

Лабораторная работа №12.

Программирование в командном процессоре ОС UNIX.

Кучеренко С.М.

Российский университет дружбы народов, Москва, Россия

Информация

- Кучеренко София
- студент 1го курса НММбд-02-22
- Российский университет дружбы народов
- 1132226498@pfur.ru
- https://github.com/sshiperr/study_2022-2023_os-intro

Цель работы — изучить основы программирования в командной оболочке OS Unix.

Выполнение лабораторной работы

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.

```
#!/bin/bash
```

```
# Unpacking parameters
```

```
while getopts i:o:p:Cn flag  
do
```

```
    case $flag in
```

```
        i) inputFile=$OPTARG..
```

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`.

```
#!/bin/bash
```

```
if (($# < 1))
```

```
then
```

```
    echo fedya needs to eat more than one argument \  
    otherwise he will not work.  
    exit
```

```
fi
```

```
    /baton $1
```

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.

```
#!/bin/bash
```

```
ext=.tmp
```

```
function CreateFiles()
{
    i=1
    while ((i <= $1))
    do
        echo Creating $i$ext
        touch $i$ext
        let i++
    done
}
```


4. Написать командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find).

```
#!/bin/bash
if (($# < 1))
then
    target=.
else
    target=$1
fi

outputFile=$(pwd)\archive.tar
tar -cf $outputFile $(find $target -maxdepth 1 -atime -7 -type f)
if (($? == 0))
then
```

Итог



В ходе выполнения лабораторной работы были изучены основы программирования в командной оболочке OS Unix. Цель работы была достигнута.