РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 13

дисциплина: Операционные системы

Студент:

Гаглоев Олег Мелорович

Преподаватель:

Велиева Т.В.

Группа: НПИбд-01-20

МОСКВА 2021 г.

Цель работы:

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

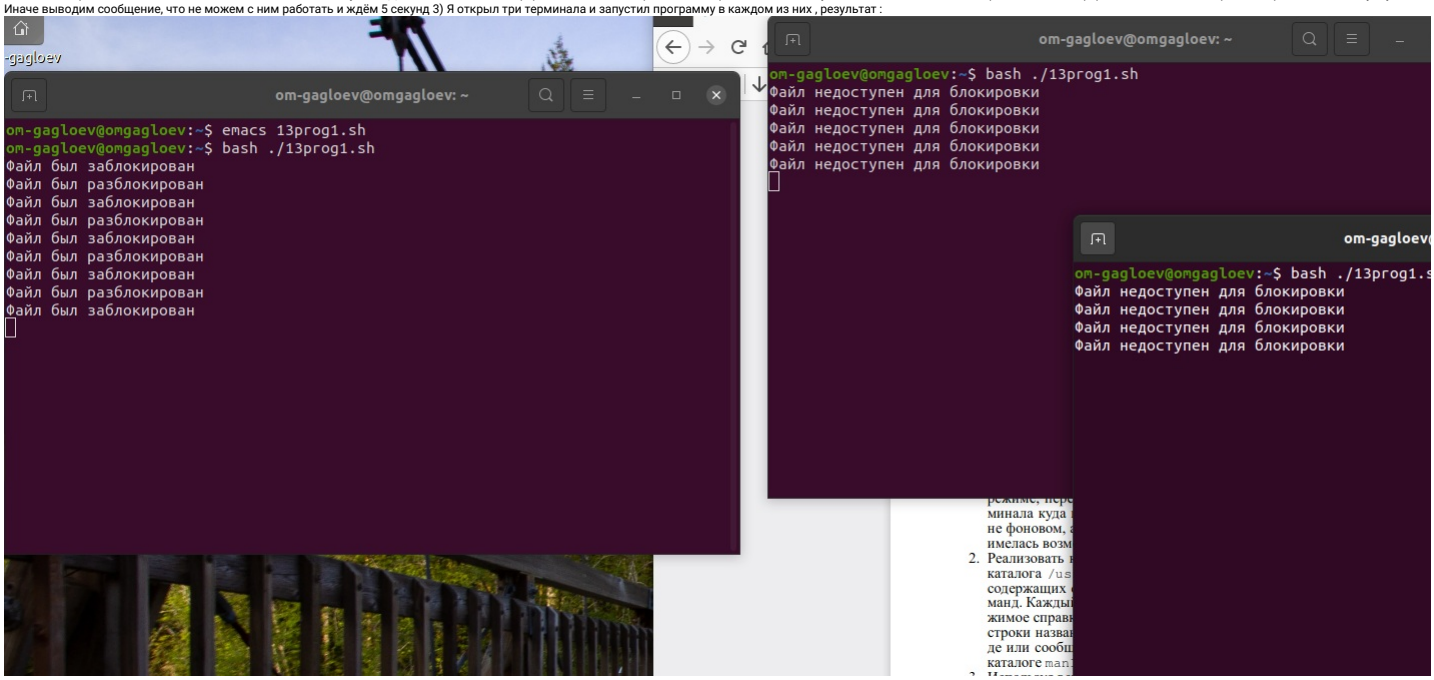
Выполнение работы:

Задание 1:

1)Создаю файл 13prog1.sh с помощью команды emacs 13prog1.sh Написал программу, выполняющую необходимые действия

```
#!/bin/bash
exec {fn}>./lock.file
while test -f ./lock.file
do
    if flock -n ${fn}
       then
       echo "Файл был заблокирован"
       sleep 3
      flock -u ${fn}
          echo "Файл был разблокирован"
          sleep 2
          else
              echo "Файл недоступен для блокировки"
              sleep 5
       fi
done
```

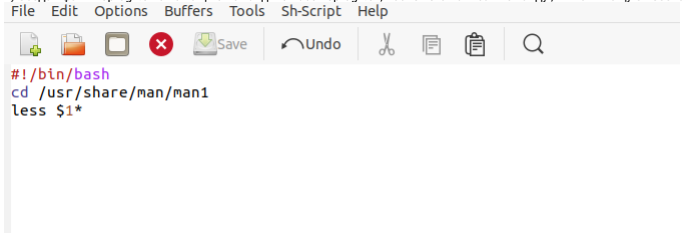
2)Опишу её в первой строке вызываем bash(интерпритатор). Далее мы присваиваем файлу номер с помощью команды exec{fn}>./lock/file. Создаём цикл, у которого условие следующее: test -f ./lock.file. Тест необходим для проверки того, существует ли файл и является он им вообще. Затем задаём условиен if flock -n ${fn} , проверяем, заблокирован ли файл, тем самым понимая, можно ли с производить с ним какие то действия, ждм 3 секунды . Далее мы выводим информацию о том, что файл заблокирован, ждем 2 секунд с помощью команды sleep 5, выводим информацию о том, что он разблокирован, ждем секунду. Иначе выводим сообщение, что не можем с ним работать и ждём 5 секунд 3) Я открыл три терминала и запустил программу в каждом из них , результат :



4) Как видим , при запуске трёх программ между ними возникают проблемы (конфликт)

Задание 2 :

1) Создал файл 13prog2.sh с помощью команды emacs 13prog2.sh, после чего написал команду , выполняющую необходимые условия



2) Опишу программу : В первой строке мы как обычно вызываем интерпритатор bash. Далее с помощью команды cd переходим в необходимый каталог (/usr/share/man/man1), в котором находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд, которые мы будем просматривать.С помощью команды less

```
om-gagloev@omgagloev:~$ emacs 13prog2.sh
om-gagloev@omgagloev:~$ bash ./13prog2.sh
om-gagloev@omgagloev:~$ bash ./13prog2.sh ls
om-gagloev@omgagloev:~$
```

просмотрим содержимое справки, используя указатель на файл. 3) Запущу свою программу использую команды bash ./13pro2.sh ls

```
.\" DO NOT MODIFY THIS FILE!  It was generated by help2man 1.47.3.
.TH LS "1" "September 2019" "GNU coreutils 8.30" "User Commands"
.SH NAME
ls \- list directory contents
.SH SYNOPSIS
.B ls
[\fI\,OPTION\/\fR]... [\fI\,FILE\/\fR]...
.SH DESCRIPTION
.\" Add any additional description here
.PP
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of \fB\-cftuvSUX\fR nor \fB\-\-sort\fR is sp
ecified.
.PP
Mandatory arguments to long options are mandatory for short options too.
.TP
\fB\-a\fR, \fB\-\-all\fR
do not ignore entries starting with .
.TP
\fB\-A\fR, \fB\-\-almost\-all\fR
do not list implied . and ..
.TP
\fB\-\-author\fR
ls.1.gz (file 1 of 9)
```

Получили справку по команде ls

Задание 3 :

1)С помощью команды emacs 13prog3.sh создал файл, в котором написал программу, выполняющую необходимые действия

```
#!/bin/bash
r=$((1+$RANDOM%10))
for((i=1;i<n;i++))
do
    echo -n $RANDOM | tr '[0-9]' '[a-z]'
done
echo $RANDOM | tr '[0-9]' '[a-z]'
```

2) Теперь опишу её : В первой строке как обычно вызов интерпритатора bash.Во второй строке я ввожу переменную r, куда помещаю рандомное число от 1 до 10. Далее я иду циклом от 1 до r. В каждой итерации я создаю рандомное число с помощью команды $RANDOM, после чего использую замену всех цифры в нем на буквы используя команду tr '[0-9]' '[a-z]', которую я нашёл в интернете и вывожу на экран с помощью команды echo и опции -n, В последней же строке я убираю функцию склеивания строк 3)Запускаю программу , используя команду bash ./13prog3.sh Видим, что количество букв действительно разное

```
om-gagloev@omgagloev:~$ bash ./13prog3.sh
jhcf
om-gagloev@omgagloev:~$ bash ./13prog3.sh
cafij
om-gagloev@omgagloev:~$ bash ./13prog3.sh
cciid
om-gagloev@omgagloev:~$ bash ./13prog3.sh
idb
om-gagloev@omgagloev:~$
```

Вывод:

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

Контрольные вопросы :

1)Нужно взять в кавычки «$1» 2)Необходимо написать переменные одну за другой. Например: A = "BC" . Либо с помощью оператора +=. Например: B += C [3] 3)Эта утилита выводит последовательность целых чисел с заданным шагом. Также можно реализовать с помощью утилиты jot. [4] 4)Результат - 3 5)B zsh можно настроить отдельные сочетания клавиш так, как вам нравится. Использование истории команд в zsh ничем особенным не отличается от bash. Но стоит обратить внимание на различия между этими двумя оболочками. Например, в zsh после for обязательно вставлять пробел, нумерация массивов в zsh начинается с 1, чего совершенно невозможно понять. Так, если вы используете shell для повседневной работы, исключающей написание скриптов, используйте zsh. Если вам часто приходится писать свои скрипты, только bash! Впрочем, можно комбинировать. 6)Синтаксис верен. 7)Преимущества: По сравнению с cmd у bash больше возможностей.

Его не нужно отдельно устанавливать, он встроен в операционную систему. Недостатки: В интернете меньше дополнительной информации про него, чем про языки программирования.