

Отчёт по лабораторной работе 12

Елизавета Александровна Гайдамака

Содержание

Цель работы	3
Задание	4
Теоретическое введение	6
Выполнение лабораторной работы	8
Выводы	10

Цель работы

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.

Теоретическое введение

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, исключающей 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.

Выполнение лабораторной работы

Код задания 1.

```
eagaidamaka@fedora: ~$ cat script1.sh
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.

```
eagaidamaka@fedora ~]$ ./script1.sh
file already locked
```

Рис. 2: Рис.2

Код файла сpp задания 2.

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

Рис. 3: Рис.3

Результат работы.

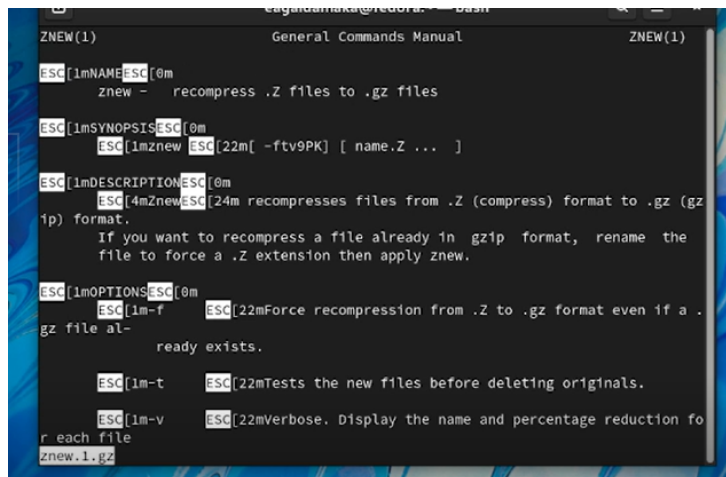


Рис. 4: Рис.4

Код задания 3.

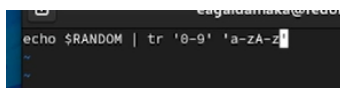


Рис. 5: Рис.5

Результат работы.

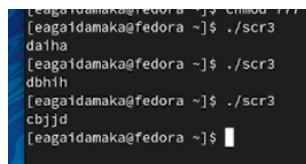


Рис. 6: Рис.6

Выводы

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