

Laboratory №13

Krupennikova V.

MAY-2021

RUDN University, Moscow, Russian Federation

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

1. Ознакомиться с теоретическим материалом.
2. Изучить основы программирования в оболочке ОС UNIX/Linux.
3. Выполнить упражнения.
4. Ответить на контрольные вопросы.

Выполнение:

1. Написала 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). Для данной задачи я создала файл: sem.sh и написала соответствующий скрипт.

```
emacs@dk6n6
File Edit Options Buffers Tools Sh-Script Help
#!/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)
```

Далее я проверила работу написанного скрипта (./sem.sh 4 7), добавив право на исполнение файла (chmod +x .sh). Скрипт работает корректно.

```
vakrupennikova@dk6n66 ~ $ chmod +x sem.sh
vakrupennikova@dk6n66 ~ $ ./sem.sh 4 7
Ожидание
Ожидание
Ожидание
Ожидание
Выполнение
Выполнение
Выполнение
Выполнение
Выполнение
Выполнение
Выполнение
vakrupennikova@dk6n66 ~ $
```

Figure 2: Проверка скрипта

После этого я изменила скрипт так, чтобы его можно было выполнять в нескольких терминалах и проверила его работу (например, команда «./sem.sh 2 3 Ожидание > /dev/pts/1 &»)

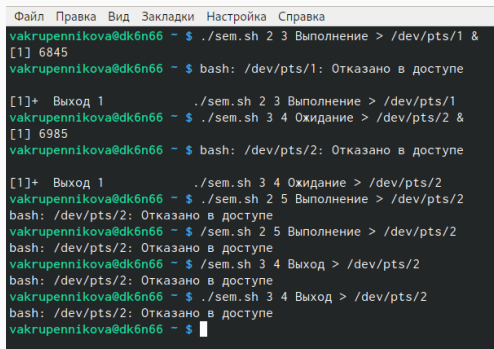
File Edit Options Buffers Tools Sh-Script Help

```
#!/bin/bash
function ogidanie
{
s1=$(date +%S")
s2=$(date +%S")
((t=s2-s1))
while ((t<t1))
do
    echo "Ожидание"
    sleep 1
    s2=$(date +%S")
    ((t=s2-s1))
done
}
function vipolnenie
{
s1=$(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 vipolnenie
    fi
    echo "Следующее действие: "
    read command
done
```

Figure 4: Изменение скрипта

Но ни одна команда не работала, так как мне “Отказано в доступе”. При этом скрипт работает корректно (команда «./sem.sh »).



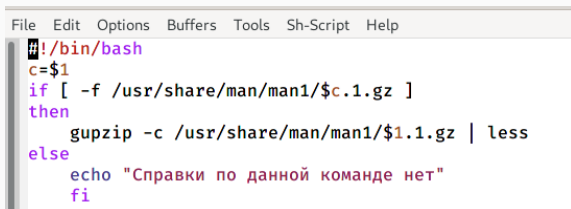
```
Файл Правка Вид Закладки Настройка Справка
vakrupennikova@dk6n66 ~ $ ./sem.sh 2 3 Выполнение > /dev/pts/1 &
[1] 6845
vakrupennikova@dk6n66 ~ $ bash: /dev/pts/1: Отказано в доступе

[1]+ Выход 1          ./sem.sh 2 3 Выполнение > /dev/pts/1
vakrupennikova@dk6n66 ~ $ ./sem.sh 3 4 Ожидание > /dev/pts/2 &
[1] 6985
vakrupennikova@dk6n66 ~ $ bash: /dev/pts/2: Отказано в доступе

[1]+ Выход 1          ./sem.sh 3 4 Ожидание > /dev/pts/2
vakrupennikova@dk6n66 ~ $ ./sem.sh 2 5 Выполнение > /dev/pts/2
bash: /dev/pts/2: Отказано в доступе
vakrupennikova@dk6n66 ~ $ ./sem.sh 2 5 Выполнение > /dev/pts/2
bash: /dev/pts/2: Отказано в доступе
vakrupennikova@dk6n66 ~ $ ./sem.sh 3 4 Выход > /dev/pts/2
bash: /dev/pts/2: Отказано в доступе
vakrupennikova@dk6n66 ~ $ ./sem.sh 3 4 Выход > /dev/pts/2
bash: /dev/pts/2: Отказано в доступе
vakrupennikova@dk6n66 ~ $
```

Figure 5: Проверка скрипта

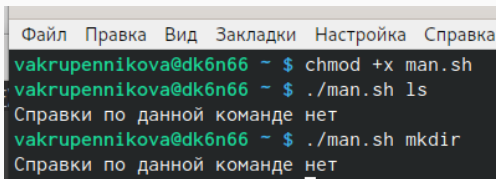
Для данной задачи я создала файл : man.sh и написала соответствующий скрипт.



```
File Edit Options Buffers Tools Sh-Script Help
#!/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
```

Figure 7: Написание скрипта

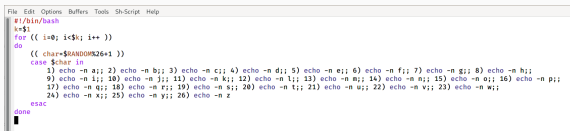
Далее я проверила работу написанного скрипта (команды «./man.sh ls», «./man.sh mkdir»), предварительно добавив право на исполнение файла (команда «chmod +x man.sh»). Скрипт сработал и вывел, что по данным командам справок нет. Скрипт работает корректно.

A screenshot of a terminal window with a menu bar at the top containing 'Файл', 'Правка', 'Вид', 'Закладки', 'Настройка', and 'Справка'. The terminal shows a user named 'vakrupennikova@dk6n66' in a shell. The first command is 'chmod +x man.sh'. The second command is './man.sh ls', which outputs 'Справки по данной команде нет'. The third command is './man.sh mkdir', which also outputs 'Справки по данной команде нет'.

```
Файл Правка Вид Закладки Настройка Справка
vakrupennikova@dk6n66 ~ $ chmod +x man.sh
vakrupennikova@dk6n66 ~ $ ./man.sh ls
Справки по данной команде нет
vakrupennikova@dk6n66 ~ $ ./man.sh mkdir
Справки по данной команде нет
```

Figure 8: Проверка скрипта

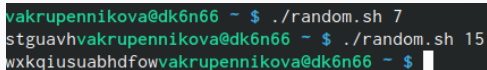
3. Используя встроенную переменную \$RANDOM, написала командный файл, генерирующий случайную последовательность букв латинского алфавита. Для данной задачи я создала файл: random.sh и написала соответствующий скрипт.

A screenshot of a terminal window with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help). The terminal shows a shell script for generating random letters. The script starts with a shebang, sets a variable k, and enters a loop that runs 26 times. In each iteration, it generates a random character from the alphabet and echoes it. The script ends with a 'done' statement.

```
#!/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
```

Figure 9: Написание скрипта

Далее я проверила работу написанного скрипта (команды «./random.sh 7» и «./random.sh 15»), предварительно добавив право на исполнение файла (команда «chmod +x random.sh»). Скрипт работает корректно.

A terminal window with a black background and green text. It shows two commands being executed: './random.sh 7' and './random.sh 15'. The first command produces a long alphanumeric string 'stguavhv' as output. The second command produces a long alphanumeric string 'wxkqiusuabhd' as output. The prompt 'vkrupennikova@dk6n66 ~ \$' is visible at the end of each line.

```
vkrupennikova@dk6n66 ~ $ ./random.sh 7
stguavhv
vkrupennikova@dk6n66 ~ $ ./random.sh 15
wxkqiusuabhd
```

Figure 10: Проверка скрипта

В ходе выполнения данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX/Linux и научилась писать небольшие командные файлы.

The end.