

отчёта по лабораторной работе 13

Ндри Ив Алла Ролан

Содержание

0.1	Цель работы	4
0.2	Выполнение работы.	4
0.3	Вывод	9
0.4	Ответы на контрольные вопросы	9

List of Tables

List of Figures

0.1 Цель работы

Познакомиться с операционной системой Linux. Получить практические навыки работы с редактором Emacs.

0.2 Выполнение работы.

1 1 Осуществили вход в систему, создали текстовый документ, затем перешли в него. Написали командный файл, реализующий упрощённый механизм семафоров. Командный файл в течение некоторого времени t_1 дожидается освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использует его в течение некоторого времени $t_2 > t_1$, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустили командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой, в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработали программу так, чтобы имелась возможность взаимодействия трёх и более процессов.

(рис. ??) (рис. ??)

```
ndriyvesalla@ndriyvesalla:~  
Файл Правка Вид Поиск Терминал Справка  
File Edit Options Buffers Tools Sh-Script Help  
#!/bin/bash  
M=10  
c=1  
d=1  
k=0  
echo  
echo <---> "10 Random Numbers"  
    while (($c!=($M+1)))  
do  
    n=$RANDOM  
    if k=0  
    then sleep 10  
    fi  
    echo $n  
    echo $d  
    ((c+=1))  
    ((d+=1))  
done  
UU-:----F1 labs10.sh All L1 (Shell-script[bash]) -----  
Loading vc-git...done
```

```

[ndriyvesalla@ndriyvesalla ~]$ emacs labs10.sh

[1]+  Stopped                  emacs labs10.sh
[ndriyvesalla@ndriyvesalla ~]$ emacs labs10.sh

[2]+  Stopped                  emacs labs10.sh
[ndriyvesalla@ndriyvesalla ~]$ chmod +x labs10.sh
[ndriyvesalla@ndriyvesalla ~]$ bash labs10.sh

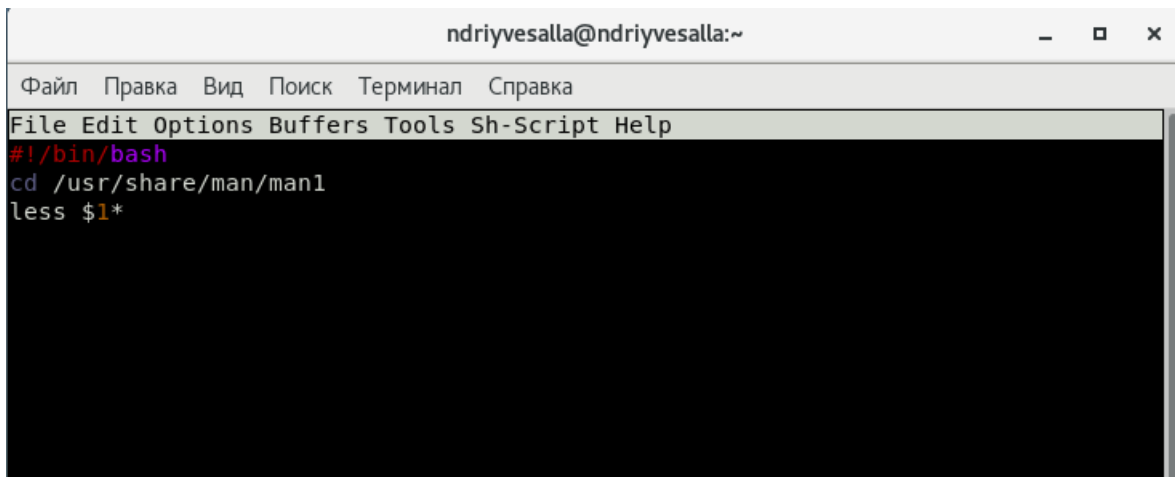
labs10.sh: line 7: ---: Нет такого файла или каталога
23750
1
31315
2
821
3
9092
4
21860
5
9938
6
16269
7
392
8
3419
9
22045
10
[ndriyvesalla@ndriyvesalla ~]$ █

```

{ # fig:001 width=70%}

fig:002 width=70%}

2 С помощью командного файла реализуем команды `man`. Изучим содержимое каталога `usr/share/man/man1`. Каждый архив можно открыть при помощи команды `less`. Командный файл получает в виде аргументы командной строки название команды и в виде результат выдает справку. (рис. ??) (рис. ??)



```
[4]+ Stopped emacs lab10_2.sh
[ndriyvesalla@ndriyvesalla ~]$ cd
[ndriyvesalla@ndriyvesalla ~]$ chmod +x lab10_2.sh
[ndriyvesalla@ndriyvesalla ~]$ bash lab10_2.sh
[ndriyvesalla@ndriyvesalla ~]$ emacs lab10_2.sh
```

{ # fig:003 width=70%}

{ #

fig:004 width=70%}

3.3 Используя встроенную переменную \$RANDOM напишем командный файл, генерирующий случайную последовательность цифр. (рис. ??) (рис. ??)

```
Файл Правка Вид Поиск Терминал Справка
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
echo "vvedite kolichechest vvodimuh boukv"
read g
Denominations="Q
W
E
R
T
Y
U
I
O
P
A
S
D
F
G
H
J
K
L
Z
X
C
V
B
N
M"
denomination=($Denominations)
num_denominations=${#denomination[*]}
g=$g+1
for ((i=1;i<$g;i++))
do
echo -n "${denomination [${(RANDOM%num_denominations)}]} "
done

-UU-:----F1 lab10_3.sh All L1 (Shell-script[bash]) -----
Loading vc-git...done
```

{ #


```

[ndriyvesalla@ndriyvesalla ~]$ bash lab10_3.sh
vvedite kolichechest vvodimuh boukv
A
[ndriyvesalla@ndriyvesalla ~]$ bash lab10_3.sh
vvedite kolichechest vvodimuh boukv
Q
[ndriyvesalla@ndriyvesalla ~]$ emacs lab10_3.sh
[3]+ Stopped emacs lab10_3.sh
[ndriyvesalla@ndriyvesalla ~]$ bash lab10_3.sh
vvedite kolichechest vvodimuh boukv
S
[ndriyvesalla@ndriyvesalla ~]$ emacs lab10_3.sh

```

fig:005 width=70%}

{ # fig:006 width=70%}

0.3 Вывод

Мы изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы, с использованием логических управляющих конструкций и циклов.

0.4 Ответы на контрольные вопросы

- 1) Нужно поставить пробелы после [и до].
- 2) С помощью знака >. 3) seq — Утилита, выводзяая последовательность целых чисел с шагом, заданным пользователем. Так же можно реализовать с помощью утилиты jot.
- 3) 3
- 4) В zsh можно настроить отдельные сочетания клавиш так, как вам нравится. Использование истории команд в zsh ничем особенным не отличается от bash. Zsh очень удобен для повседневной работы и делает добрую половину рутины за вас. Но стоит обратить внимание на различия между этими двумя оболочками. Например, в zsh после for обязательно вставлять пробел, нуме-

рация массивов в `zsh` начинается с 1, чего совершенно невозможно понять. Так, если вы используете `shell` для повседневной работы, исключаящей написание скриптов, используйте `zsh`. Если вам часто приходится писать свои скрипты, только `bash`! Впрочем, можно комбинировать.

- 5) Верно.
- 6) `Bash` является языком программирования очень высокого уровня. При должном навыке программиста, он может работать быстрее, чем `Си`. Однако его сложностью является то, что нужно следовать строгим правилам синтаксиса при написании программы.