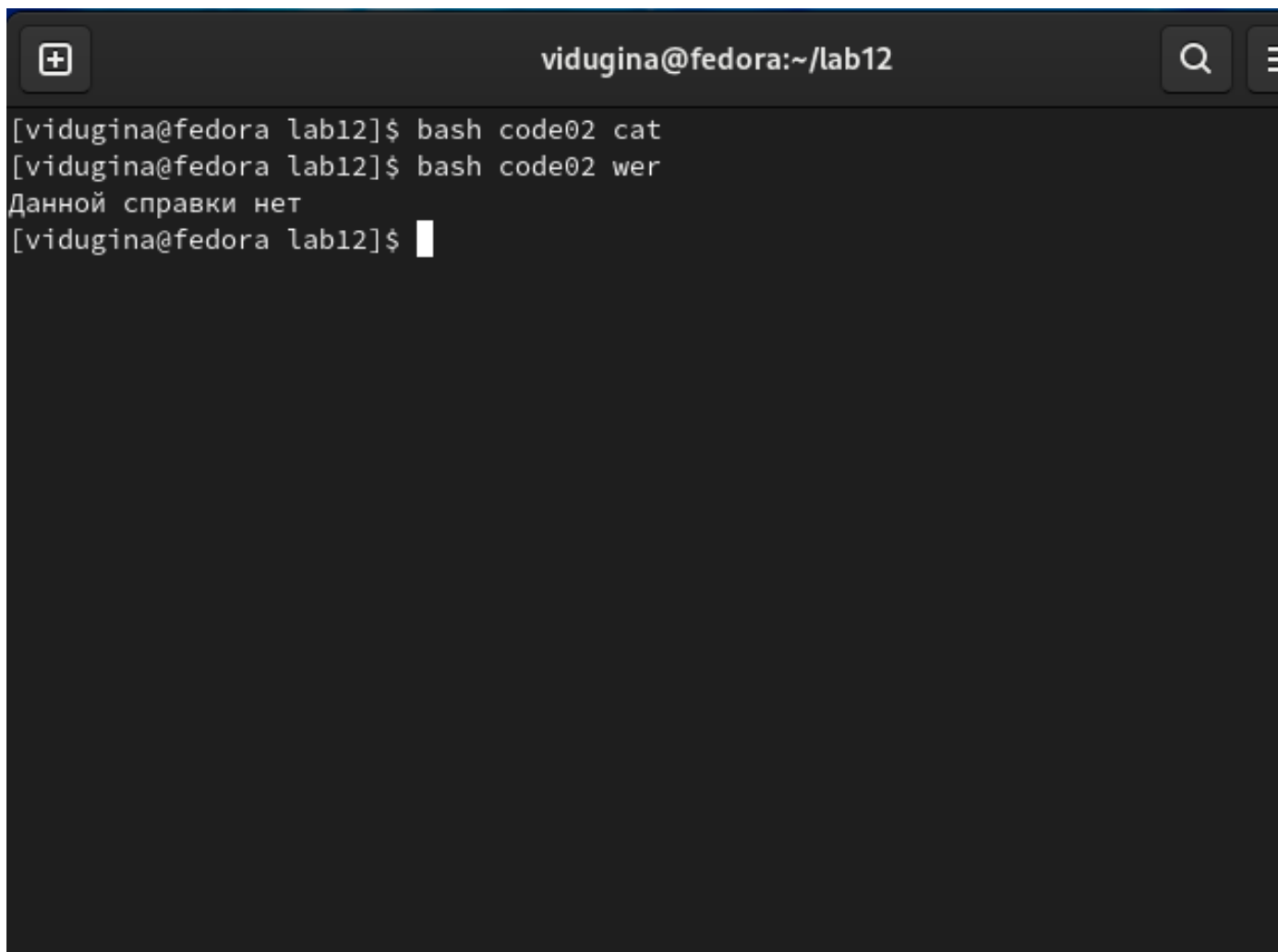
    ESC[1m-AESC[22m, ESC[1m--show-allESC[0m
        equivalent to ESC[1m-vETESC[0m

    ESC[1m-bESC[22m, ESC[1m--number-nonblankESC[0m
        number nonempty output lines, overrides ESC[1m-nESC[0m

    ESC[1m-e ESC[22mequivalent to ESC[1m-vEESC[0m

    ESC[1m-EEESC[22m, ESC[1m--show-endsESC[0m
/usr/share/man/man1/cat.1.gz
```

Рис 2.3 - Если описание команды есть

A terminal window with a dark background. The title bar shows 'vidugina@fedora:~/lab12'. The terminal content shows a user running 'bash code02 cat' and 'bash code02 wer', followed by the error message 'Данной справки нет' and a new prompt line.

```
[vidugina@fedora lab12]$ bash code02 cat
[vidugina@fedora lab12]$ bash code02 wer
Данной справки нет
[vidugina@fedora lab12]$
```

Рис 2.4 - Если описание команды нет

**3. Используя встроенную переменную \$RANDOM, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.**

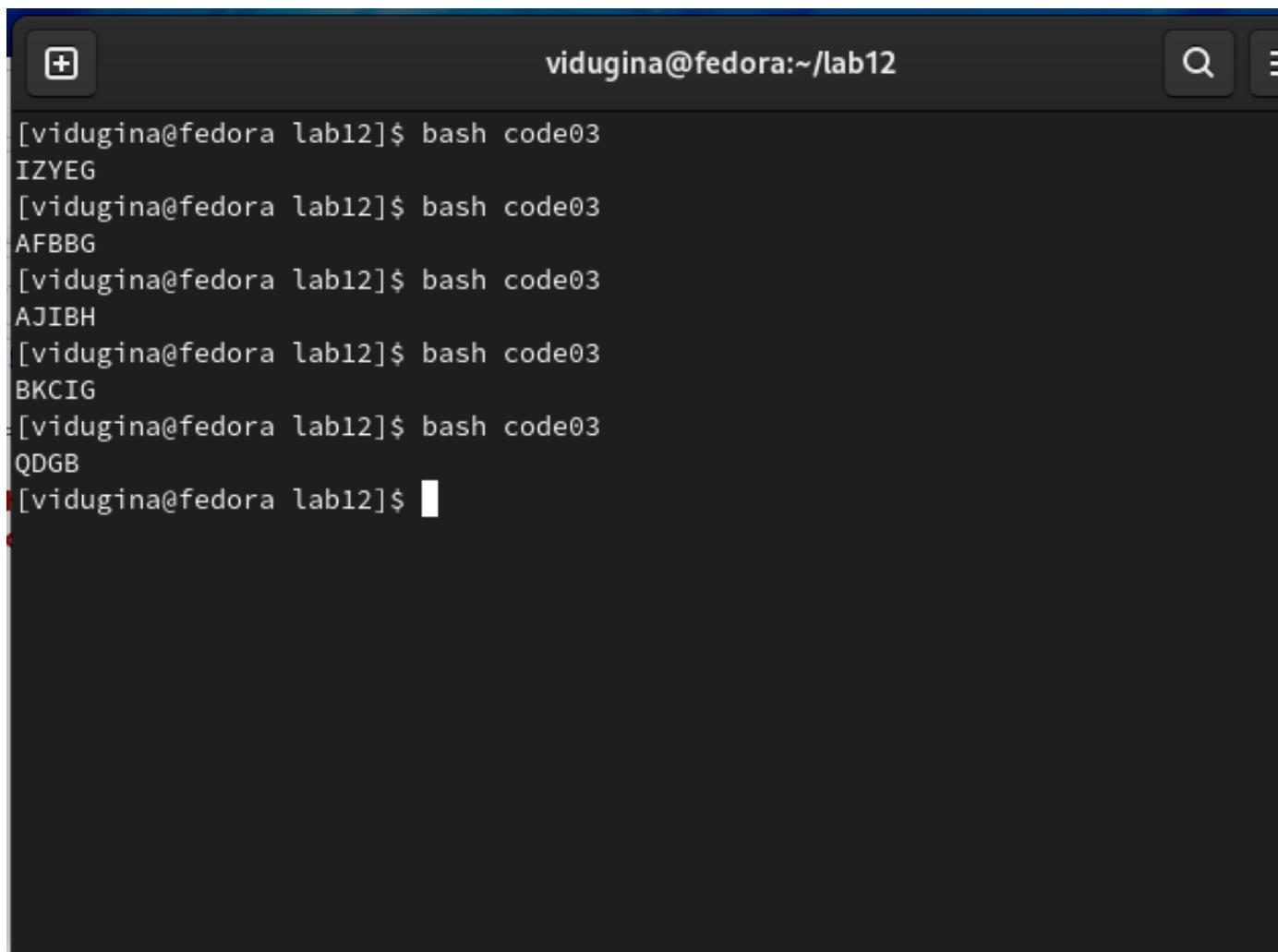
Задаём а условно рандомное число. Случайную последовательность будем генерировать используя перевод из числа в букву (`printf "\x$(printf %x $i)"` - переводит число \$i в букву и сразу выводит её). Задаём i число - последние две цифры числа a (`let i=a%100`). Далее анализируем число, так как перевести в букву можно числа только от 65 до 90. Если число меньше 65, то добавляем 10, пока оно не будет больше или равно 65. Если больше 90, то вычитаем 10. Получив подходящее число, переводим его и печатаем. Далее делим число a на 10, чтобы оно постепенно уменьшалось. И повторяем, пока число a не становится 0. Echo в конце, чтобы строки не сливались.

```
1 #!/bin/bash
2
3 a=$RANDOM
4
5 while ((a!=0))
6 do
7     let i=a%100
8     if (($i<65))
9     then
10         while (($i<65))
11         do let i=i+10
12         done
13         printf "\x$(printf %x $i)"
14
15     elif (($i>90))
16     then
17         let i=i-10
18         printf "\x$(printf %x $i)"
19
20
21     else
22         printf "\x$(printf %x $i)"
23     fi
24     let a=a/10
25 done
26 echo
```

sh    Ширина табуляции: 8    Стр 26, Стлб 5

Рис 3.1 - Код

Работа программы. Была запущена пять раз, чтобы был виден условный рандом.



```
vidugina@fedora:~/lab12
[vidugina@fedora lab12]$ bash code03
IZYEG
[vidugina@fedora lab12]$ bash code03
AFBBG
[vidugina@fedora lab12]$ bash code03
AJIBH
[vidugina@fedora lab12]$ bash code03
BKCIG
[vidugina@fedora lab12]$ bash code03
QDGB
[vidugina@fedora lab12]$
```

Рис 3.2

**Вывод:** я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## Контрольные вопросы

1. Каково предназначение команды `getopts`?
2. Какое отношение метасимволы имеют к генерации имён файлов?
3. Какие операторы управления действиями вы знаете?
4. Какие операторы используются для прерывания цикла?
5. Для чего нужны команды `false` и `true`?
6. Что означает строка `if test -f man$s/$i.$s`, встреченная в командном файле?
7. Объясните различия между конструкциями `while` и `until`.