

Лабораторная работа № 12.

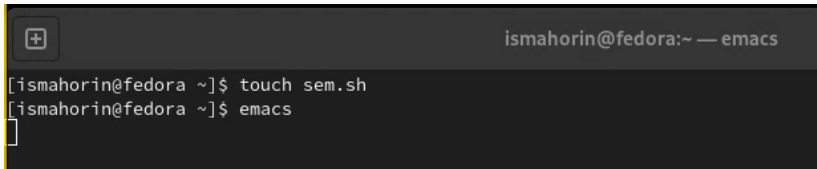
Махорин Иван Сергеевич

2022, 26 мая

RUDN, Москва, Россия

Программирование в командном
процессоре ОС UNIX. Расширенное
программирование.

Создание первого файла для скрипта



```
ismahorin@fedora:~ — emacs
[ismahorin@fedora ~]$ touch sem.sh
[ismahorin@fedora ~]$ emacs

```

Figure 1: Создание нового файла для скрипта

Написать командный файл, который реализует 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.

Написание первого скрипта



```
File Edit Options Buffers Tools Sh-Script Help
[Icons: New, Open, Save, Undo, Cut, Copy, Paste, Find]

#!/bin/bash
t1=$1
t2=$2
s1=$(date +%s)
s2=$(date +%s)
((t=$s2-$s1))
while ((t < t1))
do
    echo "Ожидание"
    sleep 1
    s2=$(date +%s)
    ((t=$s2-$s1))
done
s1=$(date +%s)
s2=$(date +%s)
((t=$s2-$s1))
while ((t < t2))
do
    echo "Выполнение"
    sleep 1
    s2=$(date +%s)
    ((t=$s2-$s1))
done
```

Figure 2: Написание первого скрипта



```
ismahorin@fedora:~ — /bin/bash ./sem.sh 4 7
[ismahorin@fedora ~]$ touch sem.sh
[ismahorin@fedora ~]$ emacs
[ismahorin@fedora ~]$ chmod +x sem.sh
[ismahorin@fedora ~]$ ./sem.sh 4 7
Ожидание
Ожидание
Ожидание
Ожидание
```

Figure 3: Право на выполнение, запуск файла и проверка

Доработаем программу так, чтобы имелась возможность взаимодействия трёх и более процессов.

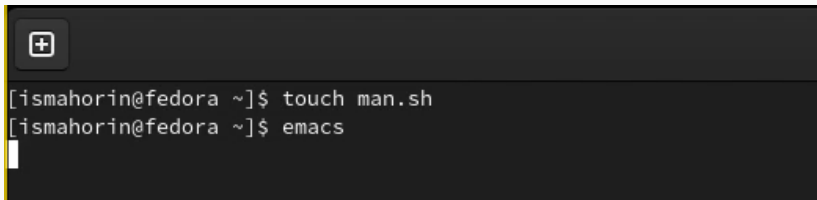


```
File Edit Options Buffers Tools Sh-Script Help
#1/bin/bash
function ogidanie
{
    s1=$(date +%s)
    s2=$(date +%s)
    ((t=s2-s1))
    while ((t < t1))
    do
        echo "Ожидание"
        sleep 1
        s2=$(date +%s)
        ((t=s2-s1))
    done
}
function vypolnenie
{
    s1=$(date +%s)
    s2=$(date +%s)
    ((t=s2-s1))
    while ((t < t2))
    do
        echo "Выполнение"
        sleep 1
        s2=$(date +%s)
        ((t=s2-s1))
    done
}
t1=$1
t2=$2
command=$3
while true
do
    if [ "$command" == "Выход" ]
    then
        echo "Выход"
        exit 0
    fi
    if [ "$command" == "Ожидание" ]
    then
        ogidanie
    fi
    if [ "$command" == "Выполнение" ]
    then
        vypolnenie
    fi
    echo "Следующее действие: "
    read command
done
```

Figure 4: Доработка первого скрипта


```
[ismahorin@fedora ~]$ emacs
[ismahorin@fedora ~]$ chmod +x sem.sh
[ismahorin@fedora ~]$ ./sem.sh 2 3 Ожидание > /dev/pts/1 &
[1] 132301
bash: /dev/pts/1: Отказано в доступе
[1]+  Выход 1                  ./sem.sh 2 3 Ожидание > /dev/pts/1
[ismahorin@fedora ~]$ ./sem.sh 2 3 Ожидание > /dev/pts/2 &
[1] 132313
bash: /dev/pts/2: Отказано в доступе
[1]+  Выход 1                  ./sem.sh 2 3 Ожидание > /dev/pts/2
[ismahorin@fedora ~]$ bash: /dev/pts/2: Отказано в доступе
bash: bash:: command not found...
Similar command is: 'bash'
[ismahorin@fedora ~]$
```

Figure 5: Сохранение и проверка

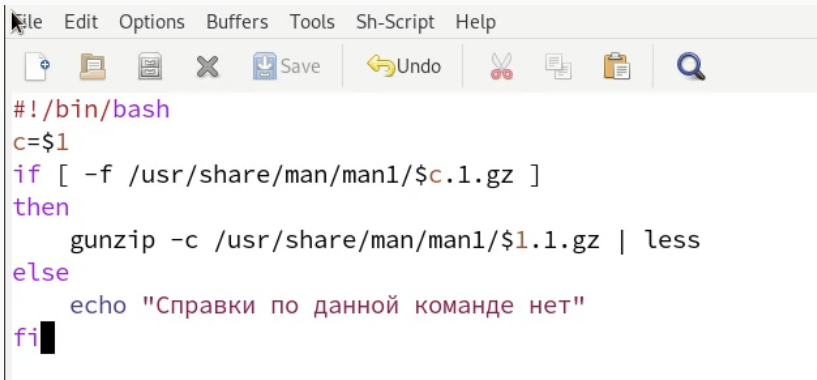


```
[ismahorin@fedora ~]$ touch man.sh
[ismahorin@fedora ~]$ emacs
```

Figure 6: Создание файла для второго задания

Приступить к реализации команды `man` с помощью командного файла. Изучить 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`.

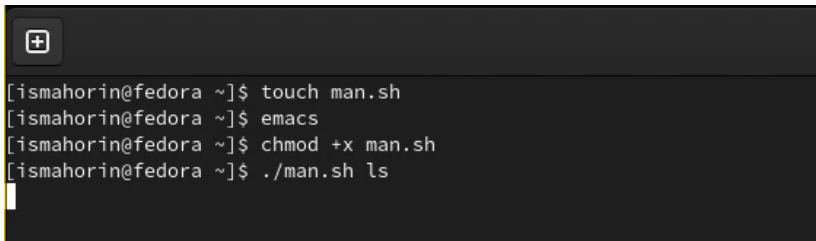
Написание второго скрипта

A screenshot of a text editor window with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a toolbar with icons for file operations. The editor contains a shell script with the following content:

```
#!/bin/bash
c=$1
if [ -f /usr/share/man/man1/$c.1.gz ]
then
    gunzip -c /usr/share/man/man1/$1.1.gz | less
else
    echo "Справки по данной команде нет"
fi
```


The script uses color-coded syntax: purple for keywords, brown for variables, and red for file paths and commands. The cursor is at the end of the last line.

Figure 7: Написание командного файла для второго задания

A terminal window with a dark background and a light-colored border. The window title bar shows a plus icon in a square. The terminal text shows a user named ismahorin on a fedora system in the home directory (~). The user runs four commands: 'touch man.sh', 'emacs', 'chmod +x man.sh', and './man.sh ls'. The output of the last command is a blank line.

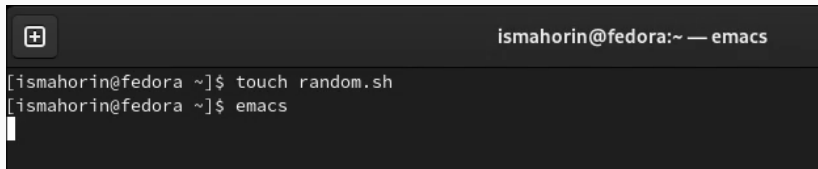
```
[ismahorin@fedora ~]$ touch man.sh
[ismahorin@fedora ~]$ emacs
[ismahorin@fedora ~]$ chmod +x man.sh
[ismahorin@fedora ~]$ ./man.sh ls
```

Figure 8: Право на выполнение, запуск файла для команды ls



```
\" DO NOT MODIFY THIS FILE!  It was generated by help2man 1.47.3.
TH LS "1" "March 2022" "GNU coreutils 8.32" "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 specified.
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
with \fB\--l\fR, print the author of each file
TP
```

Figure 9: Информация о команде ls

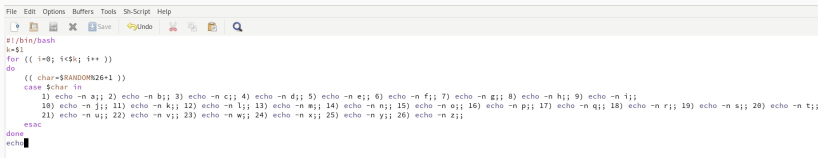


```
ismahorin@fedora:~ — emacs
[ismahorin@fedora ~]$ touch random.sh
[ismahorin@fedora ~]$ emacs
```

Figure 10: Создание третьего файла

Написать командный файл, генерирующий случайную последовательность букв латинского алфавита. Учесть, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

Написание третьего скрипта



```
File Edit Options Buffers Tools Sh-Script Help
[Icons] Save Undo [Icons] Search
#!/bin/bash
k=$1
for (( i=0; i<$k; i++ ))
do
  (( char=$RANDOM%26+1 ))
  case $char in
    1) echo -n a;; 2) echo -n b;; 3) echo -n c;; 4) echo -n d;; 5) echo -n e;; 6) echo -n f;; 7) echo -n g;; 8) echo -n h;; 9) echo -n i;;
    10) echo -n j;; 11) echo -n k;; 12) echo -n l;; 13) echo -n m;; 14) echo -n n;; 15) echo -n o;; 16) echo -n p;; 17) echo -n q;; 18) echo -n r;; 19) echo -n s;; 20) echo -n t;;
    21) echo -n u;; 22) echo -n v;; 23) echo -n w;; 24) echo -n x;; 25) echo -n y;; 26) echo -n z;;
  esac
done
echo
```

Figure 11: Написание третьего скрипта

A terminal window with a dark background. The title bar shows a window icon on the left and the text 'ismahorin@fedora:~' on the right. The terminal contains a series of commands and their outputs. The commands are: 'touch random.sh', 'emacs', 'chmod +x random.sh', './random.sh 7', './random.sh 20', and './random.sh 13'. The outputs are: 'esshcrv', 'mygngidavcewdqtsqptj', and 'pkgfixgywximq'. The prompt '[ismahorin@fedora ~]\$' is visible at the end of each line.

```
[ismahorin@fedora ~]$ touch random.sh
[ismahorin@fedora ~]$ emacs
[ismahorin@fedora ~]$ chmod +x random.sh
[ismahorin@fedora ~]$ ./random.sh 7
esshcrv
[ismahorin@fedora ~]$ ./random.sh 20
mygngidavcewdqtsqptj
[ismahorin@fedora ~]$ ./random.sh 13
pkgfixgywximq
[ismahorin@fedora ~]$
```

Figure 12: Право на выполнение, запуск файла

Выводы

В ходе выполнения лабораторной работы мы изучили основы программирования в оболочке ОС UNIX и научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Спасибо за внимание!