

Лабораторная работа №14

Отчет

Зубов Иван Александрович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	12

Список иллюстраций

3.1	Пишем код	8
3.2	Смотрим вывод кода	8
3.3	Пишем код	9
3.4	Смотрим вывод кода	10
3.5	Пишем код	11
3.6	Смотрим вывод кода	11

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 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.

3 Выполнение лабораторной работы

Выполняем первое задание. Создаем каталог lab14 и файл 1.sh. Пишем код по заданию и проверяем результат кода

```
#!/bin/bash

if [ $# -ne 2 ]; then
    echo "Использование: $0 <t1> <t2>"
    exit 1
fi

t1=$1
t2=$2
semaphore_file="semaphore.lock"

# Создаем семафор, если его нет
touch "$semaphore_file"

function access_resource {
    while ! ln "$semaphore_file" "$0.lock" 2>/dev/null; do
        echo "Ресурс занят, ожидание освобождения..."
        sleep "$t1"
    done
}

U:--- 1.sh Top L17 (Shell-script[bash])
```

Рис. 3.1: Пишем код

```
ianzubov@ianzubov:~/lab14$ chmod +x 1.sh
ianzubov@ianzubov:~/lab14$ ./1.sh 10 10 &> /dev/pts/0
Ресурс освобожден, начало использования на 10 секунд
Ресурс освобожден, использование завершено
ianzubov@ianzubov:~/lab14$ sudo ./1.sh 10 10
Ресурс освобожден, начало использования на 10 секунд
Ресурс освобожден, использование завершено
ianzubov@ianzubov:~/lab14$
```

Рис. 3.2: Смотрим вывод кода

Создаем файл 2.sh для второго задания. Пишем код и проверяем работу кода


```
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Использование: $0 <название_команды>"
    exit 1
fi

command_name=$1
man_directory="/usr/share/man/man1"

if [ -f "$man_directory/$command_name.1.gz" ]; then
    zcat "$man_directory/$command_name.1.gz" | less
else
    echo "Справка для команды '$command_name' не найдена"
fi
```

Рис. 3.3: Пишем код

```
ianzubov@ianzubov:/home/ianzubov/lab14

.\\" DO NOT MODIFY THIS FILE! It was generated by help2man 1.48.5.
.TH LS "1" "November 2024" "GNU coreutils 9.5" "User Commands"
.SH NAME
ls \- list directory contents
.SH SYNOPSIS
.B ls
[\fI\,OPTION\|\fR]... [\fI\,FILE\|\fR]...
.SH DESCRIPTION
.\" Add any additional description here
.PP
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of \fB\--cftuvSUX\|fR nor \fB\--sort\|fR is specified.
.PP
Mandatory arguments to long options are mandatory for short options too.
.TP
\fB\--a\|fR, \fB\--all\|fR
do not ignore entries starting with .
.TP
\fB\--A\|fR, \fB\--almost-all\|fR
do not list implied . and ..
.TP
\fB\--author\|fR
with \fB\--l\|fR, print the author of each file
.TP
\fB\--b\|fR, \fB\--escape\|fR
print C\--style escapes for nongraphic characters
.TP
\fB\--block-size\|fR=\fI\,SIZE\|\fR
with \fB\--l\|fR, scale sizes by SIZE when printing them;
e.g., '\--block-size=M'; see SIZE format below
.TP
\fB\--B\|fR, \fB\--ignore-backups\|fR
do not list implied entries ending with ~
.TP
\fB\--c\|fR
with \fB\--lt\|fR: sort by, and show, ctime (time of last
change of file status information);
with \fB\--l\|fR: show ctime and sort by name;
otherwise: sort by ctime, newest first
.TP
\fB\--C\|fR
list entries by columns
.TP
\fB\--color\|fR[=\fI\,WHEN\|\fR]
color the output WHEN; more info below
.TP
:
```

Рис. 3.4: Смотрим вывод кода

Делаем третье задания

Рис. 3.5: Пишем код

```
ianzubov@iazubov:~/lab14$ touch 3.sh
ianzubov@iazubov:~/lab14$ chmod +x 3.sh
ianzubov@iazubov:~/lab14$ ./3.sh
Случайная последовательность букв латинского алфавита:
UXLVPRLZXB
ianzubov@iazubov:~/lab14$ ./3.sh
Случайная последовательность букв латинского алфавита:
GNLKRAOIUI
ianzubov@iazubov:~/lab14$ ./3.sh
Случайная последовательность букв латинского алфавита:
GMQVPVONIV
ianzubov@iazubov:~/lab14$
```

Рис. 3.6: Смотрим вывод кода

4 Выводы

Я научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.