

Лабораторная работа № 14

Программирование в командном процессоре ОС Unix. Расширенное программирование

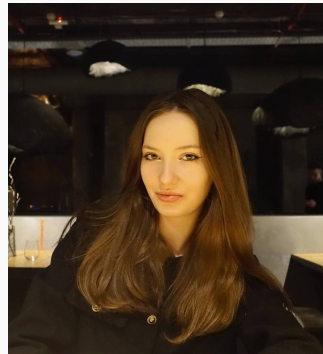
Скобеева А.А.

07 марта 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Скобеева Алиса Алексеевна
- студентка 1-го курса направления “Прикладная информатика”
- Российский университет дружбы народов
- 1132246836@pfur.ru



Вводная часть

- Данная презентация актуальна для студентов технических направлений, изучающих предмет “Архитектура компьютеров: операционные системы”, а также для всех, кто интересуется программированием в командном процессоре ОС Unix

- Командный процессор ОС Unix
- Fedora Sway

- Изучить ОП в оболочке ОС Unix
- Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов

- Github
- Материалы ТУИС

Основная часть

1-ый командный файл

- Пишем командный файл, реализующий упрощенный механизм семафоров. Также дорабатываем программу так, чтобы имелась возможность взаимодействия трёх и более процессов.

```
foot
GNU nano 2.1                                semafor1.sh
#!/bin/bash

# Имя файла-семафора
semaphore_file="/tmp/my_semaphore"

# Время ожидания (t1) в секундах
wait_time=5

# Время удержания (t2) в секундах
hold_time=3

# Функция для захвата семафора
acquire_semaphore() {
    if [ -e "$semaphore_file" ]; then
        echo "Процесс $$ ждет освобождения ресурса..."
        start_time=$(date +%s)
        while [ -e "$semaphore_file" ]; do
            sleep 1
            current_time=$(date +%s)
            elapsed_time=$((current_time - start_time))
            if [ "$elapsed_time" -ge "$wait_time" ]; then
                echo "Процесс $$ превысил время ожидания."
            fi
        done
    fi
}
```

Итог работы 1-го файла

```
1 2 foot 29% 100% 10.0.2.15/24 17% 15% us 32% 22:27
foot
h
[aaskobeeva@fedora ~]$ nano semafor1.sh
[aaskobeeva@fedora ~]$ ./semafor1.sh > /dev/tty2 &
[1] 4484
[aaskobeeva@fedora ~]$ ./semafor1.sh > /dev/tty3 &
[2] 4493
bash: /dev/tty3: Отказано в доступе
[1] Завершён ./semafor1.sh > /dev/tty2
[aaskobeeva@fedora ~]$ sudo ./semafor1.sh > /dev/tty2 &
[3] 4665
[aaskobeeva@fedora ~]$ [sudo] пароль для aaskobeeva:
123
bash: 123: команда не найдена
[2] Выход 1 ./semafor1.sh > /dev/tty3
[3]+ Остановлен sudo ./semafor1.sh > /dev/tty2
[aaskobeeva@fedora ~]$ ./semafor1.sh > /dev/pts/2 &
[4] 4722
[aaskobeeva@fedora ~]$ ./semafor1.sh > /dev/pts/3 &
[5] 4756
[4] Завершён ./semafor1.sh > /dev/pts/2
[aaskobeeva@fedora ~]$

foot
[aaskobeeva@fedora ~]$ tty
/dev/pts/2
[aaskobeeva@fedora ~]$ Процесс 4722 захватил ресурс.
Процесс 4722 использует ресурс в течение 3 секунд...
Процесс 4722 освободил ресурс.

foot
[aaskobeeva@fedora ~]$ tty
/dev/pts/3
[aaskobeeva@fedora ~]$ Процесс 4756 захватил ресурс.
Процесс 4756 использует ресурс в течение 3 секунд...
Процесс 4756 освободил ресурс.
```

2-ой командный файл

- Реализовываем команду `man` с помощью командного файла

```
foot
GNU nano 8.1                                mantask.sh
#!/bin/bash

# Каталог с файлами справки
man_dir="/usr/share/man/man1"

# Команда, для которой нужно получить справку
command="$1"

# Проверка наличия аргумента
if [ -z "$command" ]; then
    echo "Использование: ./my_man <команда>"
    exit 1
fi

# Имя файла справки
man_file="$man_dir/$command.1.gz"

# Проверка существования файла
if [ ! -f "$man_file" ]; then
    echo "Справка для команды '$command' не найдена."
    exit 1
fi

# Открываем файл с помощью less
zless "$man_file"
```

Итог работы 2-го файла

```
foot
.\" 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      I
```

3-ий командный файл

- Пишем командный файл, который используя встроенную переменную \$RANDOM генерирует случайную последовательность букв латинского алфавита

```
foot
GNU nano 8.1                                random.sh
#!/bin/bash

# Длина последовательности
length=10

# Алфавит
alphabet="abcdefghijklmnopqrstuvwxyz"

# Функция для генерации случайной буквы
generate_random_letter() {
    random_index=$((RANDOM % ${#alphabet}))
    echo "${alphabet:$random_index:1}"
}

# Генерация последовательности
random_string=""
for i in $(seq 1 $length); do
    random_string+="$(generate_random_letter)"
done

echo "$random_string"
```

```
[aaskobeeva@fedora ~]$ touch random.sh
[aaskobeeva@fedora ~]$ chmod +x random.sh
[aaskobeeva@fedora ~]$ nano random.sh
[aaskobeeva@fedora ~]$ ./random.sh
ncztugwknc
[aaskobeeva@fedora ~]$ ./random.sh
brhlotmebj
[aaskobeeva@fedora ~]$
```

- Мы успешно написали 3 программы с использованием логических управляющих конструкций и циклов