

Презентация по лабораторной работе 12

Елизавета Александровна Гайдамака

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

Задание

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 имелась возможность взаимодействия трёх и более процессов.

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`.
3. 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.

1. Найдите синтаксическую ошибку в следующей строке:

```
while [$1 != "exit"]
```

Надо "\$1".

2. Как объединить (конкатенация) несколько строк в одну?

Самый простой способ объединить две или более строковые переменные — записать их одну за другой

3. Найдите информацию об утилите `seq`. Какими иными способами можно реализовать её функционал при программировании на `bash`?

Можно самому написать числа.

4. Какой результат даст вычисление выражения $\$((10/3))$?

3

5. Укажите кратко основные отличия командной оболочки zsh 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, нумерация 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.

6. Проверьте, верен ли синтаксис данной конструкции:

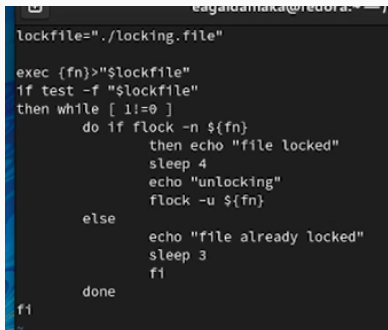
```
for ((a=1; a <= LIMIT; a++))
```

Да.

7. Сравните язык `bash` с какими-либо языками программирования. Какие преимущества у `bash` по сравнению с ними? Какие недостатки?

В `bash` есть много встроенных команд.

Код задания 1.

A screenshot of a terminal window with a dark background. The prompt is 'eagaidamaka@redora: ~'. The script content is as follows:

```
lockfile="./locking.file"

exec {fn}>"$lockfile"
if test -f "$lockfile"
then while [ 1!=0 ]
do if flock -n ${fn}
then echo "file locked"
sleep 4
echo "unlocking"
flock -u ${fn}

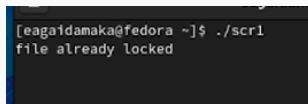
else

echo "file already locked"
sleep 3
fi

done
fi
```

Рис. 1: Рис.1

Результат работы.

A terminal window with a dark background and light blue text. The prompt is [eagaidamaka@fedora ~]\$ and the command is ./scr1. The output is file already locked.

```
[eagaidamaka@fedora ~]$ ./scr1  
file already locked
```

Рис. 2: Рис.2

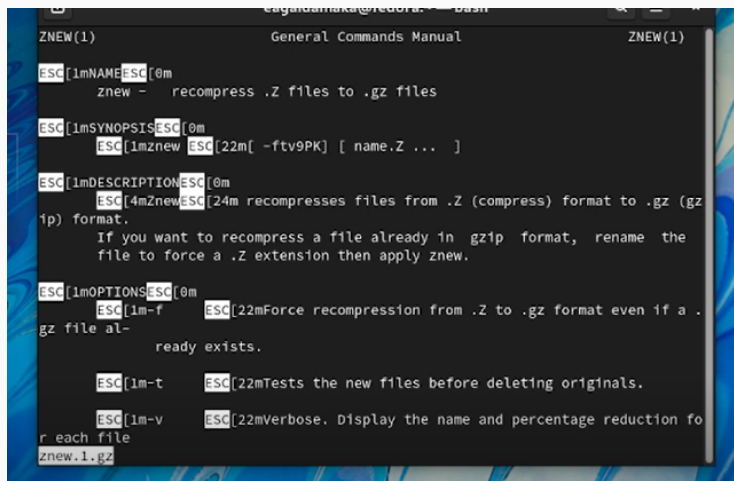
Код файла срр задания 2.

A terminal window with a dark background and a blue vertical bar on the left. It contains three lines of shell script code: 'cd /usr/share/man/man1', 'file=\$(ls | grep \$1)', and 'less \$file'. A white cursor is positioned at the end of the third line.

```
cd /usr/share/man/man1
file=$(ls | grep $1)
less $file
```

Рис. 3: Рис.3

Результат работы.



```
ZNEW(1)                                General Commands Manual                                ZNEW(1)

ESC[1mNAMEESC[0m
    znew - recompress .Z files to .gz files

ESC[1mSYNOPSISESC[0m
    ESC[1mznew ESC[22m[ -ftv9PK] [ name.Z ... ]

ESC[1mDESCRIPTIONESC[0m
    ESC[4mZnewESC[24m recompresses files from .Z (compress) format to .gz (gzip) format.
    If you want to recompress a file already in gzip format, rename the file to force a .Z extension then apply znew.

ESC[1mOPTIONSESC[0m
    ESC[1m-f ESC[22mForce recompression from .Z to .gz format even if a .gz file already exists.

    ESC[1m-t ESC[22mTests the new files before deleting originals.

    ESC[1m-v ESC[22mVerbose. Display the name and percentage reduction for each file
znew.1.gz
```

Рис. 4: Рис.4

Код задания 3.

A screenshot of a terminal window with a dark background. The prompt is 'eagandamika@fedor'. The command entered is 'echo \$RANDOM | tr '0-9' 'a-zA-z''. The cursor is at the end of the command.

```
eagandamika@fedor  
echo $RANDOM | tr '0-9' 'a-zA-z'
```

Рис. 5: Рис.5

Результат работы.

```
[eaga1damaka@fedora ~]$ ./scr3  
daiha  
[eaga1damaka@fedora ~]$ ./scr3  
dbh1h  
[eaga1damaka@fedora ~]$ ./scr3  
cbjdd  
[eaga1damaka@fedora ~]$
```

Рис. 6: Рис.6

Благодаря данной работе я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.