

# презентация по лабораторной работе 12

Программирование в командном процессоре ОС UNIX. Расширенное программирование

---

Боровиков Д.А.

Российский университет дружбы народов, Москва, Россия

## Информация

---

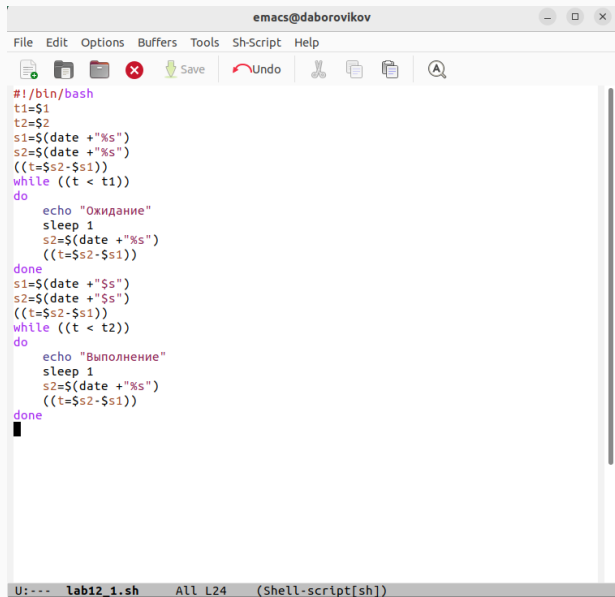
- Боровиков Даниил Александрович
- Студент ФМиЕН РУДН
- Группа НПИбд-01-22

## Вводная часть

---

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

## Написание скрипта lab12\_1.sh



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

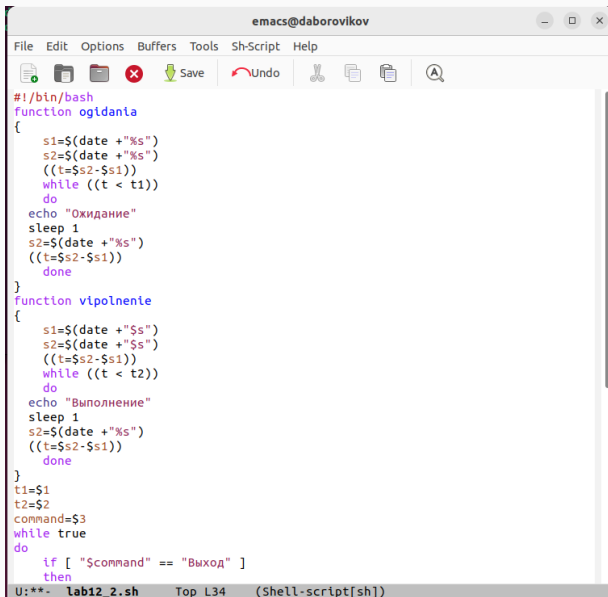
#!/bin/bash
t1=$1
t2=$2
s1=$(date +%s)
s2=$(date +%s)
((t=s2-s1))
while ((t < t1))
do
    echo "Ожидание"
    sleep 1
    s2=$(date +%s)
    ((t=s2-s1))
done
s1=$(date +%s)
s2=$(date +%s)
((t=s2-s1))
while ((t < t2))
do
    echo "Выполнение"
    sleep 1
    s2=$(date +%s)
    ((t=s2-s1))
done
█

U:--- lab12_1.sh All L24 (Shell-script[sh])
```

```
user@daborovikov:~$ chmod +x lab12_1.sh
user@daborovikov:~$ ./lab12_1.sh 4 7
Ожидание
Ожидание
Ожидание
Ожидание
```

Рис. 2: Право на выполнение и запуск lab12\_1.sh

## Написание скрипта lab12\_2.sh



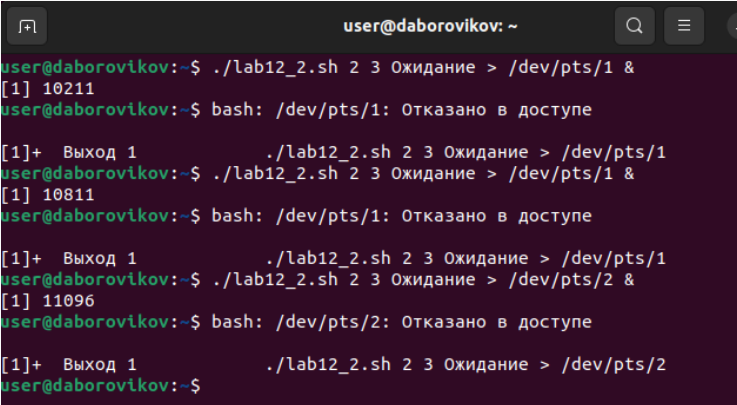
```
#!/bin/bash
function ogidania
{
    s1=$(date +%s)
    s2=$(date +%s)
    ((t=s2-s1))
    while ((t < t1))
    do
        echo "Ожидание"
        sleep 1
        s2=$(date +%s)
        ((t=s2-s1))
    done
}
function vipolnenie
{
    s1=$(date +%s)
    s2=$(date +%s)
    ((t=s2-s1))
    while ((t < t2))
    do
        echo "Выполнение"
        sleep 1
        s2=$(date +%s)
        ((t=s2-s1))
    done
}
t1=$1
t2=$2
command=$3
while true
do
    if [ "$command" == "Выход" ]
    then

```

U:\*~ lab12\_2.sh Top L34 (Shell-script[sh])



## Право на выполнение и запуск lab12\_2.sh



```
user@daborovikov: ~  
user@daborovikov:~$ ./lab12_2.sh 2 3 Ожидание > /dev/pts/1 &  
[1] 10211  
user@daborovikov:~$ bash: /dev/pts/1: Отказано в доступе  
[1]+  Выход 1      ./lab12_2.sh 2 3 Ожидание > /dev/pts/1  
user@daborovikov:~$ ./lab12_2.sh 2 3 Ожидание > /dev/pts/1 &  
[1] 10811  
user@daborovikov:~$ bash: /dev/pts/1: Отказано в доступе  
[1]+  Выход 1      ./lab12_2.sh 2 3 Ожидание > /dev/pts/1  
user@daborovikov:~$ ./lab12_2.sh 2 3 Ожидание > /dev/pts/2 &  
[1] 11096  
user@daborovikov:~$ bash: /dev/pts/2: Отказано в доступе  
[1]+  Выход 1      ./lab12_2.sh 2 3 Ожидание > /dev/pts/2  
user@daborovikov:~$
```

Рис. 4: Право на выполнение и запуск lab12\_2.sh

## Написание скрипта lab12\_3.sh

fi" data-bbox="187 138 802 739"/>

```
emacs@daborovikov
File Edit Options Buffers Tools Sh-Script Help
[Icons: New, Open, Save, Close, Undo, Redo, Cut, Copy, Paste, Search]
#!/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
```

Рис. 5: Написание скрипта lab12\_3.sh

Право на выполнение и запуск lab12\_3.sh](7.png){#fig:007 width=70%}

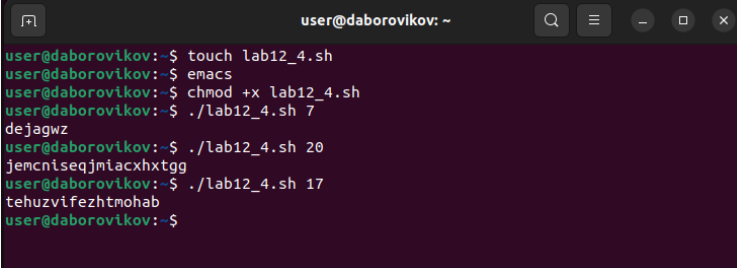
# Написание скрипта lab12\_4.sh



```
#!/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
```

Рис. 6: Написание скрипта lab12\_4.sh

## Право на выполнение и запуск lab12\_4.sh



```
user@daborovikov: ~  
user@daborovikov:~$ touch lab12_4.sh  
user@daborovikov:~$ emacs  
user@daborovikov:~$ chmod +x lab12_4.sh  
user@daborovikov:~$ ./lab12_4.sh 7  
dejagwz  
user@daborovikov:~$ ./lab12_4.sh 20  
jemcniseqjmiacxhxtgg  
user@daborovikov:~$ ./lab12_4.sh 17  
tehuzvifezhtmohab  
user@daborovikov:~$
```

Рис. 7: Право на выполнение и запуск lab12\_4.sh

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