

Лабораторная работа №12

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

Матвеева Анастасия Сергеевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	11
	Список литературы	12

Список иллюстраций

2.1	...	6
2.2	...	7
2.3	...	8
2.4	...	8
2.5	...	9
2.6	...	9
2.7	...	9
2.8	...	10
2.9	...	10

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

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 фоновом, а в привилегированном режиме. Доработать программу так, чтобы имела возможность взаимодействия трёх и более процессов.

A screenshot of a terminal window with a dark background. It shows two lines of text: the first line is 'asmatveeva1@dk2n24 ~ \$ touch lab12.sh' and the second line is 'asmatveeva1@dk2n24 ~ \$ chmod +x lab12.sh'. The text is in a monospaced font with green and blue colors for the prompt and command parts.

```
asmatveeva1@dk2n24 ~ $ touch lab12.sh
asmatveeva1@dk2n24 ~ $ chmod +x lab12.sh
```

Рис. 2.1: ...

```
#!/bin/bash
lockfile="./lockfile"
exec {fn}>$lockfile
echo "lock"
until flock -n ${fn}
do
    echo "not lock"
    sleep 1
    flock -n ${fn}
done
for ((i=0;i<=5; i++))
do
    echo "work"
    sleep 1
done
```

Рис. 2.2: ...

```
asmatveeva1@dk2n24 ~ $ ./lab12.sh
lock
not lock
not lock
not lock
not lock
not lock
not lock
^Xnot lock
^C
asmatveeva1@dk2n24 ~ $
```

Рис. 2.3: ...

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`.

```
asmatveeva1@dk2n24 ~ $ touch lab12-1.sh
asmatveeva1@dk2n24 ~ $ chmod +x lab12-1.sh
```

Рис. 2.4: ...


```
#1/bin/bash
cd /urs/share/man/man1
less $1*
```

Рис. 2.5: ...

```
asmatveeva1@dk2n24 ~ $ touch lab12-2.sh
asmatveeva1@dk2n24 ~ $ chmod +x lab12-2.sh
```

Рис. 2.6: ...

- 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.

```
#!/bin/bash
M=10
c=1
d=1
echo
echo "10 random words:"
while (($c≠(($M+1))))
do
    echo $((i=1;i<10;i++)); do printf '%s' "${RANDOM:0:1}; done) '[0-9]' '[a-z]
s]'
    echo $d
    ((c+=1))
    ((d+=1))
done
```

Рис. 2.7: ...

```
asmatveeva1@dk2n24 ~ $ ./lab12-2.sh
```

Рис. 2.8: ...

```
10 random wrds:  
bcccbbbbfc  
1  
bhcccccdcc  
2  
bidgccbgcc  
3  
bcbccccgcc  
4  
dbbbbbbdfif  
5  
cbbcebbgcb  
6  
bgdihbcbjc  
7  
dcbccbbbch  
8  
dccbcbbcbbj  
9
```

Рис. 2.9: ...

3 Выводы

Мы научились программировать в оболочке ОС UNIX и научились более сложными командами.

Список литературы