

Отчёт по лабораторной работе №12

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

Киньябаева Аиша Иделевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Контрольные вопросы	10
5	Выводы	12

Список иллюстраций

3.1	1 задание	7
3.2	Результат	7
3.3	2 задание”	8
3.4	Результат	8
3.5	3 задание	9
3.6	Результат	9

Список таблиц

1 Цель работы

Целью данной работы является изучение основ программирования в оболочке ОС UNIX, обучение более сложным командным файлам с использованием логических управляющих конструкций и циклов.

2 Задание

Создание командных файлов

3 Выполнение лабораторной работы

Пишу первую программу, которая должна копировать действия команды семафор (рис. 3.1)

```
home > aikinjabaeva > work > os > lab12 > $ semafor.sh
1  #!/bin/bash
2  if [ $# -ne 3 ];
3  then
4      echo "Необходимо задать три параметра: t1, t2 и имя процесса"
5      exit 1
6  fi
7
8  t1=$1
9  t2=$2
10 name=$3
11
12 touch semafor
13 echo "1" > semafor
14
15 while true;
16 do
17     if [ "$(cat semafor)" = "1" ];
18     then
19         echo "Ресурс свободен, ожидаем его освобождения"
20         sleep $t1
21         echo "$name: Захватываю ресурс"
22         echo "0" > semafor
23     fi
24     echo "$name: Использую ресурс"
25     sleep $t2
26     echo "$name: Освобождаю ресурс"
27     echo "1" > semafor
28 done
29
```

Рис. 3.1: 1 задание

Далее запускаю программу и смотрю на полученный результат (рис. 3.2)

```
aikinjabaeva@aikinjabaeva-VirtualBox:~/work/os/lab12$ bash semafor.sh 5 10 aisha
ha
Ресурс свободен, ожидаем его освобождения
sleep: неверный временной интервал «aisha: Захватываю ресурс»
По команде «sleep --help» можно получить дополнительную информацию.
aisha: Использую ресурс
aisha: Освобождаю ресурс
Ресурс свободен, ожидаем его освобождения
sleep: неверный временной интервал «aisha: Захватываю ресурс»
По команде «sleep --help» можно получить дополнительную информацию.
aisha: Использую ресурс
```

Рис. 3.2: Результат

Пишу вторую программу, которая должна осуществлять поиск и вывод справок по командам (рис. 3.3)

```
home > aikinjyabaeva > work > os > lab12 > $ main2.sh
1  #!/bin/bash
2  if [ -z "$1" ]; then
3      echo "Название команды не задано"
4      exit 1
5  fi
6
7  file=$(find /usr/share/man/man1 -name "$1.1.gz")
8
9  if [ -z "$file" ]; then
10     echo "Справка по команде $1 не найдена"
11     exit 2
12 fi
13
14 gunzip -c "$file" | less
```

Рис. 3.3: 2 задание”

Запускаю программу и смотрю полученный результат, т.е. справка на определенную команду (рис. 3.4)

```
.\
.\
.\ Copyright 1988, 1998 The Open Group
.\
.\
.\ Permission to use, copy, modify, distribute, and sell this software and its
.\ documentation for any purpose is hereby granted without fee, provided that
.\ the above copyright notice appear in all copies and that both that
.\ copyright notice and this permission notice appear in supporting
.\ documentation.
.\
.\
.\ The above copyright notice and this permission notice shall be included
.\ in all copies or substantial portions of the Software.
.\
.\
.\ THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
.\ OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
.\ MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
.\ IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR
.\ OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
.\ ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
.\ OTHER DEALINGS IN THE SOFTWARE.
.\
```

Рис. 3.4: Результат

Следующая программа выводит случайную последовательность латинских букв (у меня это послезовательность из 10 символов) (рис. 3.5), (рис. 3.6)


```

home > aikinyabaeva > work > os > lab12 > $ main3.sh
1  #!/bin/bash
2  alphabet="abcdefghijklmnopqrstuvwxyz"
3  length=${#alphabet}
4  rand_str=""
5  for (( i=0; i<10; i++ ))
6  do
7      index=$(( RANDOM % LENGTH ))
8      rand_str+="${alphabet:index:1}"
9  done
10 echo "Случайная последлвательность букв: $rand_str"

```

Рис. 3.5: 3 задание

```

aikinyabaeva@aikinyabaeva-VirtualBox:~/work/os/lab12$ bash main3.sh
Случайная последлвательность букв: rkcjmbuhuf
aikinyabaeva@aikinyabaeva-VirtualBox:~/work/os/lab12$

```

Рис. 3.6: Результат

Ну и в завершение работы выкладываю все на гит

4 Контрольные вопросы

1. Ошибка в этой строке заключается в использовании квадратных скобок. В командной оболочке Bash они используются для выполнения условных выражений вместе с командой `test`. Правильно будет так: `while (($1 != "exit"))`
2. В bash для объединения нескольких строк в одну можно использовать оператор конкатенации `+` или `..`. Например: `string1="Hello" string2="world" result=$string1$string2 echo $result`

Вывод будет: Helloworld

3. Утилита `seq` в Linux используется для создания итераторов, которые генерируют последовательности чисел. Она имеет следующий синтаксис: `seq ОПЦИИ... ПОСЛЕДНЕЕ`. Например, команда `"seq 1 5"` выведет последовательность чисел от 1 до 5. Некоторые из опций, которые можно использовать с утилитой `seq`: `-f` `--format=ФОРМАТ` используйте указанный формат для каждого числа. `-s` `--separator=СЕПАРАТОР` используйте указанный разделитель между числами
4. Результат вычисления выражения будет равен 3, так как при использовании символов двойных круглых скобок происходит округление до ближайшего целого числа.
5. Некоторые отличия командной оболочки `zsh` от `bash` включают:

- Мощная автодополнение и расширенная подсказка, которые делают работу с командами более эффективной и быстрой.
- Zsh имеет более интуитивный синтаксис для управления массивами и строками, что делает ее удобным выбором для обработки текста.
- В zsh доступен “глубокий патчинг”, позволяющий изменять внутреннюю логику самой оболочки.
- Zsh имеет более развитую систему конфигурации и управления плагинами, что делает ее наиболее гибкой и настраиваемой оболочкой в мире Linux.

6. Синтаксис верный, но надо уточнить значение переменной LIMIT. Также, важно помнить, что в данном цикле нет тела цикла - то есть ничего не будет происходить, пока не будет добавлен код, который будет выполняться внутри цикла.

7. C++:

ПРЕИМУЩЕСТВА BASH: - Встроенная поддержка системных команд и утилит - Незаменим при работе с системными скриптами на Unix-подобных операционных системах - Гибкость и легкость в написании коротких скриптов, относительно C++

НЕДОСТАТКИ BASH: - Недостаточная мощность в сравнении с C++ - Ограниченный функционал - Сложные программы сложнее считывать и понимать в bash, чем в C++

5 Выводы

В ходе данной лабораторной работы были изучены основы программирования и созданы командные файлы.