

Лаборная работа №12

НММ-6д-02-22

Крухмалев Артём Владиславович

Содержание

1	Цель работы	3
2	Задание	4
3	Выполнение лабораторной работы	5
4	Контрольные вопросы	8
5	Выводы	11

1 Цель работы

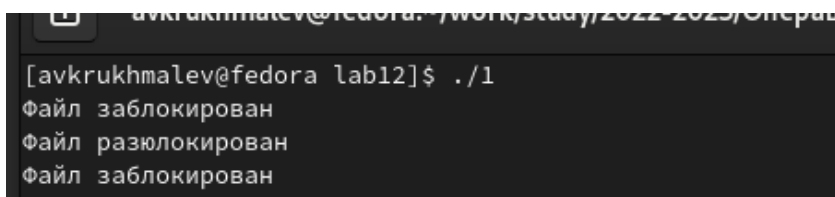
Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать командные файлы.

2 Задание

Написать программы

3 Выполнение лабораторной работы

1. Первая программа-блокирует файл



```
avkrukhtmalev@fedora: ~/work/study/2022-2023/Операци  
[avkrukhtmalev@fedora lab12]$ ./1  
Файл заблокирован  
Файл разблокирован  
Файл заблокирован
```

Рис. 3.1: 1

```
#!/bin/bash

LOCKFILE="./lock.file"
exec {fn}>$LOCKFILE

if test -f "$LOCKFILE"
then

    while
        [ 1 = 1 ]
    do
        if flock -n ${fn}
        then
            echo "Файл заблокирован"
            sleep 3
            echo "Файл разблокирован"
            flock -u ${fn}
        else
            echo "Файл заблокирован"
            sleep 3
        fi
    done
fi
```

Рис. 3.2: 1.2

2. Вторая программа

4 Контрольные вопросы

Каково предназначение команды `getopts`? Весьма необходимой при программировании является команда `getopts`, которая осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных. Синтаксис команды следующий: `getopts option-string variable [arg ...]` Флаги – это опции командной строки, обычно помеченные знаком минус; Например, `-F` является флагом для команды `ls -F`. Иногда эти флаги имеют аргументы, связанные с ними. Программы интерпретируют эти флаги, соответствующим образом изменяя свое поведение. Строка опций `option-string` — это список возможных букв и чисел соответствующего флага. Если ожидается, что некоторый флаг будет сопровождаться некоторым аргументом, то за этой буквой должно следовать двоеточие. Соответствующей переменной присваивается буква данной опции. Если команда `getopts` может распознать аргумент, она возвращает истину. Принято включать `getopts` в цикл `while` и анализировать введенные данные с помощью оператора `case`. Предположим, необходимо распознать командную строку следующего формата: `testprog -infile_in.txt -outfile_out.doc -L -t -r` Вот как выглядит использование оператора `getopts` в этом случае:

```
while getopts o:i:Ltr optletter do
case $optletter in
o) iflag=1; ival=$OPTARG;; i) iflag=1; ival=$OPTARG;; L)
Lflag=1;; t) tflag=1;; r) rflag=1;; *) echo Illegal option $optletter esac
done
```

 Функция `getopts` включает две специальные переменные среды – `OPTARG` и `OPTIND`. Если ожидается дополнительное значение, то `OPTARG` устанавливается в значение этого аргумента (будет равна `file_in.txt` для опции `i` и `file_out.doc` для опции `o`). `OPTIND` является числовым индексом на упомянутый аргумент. Функция `getopts`

также понимает переменные типа

массив, следовательно, можно использовать ее в функции не только для синтаксического анализа аргументов функций, но и для анализа введенных пользователем данных.

Какое отношение метасимволы имеют к генерации имён файлов? •

- — соответствует произвольной, в том числе и пустой строке;
- ? — соответствует любому одному символу;
- [c1-c1] — соответствует любому символу, лексикографически находящемуся между символами c1 и c2.
- echo * — выведет имена всех файлов текущего каталога, что представляет собой простейший аналог команды ls;
- ls *.c — выведет все файлы с последними двумя символами, равными .c.
- echo prog.? — выдаст все файлы, состоящие из пяти или шести символов, первыми пятью символами которых являются prog. .
- [a-z]* — соответствует произвольному имени файла в текущем каталоге, начинающемуся с любой строчной буквы латинского алфавита.

Какие операторы управления действиями вы знаете? Часто бывает необходимо обеспечить проведение каких-либо действий циклически и управление дальнейшими действиями в зависимости от результатов проверки некоторого условия. Для решения подобных задач язык программирования bash предоставляет Вам возможность использовать такие управляющие конструкции, как for, case, if и while. С точки зрения командного процессора эти управляющие конструкции являются обычными командами и могут использоваться как при создании командных файлов, так и при работе в интерактивном режиме. Команды, реализующие подобные конструкции, по сути дела являются операторами языка программирования bash. Поэтому при описании языка программирования bash термин оператор будет использоваться наравне с термином команда.

Какие операторы используются для прерывания цикла? Два несложных способа позволяют вам прерывать циклы в оболочке bash. Команда break завершает

выполнение цикла, а команда `continue` завершает данную итерацию блока операторов. Команда `break` полезна для завершения цикла `while` в ситуациях, когда условие перестает быть правильным. Пример бесконечного цикла `while`, с прерыванием в момент, когда файл перестает существовать: `while true do if [! -f $file] then break fi sleep 10 done`

Для чего нужны команды `false` и `true`? Команды ОС UNIX возвращают код завершения, значение которого может быть использовано для принятия решения о дальнейших действиях. Команда `test`, например, создана специально для использования в командных файлах. Единственная функция этой команды заключается в выработке кода завершения.

Что означает строка `if test -f mans/i.s, ?mans/i.s`

Объясните различия между конструкциями `while` и `until`. Если речь идет о 2-х параллельных действиях, то это `while`. когда мы показываем, что сначала делается 1-е действие. потом оно заканчивается при наступлении 2-го действия, применяем `until`.

5 Выводы

Научился писать командные файлы