## **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**

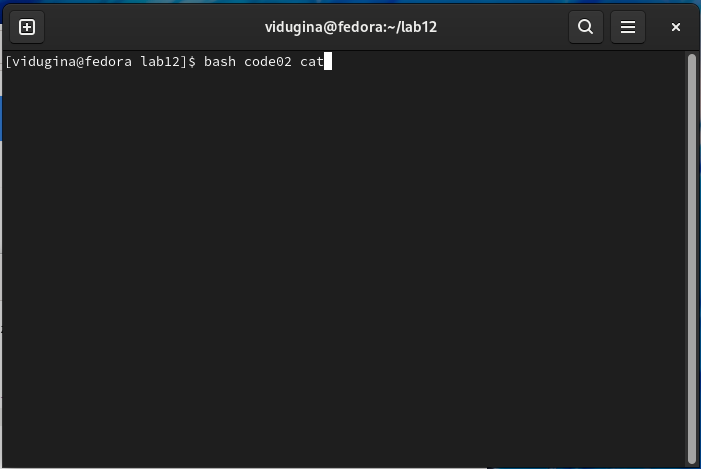
#### 1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени t2<>t1, также выдавая информацию о 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

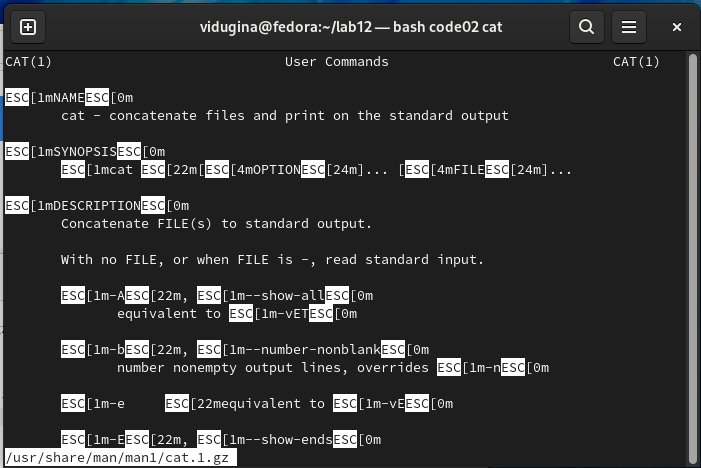
#### 2. Реализовать команду man с помощью командного файла. Изучите содержимое каталога /usr/share/man/man1. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой less сразу же просмотрев содержимое справки. 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.

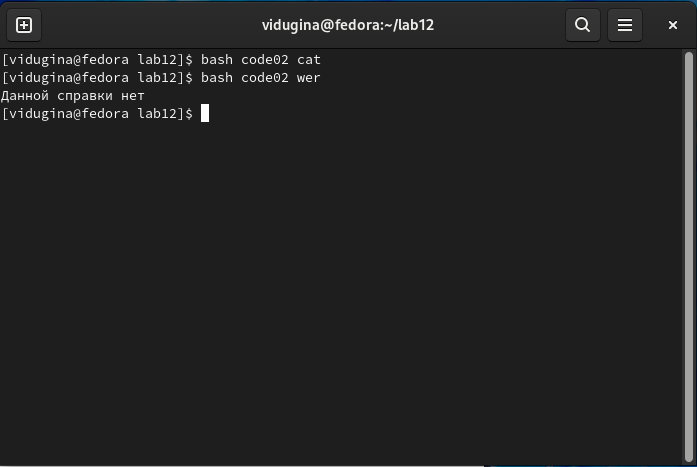
Задаём M полный путь к файлу из каталога man1, в зависимости от аргумента из командной строки ($1). Далее проверяем, существует ли данный файл (test -f $M). Если да - открываем с помощью команды less, нет - выдаём сообщение об этом.

 *Рис 2.1 - Код*

Работа программы.

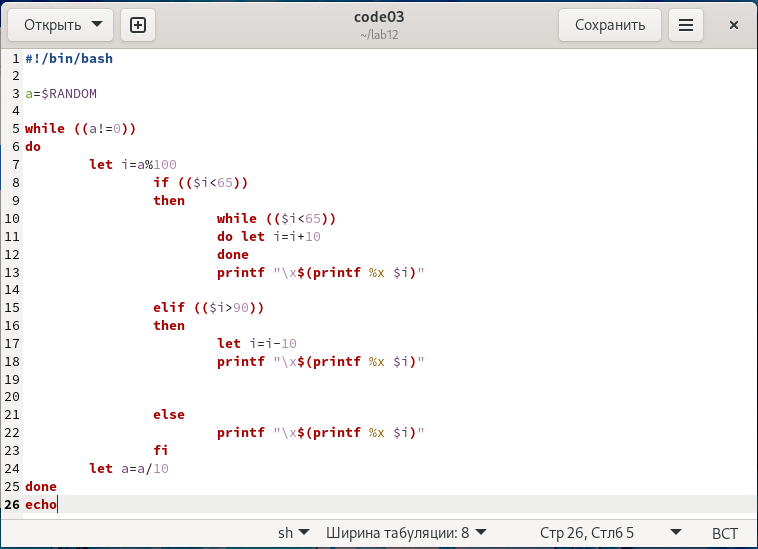
 *Рис 2.2*

 *Рис 2.3 - Если описание команды есть*

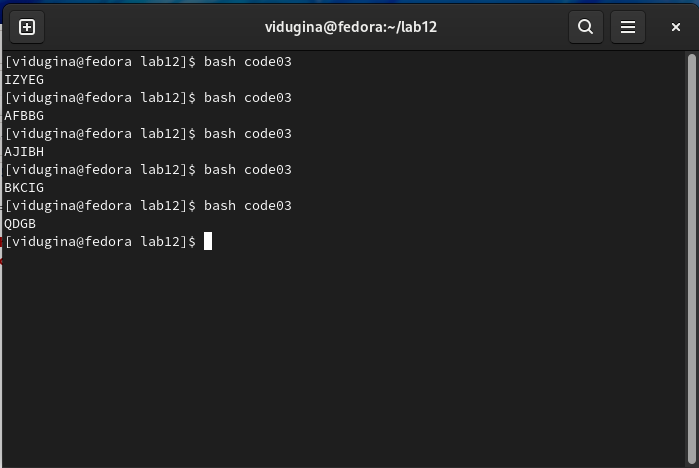
 *Рис 2.4 - Если описание команды нет*

#### 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 a 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 - последния две цифры числа а (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.

 *Рис 3.1 - Код*

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

 *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.