

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра Фундаментальная информатика и информационные
технологии

ОТЧЕТ

по лабораторной работе № 13

дисциплина: *Операционные системы*

Студент: Абушек Дмитрий Олегович

Группа: НФИбд-01-20

МОСКВА

2021г.

Цель:

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

Ход работы:

1. Напишем командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени t2<>t1, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом)

```
lockfile="./lab13/locking.file"
exec fn>$lockfile
if test -f "$lockfile"
then
    while [1 !=0 ]
    do
        if flock -n ${fn}
        then
            echo "File was locked"
            sleep 2
            echo "Inlocking..."
            flock -u ${fn}
        else
            echo "File already locked"
            sleep 2
        fi
    done
fi
```

2. Реализуем команду man с помощью командного файла.

```
command=""
while getopts "n" opt
do
    case $opt in
        n) command="$OPTARG";;
    esac
done
if test -f "/usr/share/man/man1/$command.1.gz"
then less /usr/share/man/man1/"$command".1.gz
fi
```

3. Используя встроенную переменную \$RANDOM, напишем командный файл, генерирующий случайную последовательность букв латинского алфавита

```
echo $RANDOM | tr '0-9' 'a-zA-Z'
```

```
[dmitryabushek@MacBook-Pro-Dmitrij ~ % sh lab133.sh  
iacc
```

Вывод:

В ходе лабораторной работы мы изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов

Контрольные вопросы:

1. Найдите синтаксическую ошибку в следующей строке:

```
while [$1 != "exit"]
```

\$1.

Так же между скобками должны быть пробелы. В противном случае скобки и рядом стоящие символы будут восприниматься как одно целое

2. Как объединить (конкатенация) несколько строк в одну?

```
cat file.txt | xargs | sed -e 's/\./.\n/g'
```

3. Найдите информацию об утилите seq. Какими иными способами можно реализовать её функционал при программировании на bash?

seq - выдает последовательность чисел.

Реализовать ее функционал можно командой

```
for n in {1..5}
```

```
do <КОМАНДА>
```

```
done
```

4. Какой результат даст вычисление выражения $\$((10/3))$?

3

5. Укажите кратко основные отличия командной оболочки zsh от bash.

Zsh очень сильно упрощает работу. Но существуют различия. Например, в zsh после for обязательно вставлять пробел, нумерация массивов в zsh начинается с 1 (что не особо удобно на самом деле).

Если вы собираетесь писать скрипт, который легко будет запускать множество разработчиков, то рекомендуется Bash. Если скрипты вам не нужны - Zsh (более простая работа с файлами, например)

6. Проверьте, верен ли синтаксис данной конструкции `for ((a=1; a <= LIMIT; a++))`

Верен

7. Сравните язык `bash` с какими-либо языками программирования. Какие преимущества у `bash` по сравнению с ними? Какие недостатки?

`Bash` позволяет очень легко работать с файловой системой без лишних конструкций (в отличие от обычного языка программирования). Но относительно обычных языков программирования `bash` очень сжат. Тот же `Си` имеет гораздо более широкие возможности для разработчика.