

Лабораторная работа №14

Презентация

Зубов И.А.

16 мая 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Зубов Иван Александрович
- Студент
- Российский университет дружбы народов
- 1132243112@pfur.ru

Задания

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

Выполнение лабораторной работы

Создаем файл 1.sh, пишем код и проверяем его работу

```
#!/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
    echo "Ресурс освобожден, начало использования на $t2 секунд"
    sleep "$t2"
    echo "Ресурс освобожден, использование завершено"
}

# Пример использования
access_resource "$t1" "$t2"
```

U1:-- 1.sh top 1.17 (Shell-script{bash})

```
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$
```

Задание 2

Создаем файл 2.sh пишем код и проверяем его работу

```
#!/bin/bash

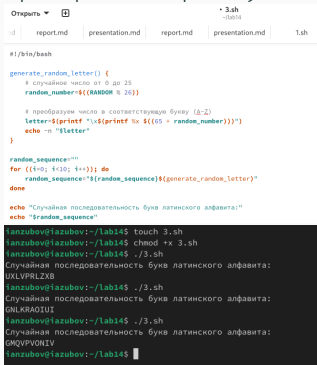
if [ $# -ne 1 ]; then
    echo "Использование: $0 <название_команды>"
    exit 1
fi

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
```

```
ianzubov@ianzubov/home/ianzubov/lab14:
% ls -l NOT MODIFY THIS FILE: It was generated by help2man 1.48.5.
% ls -l "11 November 2024" "GNU coreutils 9.5" "User Commands"
% SH NAME
% ls -l list directory contents
% SH SYNOPSIS
% s ls
% {f1,OPTION.../fR}... {f2,f1LE/fR}...
% SH DESCRIPTION
% s Add any additional description here
%
% list information about the FILES (the current directory by default).
% Sort entries alphabetically if none of /VB/-<fuuSUX/fR nor /VB/-<sort/fR is specified.
%
% Mandatory arguments to long options are mandatory for short options too.
%
% .TP
% /VB/-a/fR, /VB/-a-ll/fR
% do not ignore entries starting with .
%
% /VB/-A/fR, /VB/-A-most-ll/fR
% do not list implied . and ..
%
% .TP
% /VB/-l-author/fR
% with /VB/-l/fR, print the author of each file
%
% .TP
% /VB/-b/fR, /VB/-l-escape/fR
% print C-style escapes for non-graphic characters
%
% .TP
% /VB/-l-block-size/fR,{f1,f2},SIZE/fR
% with /VB/-l/fR, scale sizes by SIZE when printing them;
% e.g., '-l--block-size=M' see SIZE format below
%
% .TP
% /VB/-l-B/fR, /VB/-l-ignore-backups/fR
% do not list implied entries ending with -
%
% .TP
% /VB/-c/fR
% with /VB/-l/fR: sort by, and show, ctime (time of last
% change of file status information);
% with /VB/-l/fR: show ctime and sort by name;
% otherwise: sort by ctime, newest first
%
% .TP
% s See also
```


Создаем файл 3.sh, пишем код и проверяем его работу



```
Открыть ▾ ⓘ • 3.sh -lab14
id | report.md | presentation.md | report.md | presentation.md | 1.sh

#!/bin/bash

generate_random_letter() {
    # случайное число от 0 до 25
    random_number=$((RANDOM % 26))

    # преобразуем число в соответствующую букву (A-Z)
    letter=$(printf "%x$(printf %x $((65 + random_number)))")
    echo -n "$letter"
}

random_sequence=""
for ((i=0; i<10; i++)); do
    random_sequence+=$(generate_random_letter)
done

echo "Случайная последовательность букв латинского алфавита:"
echo "$random_sequence"

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$
```