

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 13

**Программирование в командном процессоре ОС UNIX. Расширенное
программирование**

дисциплина: Операционные системы

Студент: Тазаева Анастасия Анатольевна

Группа: НПИбд-02-20

МОСКВА 2021г.

Содержание:

1. Цель работы
2. Задания
3. Ход работы
4. Контрольные вопросы(ответы)
5. Выводы

Цель работы:

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

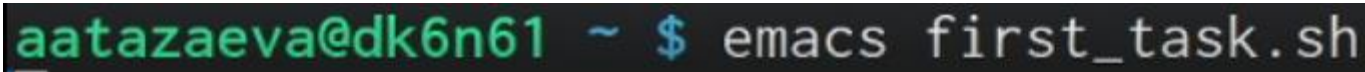
Задания:

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

- фоном, 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`.
 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.
-

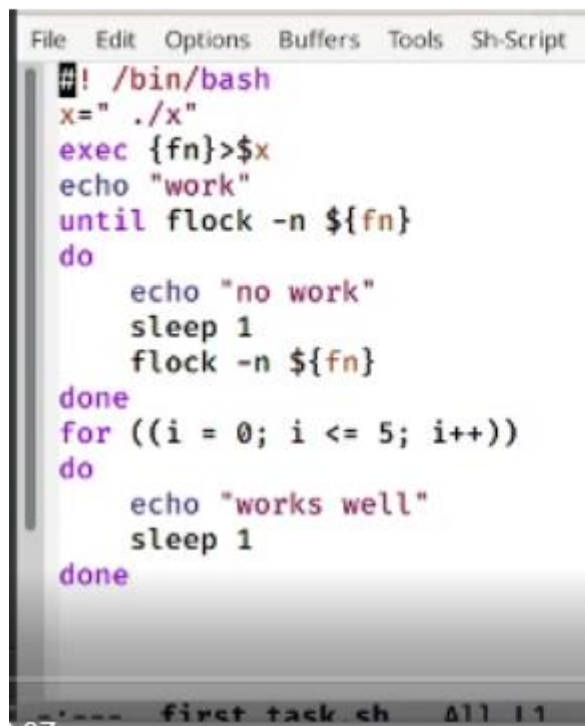
Ход работы:

1. С помощью команды `emacs` создала файл(рис.1), в котором и написала код, который реализует механизм семафоров(рис.2). Не забываем дать права на выполнение юзеру с помощью команды `chmod u+x (файл)`(рис.3)



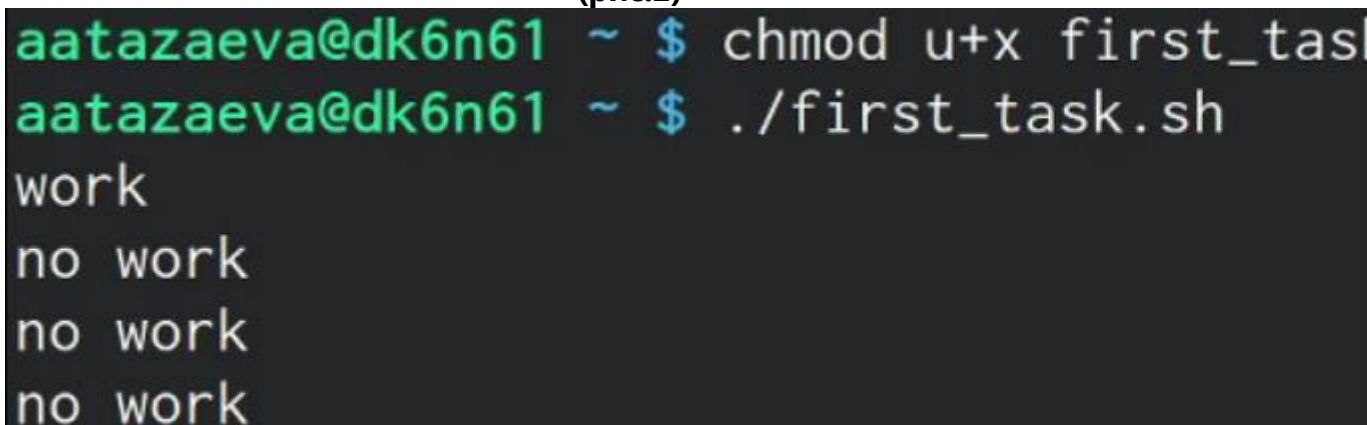
```
aatazaeva@dk6n61 ~ $ emacs first_task.sh
```

(рис.1)



```
File Edit Options Buffers Tools Sh-Script
#!/bin/bash
x=" ./x"
exec {fn}>$x
echo "work"
until flock -n ${fn}
do
    echo "no work"
    sleep 1
    flock -n ${fn}
done
for ((i = 0; i <= 5; i++))
do
    echo "works well"
    sleep 1
done
```

(рис.2)



```
aatazaeva@dk6n61 ~ $ chmod u+x first_task.sh
aatazaeva@dk6n61 ~ $ ./first_task.sh
work
no work
no work
no work
```

(рис.3)

Проверила работу(рис.3). Работает успешно.

2. Далее реализовала команду man с помощью командного файла. Сначала нам необходимо было изучить содержимое каталога /usr/share/man/man1(рис.4-5). Заметила, что там все файлы имеют расширение 1.bz2. Создаем файл для второго задания (emacs second_task.sh). Даем себе права на выполнение (chmod u+x srcond_task.sh). Ну и сам командный файл выглядит так(рис.6-7)

```
aatazaeva@dk6n61 ~ $ cd /usr/share/man/m
aatazaeva@dk6n61 /usr/share/man/man1 $ 1
```

(рис.4)

```
zdiff.1.bz2
zeisstopnm.1.bz2
zenity.1.bz2
zforce.1.bz2
zgrep.1.bz2
zip.1.bz2
zipcloak.1.bz2
zipcmp.1.bz2
zipdetails.1.bz2
zipdetails-2.84.0-perl-5.30.3.1.bz2
zipgrep.1.bz2
zipinfo.1.bz2
zipmerge.1.bz2
zipnote.1.bz2
zipnote.1.bz2
```

(рис.5)

```
emacs@dk6n61
File Edit Options Buffers Tools Sh-Script Help
#! /bin/bash
cd /usr/share/man/man1
less $1*

-:--- second_task.sh All L1 (Shell-script[bash]) Пт июн 4 16:55
Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Emacs Tutorial Learn basic keystroke commands (Учебник Emacs)
Emacs Guided Tour Overview of Emacs features at gnu.org
View Emacs Manual View the Emacs manual using Info
Absence of Warranty GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions Conditions for redistributing and changing Emacs
Ordering Manuals Purchasing printed copies of manuals
To quit a partially entered command, type Control-g.

This is GNU Emacs 27.1 (build 1, x86_64-pc-linux-gnu, GTK+ Version 3.24.22, cairo version 1.16.0)
of 2021-01-11
Copyright (C) 2020 Free Software Foundation, Inc.

Dismiss this startup screen ☐ Never show it again.

U:%%- *GNU Emacs* All L3 (Fundamental) Пт июн 4 16:55 1.63
Indentation setup for shell type bash

zvbi-ntsc-cc.1.bz2
aatazaeva@dk6n61 /usr/share/man/man1 $ co
aatazaeva@dk6n61 ~ $ emacs second_task.sh
```

(рис.6)

```
aatazaeva@dk6n61 ~ $ chmod u+x second_ta
```

(рис.7)

Проверила работу командного файла.(рис.8) Код справился с заданием и вывел нам информацию и zstd.(рис.9)


```
zstd.1.bz2
zstdcat.1.bz2
zstdgrep.1.bz2
zstdless.1.bz2
zvbi-chains.1.bz2
zvbid.1.bz2
zvbi-ntsc-cc.1.bz2
aatazaeva@dk6n61 /usr/share/man/man1 $ c
aatazaeva@dk6n61 ~ $ emacs second_task.s
aatazaeva@dk6n61 ~ $ chmod u+x second_ta
aatazaeva@dk6n61 ~ $ ./second_task.sh zs
```

(рис.8)

```
ZSTD(1)      User Commands

NAME
    zstd - zstd, zstdmt, unzstd, zs
    Compress or decompress .zst files

SYNOPSIS
    zstd [OPTIONS] [-|INPUT-FILE] [-|
    PUT-FILE]

    zstdmt is equivalent to zstd -T0

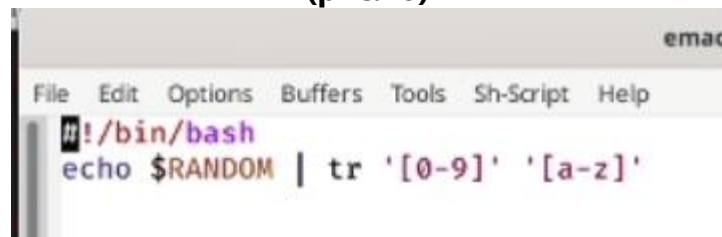
    unzstd is equivalent to zstd -d
```

(рис.9)

3. Создаем командный файл, который будет выводить на экран случайные буквы. Для этого создаем файл и даем ему права(рис.10):
emacs third_task.sh
chmod u+x third_task.sh

```
aatazaeva@dk6n61 ~ $ emacs third_task.sh  
aatazaeva@dk6n61 ~ $ chmod u+x third_tas
```

(рис.10)



(рис.11)

```
aatazaeva@dk6n61 ~ $ ./third_task.sh
bcjfg
aatazaeva@dk6n61 ~ $ ./third_task.sh
bhdfe
aatazaeva@dk6n61 ~ $ ./third_task.sh
bihdd
aatazaeva@dk6n61 ~ $ ./third_task.sh
jhbh
aatazaeva@dk6n61 ~ $ ./third_task.sh
cjejj
aatazaeva@dk6n61 ~ $ ./third_task.sh
gadc
aatazaeva@dk6n61 ~ $ ./third_task.sh
hafh
aatazaeva@dk6n61 ~ $ ./third_task.sh
baidd
aatazaeva@dk6n61 ~ $ ./third_task.sh
bgagb
```

(рис.12)

Далее сам код и его проверка(рис.11 и 12).

Контрольные вопросы:

1. Найдите синтаксическую ошибку в следующей строке:

```
while [$1 != "exit"]
```

```
$1.
```

Так же между скобками должны быть пробелы. В противном случае скобки и рядом стоящие символы будут восприниматься как одно целое

2. Как объединить (конкатенация) несколько строк в одну?

```
cat file.txt | xargs | sed -e 's/. /\n/g'
```

3. Найдите информацию об утилите seq. Какими иными способами можно реализовать её функционал при программировании на bash?

seq - выдает последовательность чисел.

Реализовать ее функционал можно командой

```
for n in {1..5}
```

```
do <КОМАНДА>
```

```
done
```

4. Какой результат даст вычисление выражения $\$((10/3))$?

3

5. Укажите кратко основные отличия командной оболочки zsh от bash.

Zsh очень сильно упрощает работу. Но существуют различия. Например, в zsh после for обязательно вставлять пробел, нумерация массивов в zsh начинается с 1 (что не особо удобно на самом деле).

Если вы собираетесь писать скрипт, который легко будет запускать множество разработчиков, то я рекомендую Bash. Если скрипты вам не нужны - Zsh (более простая работа с файлами, например)

6. Проверьте, верен ли синтаксис данной конструкции for ((a=1; a <= LIMIT; a++))

Верен

7. Сравните язык bash с какими-либо языками программирования. Какие преимущества у bash по сравнению с ними? Какие недостатки?

Bash позволяет очень легко работать с файловой системой без лишних конструкций (в отличие от обычного языка программирования). Но относительно обычных языков программирования bash очень сжат. Тот же Си имеет гораздо более широкие возможности для разработчика.

ВЫВОД:

В ходе лабораторной работы я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов