

Операционные системы

Лабораторная работа № 12. Программирование в командном процессоре ОС UNIX.
Расширенное программирование.

Абдеррахим Мугари.

29 апреля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Абдеррахим Мугари
- Студент
- Российский университет дружбы народов
- 1032215692@pfur.ru
- <https://github.com/iragoum>



- Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

- Терминал Unix.
- Файловая система.
- Текстовый редактор emacs .

- Здесь я написал файл `bash`, реализующий упрощенный механизм семафора. Командный файл должен некоторое время t_1 ждать освобождения ресурса, выдавая сообщение об этом, и после ожидания его освобождения использовать его в течение некоторого времени $t_2 < t_1$, также предоставляя информацию о том, что ресурс используется соответствующим командным файлом (процессом)

Написание первой программы:

```
sem_wait.sh - GNU Emacs at fedora
File Edit Options Buffers Tools Sh-Script Help

# Имя файла блокировки
LOCKFILE=/tmp/semaphore.lock

# Имя FIFO-файла
FIFOFILE=/tmp/semaphore.fifo

# Время ожидания освобождения ресурса в секундах
t1=5

# Время использования ресурса в секундах
t2=10

# Создание FIFO-файла
if [ ! -e $FIFOFILE ]; then
  mkfifo $FIFOFILE
fi

# Ожидание освобождения ресурса
while true
do
  if [ ! -f $LOCKFILE ]; then
    echo "Ресурс свободен. Получение блокировки."
    touch $LOCKFILE
    echo "$$" > $FIFOFILE # Запись PID процесса в FIFO-файл
    break
  else
    echo "Ресурс занят. Ожидание освобождения."
    sleep $t1
  fi
done

# Использование ресурса
echo "Ресурс используется на $t2 секунд."
echo "PID процесса: $$"
sleep $t2
```

2. Программное

звания в обол
ользованием

ыполнении

реализующий
ие некоторо
ение, а дож
ени t2<>t1,
зующим ком
уального тер
у#, где # — н
ен этот файл
так, чтобы и

ющую коман
ем находятс
вленных в си
разу же прос
умента коман
этой коман
ет в каталог
ную \$RANDOM
ность букв ла

```
sem_signal.sh - GNU Emacs at fedora
File Edit Options Buffers Tools Sh-Script Help

#!/bin/bash

# Имя файла блокировки
LOCKFILE=/tmp/semaphore.lock

# Имя FIFO-файла
FIFOFILE=/tmp/semaphore.fifo

# Освобождение ресурса
echo "Ресурс освобожден."
rm $LOCKFILE

# Чтение PID процесса из FIFO-файла и отправка ему сигнала
if [ -e $FIFOFILE ]; then
  read pid < $FIFOFILE
  echo "Отправка сигнала процессу $pid."
  kill -SIGUSR1 $pid
fi

U:--- sem_signal.sh All L3 (Shell-script[bash])
Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Emacs Tutorial Learn basic keystroke commands
Emacs Guided Tour Overview of Emacs features at gnu.org
View Emacs Manual View the Emacs manual using Info
Absence of Warranty GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions Conditions for redistributing and changing Emacs
Defining Macros Defining macros
```

Рис. 1: Написание первой программы

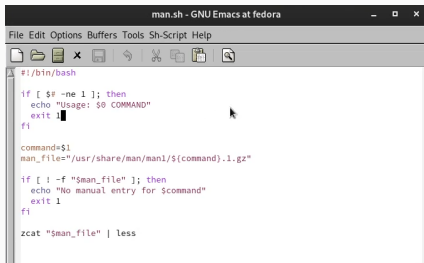
выполнение первой программы

- Здесь я выполнил код первой программы, и она заработала так, как и должна была работать

Рис. 2: выполнение первой программы

Написание второй программы

- После этого я написал скрипт `bash`, который использует содержимое `tar log /usr/share/man/man1`. Он содержит архивы текстовых файлов, содержащих справку по большинству программ и команд, установленных в системе, а затем отображает справку по выбранной команде



```
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Usage: $0 COMMAND"
    exit 1
fi

command=$1
man_file="/usr/share/man/man1/${command}.1.gz"

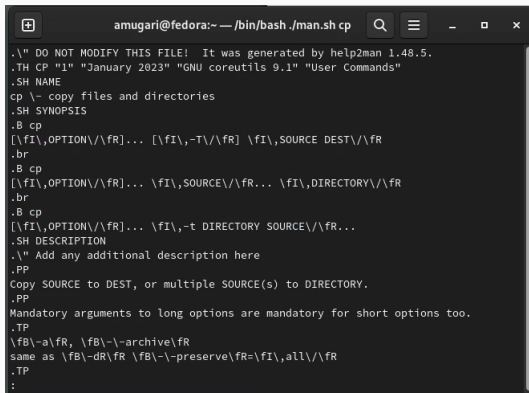
if [ ! -f "$man_file" ]; then
    echo "No manual entry for $command"
    exit 1
fi

zcat "$man_file" | less
```

Рис. 3: Написание второй программы

выполнение второй программы

- после этого я выполнил код скрипта, чтобы проверить справку команды cp, и он показал мне справку этой команды

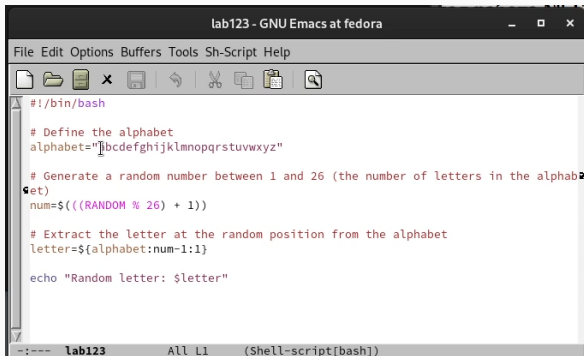
A terminal window with a dark background and light text. The title bar shows the user 'amugari' on a 'fedora' machine, in a shell '/bin/bash', running the command './man.sh cp'. The terminal output is the manual page for the 'cp' command, starting with a warning not to modify the file, followed by metadata like 'January 2023' and 'GNU coreutils 9.1'. It then lists the command name 'cp', its synopsis 'cp [- options] SOURCE DEST...', and a detailed description of its function: copying files and directories. It also lists various options like '-a', '-d', and '-p' with their descriptions, and ends with a colon.

```
amugari@fedora:~ — /bin/bash ./man.sh cp
.\" DO NOT MODIFY THIS FILE! It was generated by help2man 1.48.5.
.TH CP "1" "January 2023" "GNU coreutils 9.1" "User Commands"
.SH NAME
cp \- copy files and directories
.SH SYNOPSIS
.B cp
[\fI\,OPTION\|\fR]... [\fI\,-T\|\fR] \fI\,SOURCE DEST\|\fR
.br
.B cp
[\fI\,OPTION\|\fR]... \fI\,SOURCE\|\fR... \fI\,DIRECTORY\|\fR
.br
.B cp
[\fI\,OPTION\|\fR]... \fI\,-t DIRECTORY SOURCE\|\fR...
.SH DESCRIPTION
.\" Add any additional description here
.PP
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.
.PP
Mandatory arguments to long options are mandatory for short options too.
.TP
\fB\-a\fR, \fB\-\-archive\fR
same as \fB\-d\fR \fB\-\-preserve\fR=\fI\,all\|\fR
.TP
:
```

Рис. 4: выполнение второй программы

Написание третьей программы]

- и, наконец, используя переменную `$random`, я написал программу, которая генерирует случайный латинский алфавит



```
#!/bin/bash

# Define the alphabet
alphabet="abcdefghijklmnopqrstuvwxyz"

# Generate a random number between 1 and 26 (the number of letters in the alphabet)
num=$((RANDOM % 26) + 1)

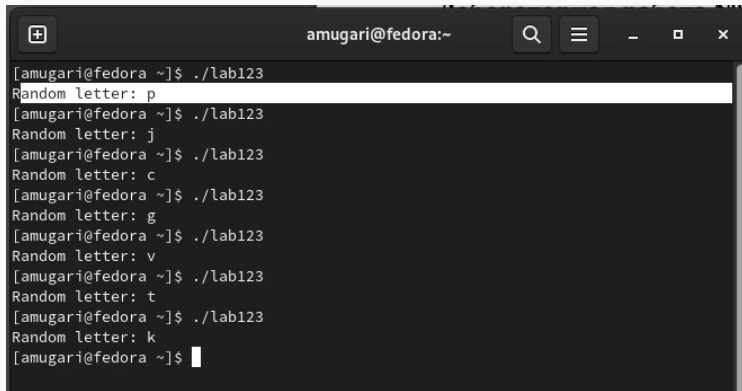
# Extract the letter at the random position from the alphabet
letter=${alphabet:num-1:1}

echo "Random letter: $letter"
```

Рис. 5: Написание третьей программы

выполнение третьей программы

- а потом я выполнил программу, и все заработало



```
amugari@fedora:~  
[amugari@fedora ~]$ ./lab123  
Random letter: p  
[amugari@fedora ~]$ ./lab123  
Random letter: j  
[amugari@fedora ~]$ ./lab123  
Random letter: c  
[amugari@fedora ~]$ ./lab123  
Random letter: g  
[amugari@fedora ~]$ ./lab123  
Random letter: v  
[amugari@fedora ~]$ ./lab123  
Random letter: t  
[amugari@fedora ~]$ ./lab123  
Random letter: k  
[amugari@fedora ~]$
```

Рис. 6: выполнение третьей программы

- В ходе этой лабораторной работы у меня была возможность научиться программировать в операционной системе UNIX и приобрести практические навыки написания сложного кода с использованием логических управляющих структур и циклов.

Выводы, согласованные с целью
работы:

Выводы, согласованные с целью работы:

- Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.