

## **Лабораторная работа №1 по операционной системе**

Имя: Чу минь Тиеп

Группа: K33401

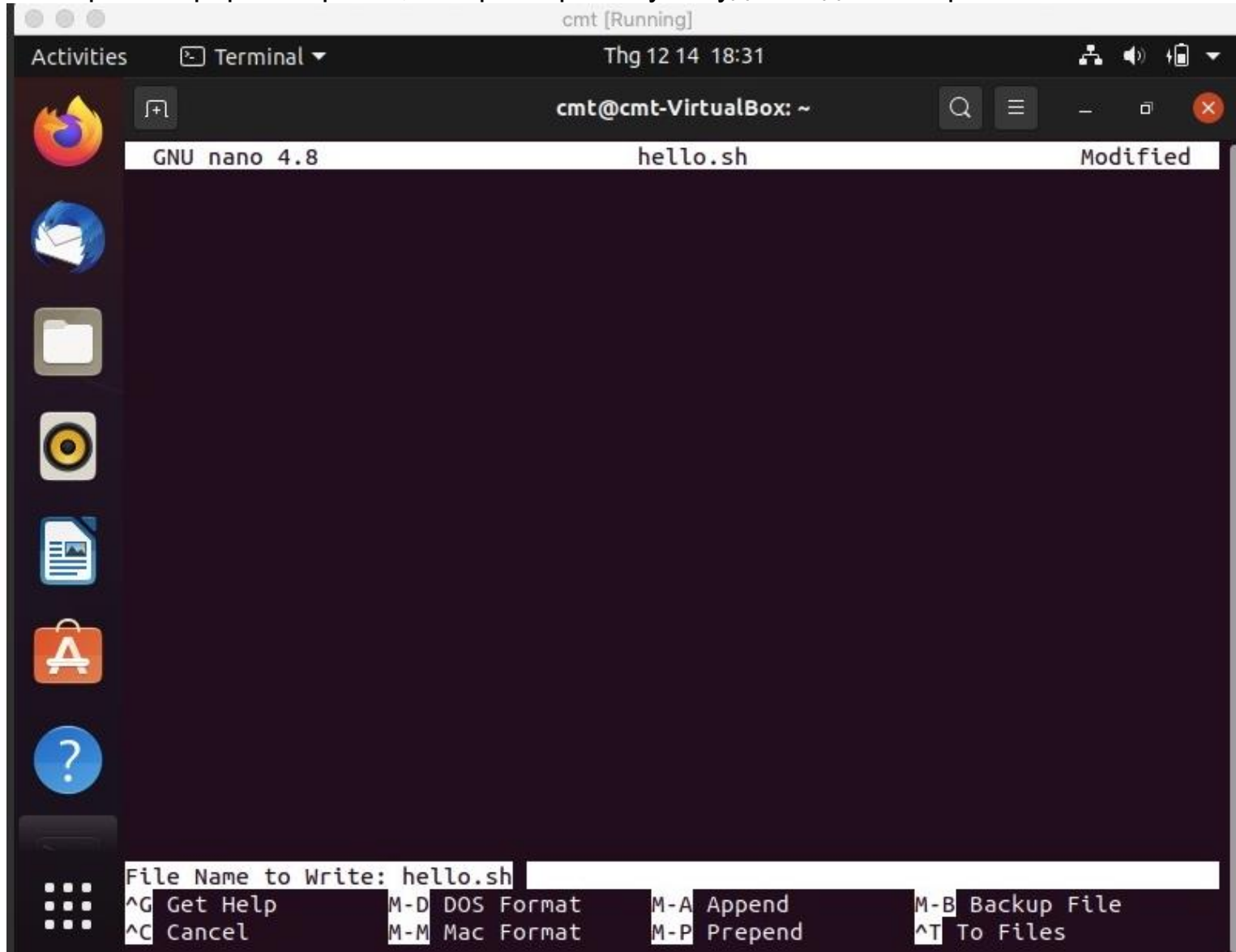
Преподаватель: Ватьян Александра Сергеевна

## Задание 1

1. Заменить путь к интерпретатору bash в приведённом выше сценарии на следующий: “#!/usr/bin/rm”, после чего запустите скрипт. Что произойдёт? Объясните результат

## Решение

Теперь интерпретатор - rm, который при запуске удалит данный файл.



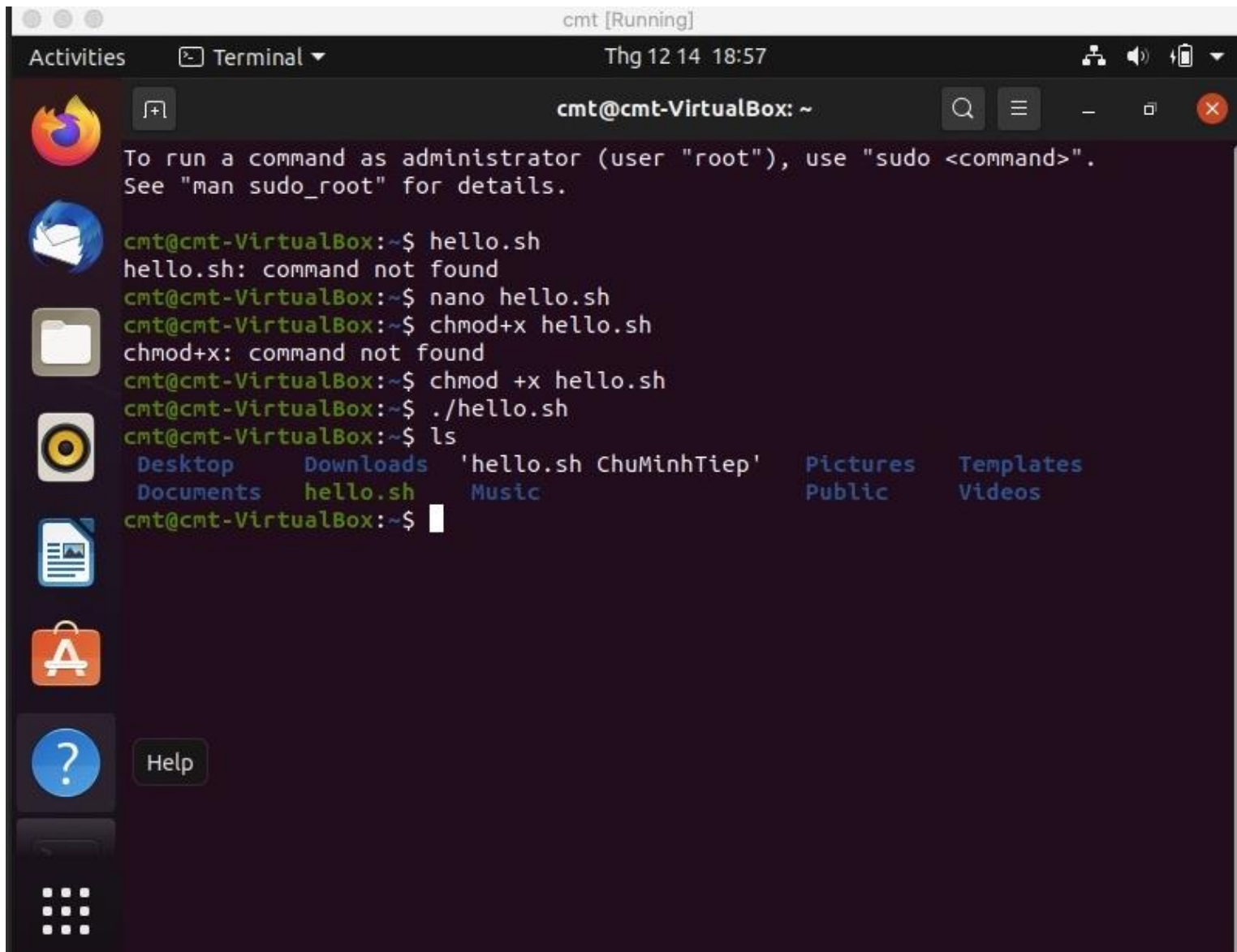
```

..          (parent dir)  .cache          (dir)  .config          (dir)
.gnupg      (dir)        .local          (dir)  .ssh            (dir)
Desktop     (dir)        Documents       (dir)  Downloads       (dir)
Music       (dir)        Pictures        (dir)  Public          (dir)
Templates   (dir)        Videos         (dir)  .bash_history    13 B
.bash_logout 220 B      .bashrc         3 KB  .hello.sh.swp    1 KB
.profile     807 B      hello.sh        0 B

```



^B Back  
^F Forward



The screenshot shows a terminal window titled 'cmt [Running]' with a dark background. The prompt is 'cmt@cmt-VirtualBox: ~'. A message at the top says: 'To run a command as administrator (user "root"), use "sudo <command>". See "man sudo\_root" for details.' The user enters the following commands and receives the following output:

```
cmt@cmt-VirtualBox:~$ hello.sh
hello.sh: command not found
cmt@cmt-VirtualBox:~$ nano hello.sh
cmt@cmt-VirtualBox:~$ chmod+x hello.sh
chmod+x: command not found
cmt@cmt-VirtualBox:~$ chmod +x hello.sh
cmt@cmt-VirtualBox:~$ ./hello.sh
cmt@cmt-VirtualBox:~$ ls
Desktop      Downloads  'hello.sh ChuMinhTiep'  Pictures  Templates
Documents    hello.sh   Music                  Public    Videos
cmt@cmt-VirtualBox:~$
```

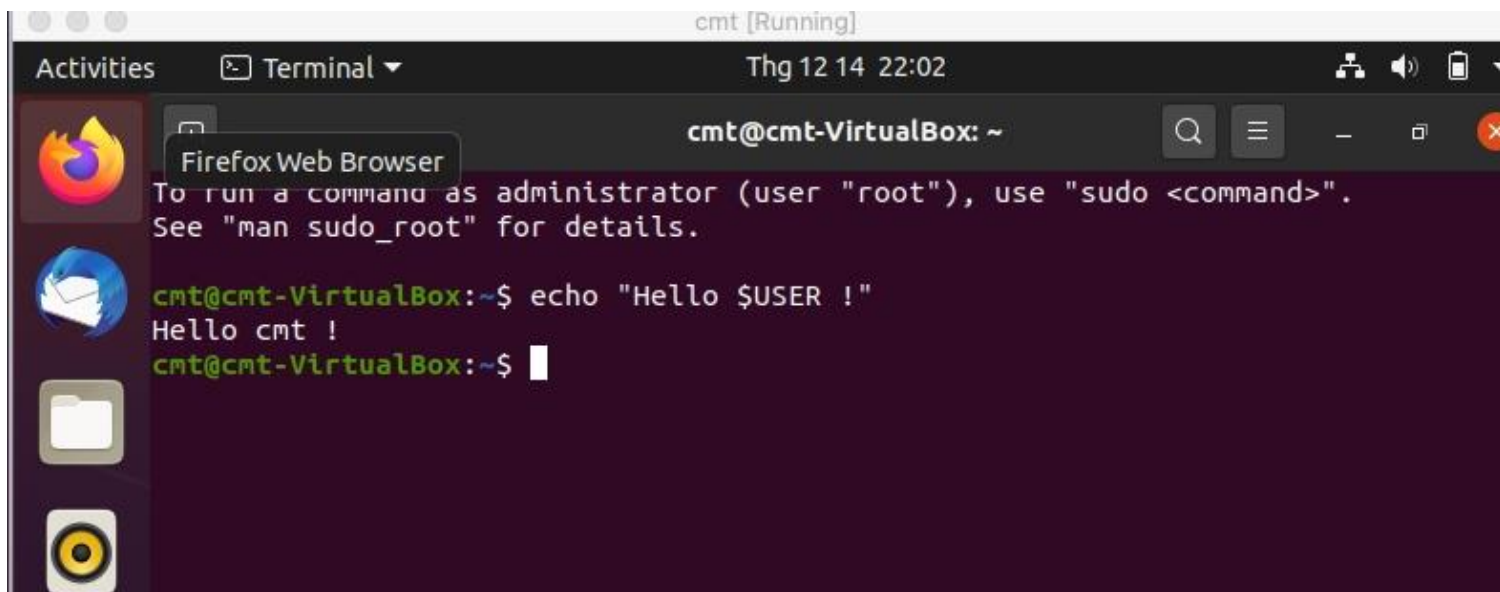
On the left side of the terminal, there is a vertical dock with icons for Firefox, a mail client, a file manager, a music player, a document viewer, an application store, a help icon, and a grid icon. A 'Help' button is also visible next to the help icon.

2. Напишите сценарий “hello.sh”, который будет выводить пользователю приветствие в виде “Hello, Username”, где “Username” имя вашего пользователя. **Hint:** Для получения имени пользователя, используйте переменную окружения USER.

## Решение

```
#!/usr/bin/bash

echo "Hello, "$USER
```



## Задание 2

Напишите сценарий "backup.sh", который будет создавать копию всех файлов в текущем каталоге (добавляя к имени каждого создаваемого файла расширение bak). После выполнения сценария, добавьте в каталог несколько новых файлов и убедитесь, что после повторного запуска сценария копии будут созданы только для тех файлов, которые не имеют расширение bak.

## Решение

```
File Edit View Search Terminal Help
[dmitry@localhost lab1]$ nano backup.sh
[dmitry@localhost lab1]$ ls
backup.sh task1_1.sh task1_2.sh
[dmitry@localhost lab1]$ ./backup.sh
Backupping file backup.sh
Backupping file task1_1.sh
Backupping file task1_2.sh
[dmitry@localhost lab1]$ ls
backup.sh backup.sh.bak task1_1.sh task1_1.sh.bak task1_2.sh task1_2.sh.bak
[dmitry@localhost lab1]$
```

```
File Edit View Search Terminal Help
[dmitry@localhost lab1]$ ls
backup.sh backup.sh.bak f1.txt f2.txt task1_1.sh task1_1.sh.bak task1_2.sh task1_2.sh.bak
[dmitry@localhost lab1]$ ./backup.sh
Backupping file backup.sh
Skipping file backup.sh.bak
Backupping file f1.txt
Backupping file f2.txt
Backupping file task1_1.sh
Skipping file task1_1.sh.bak
Backupping file task1_2.sh
Skipping file task1_2.sh.bak
[dmitry@localhost lab1]$ ls
backup.sh f1.txt f2.txt task1_1.sh task1_2.sh
backup.sh.bak f1.txt.bak f2.txt.bak task1_1.sh.bak task1_2.sh.bak
[dmitry@localhost lab1]$
```

```
#!/bin/bash

for file in *
do
    if [ ${file: -4} != ".bak" ]
    then
        cp $file $file.bak
        echo "Making backup of $file"
    else
        echo "Skipping $file"
    fi
done
```

### Задание 3

Написать bash-скрипт с именем «trash.sh», который будет имитировать присутствие корзины в системе. Ваш скрипт должен принимать произвольное число аргументов, указывающих на файлы и директории, которые должны быть перемещены в корзину. Корзина должна создаваться в домашней директории пользователя, от имени которого запущен скрипт, с именем «.trash».

Должны быть реализованы следующие опции:

1. -h — вывод справки по работе со скриптом;
2. -e — очистить корзину;
3. -l — вывести содержимое корзины;
4. -r filename — восстановить из корзины файл с именем «filename» в текущую директорию (подумайте о возможности восстанавливать файл в указанную пользователем директорию).

Следует предусмотреть обработку ситуации, когда имя файла, перемещаемого в корзину, совпадает с именем файла, уже присутствующего в корзине, но при этом файл все равно должен быть перемещен, а файл, находившийся в корзине, не должен быть затерт/заменен.

### Решение

```

#!/bin/bash

if [ ! -d ~/.trash ]; then
    mkdir ~/.trash
fi

usage="
Usage: trash.sh [OPTIONS][FILE]\n
Here are the OPTIONS:\n
-h to display this help and exit\n
-e to empty trash\n
-l to list contents\n
-r FILENAME DIRECTORY: restore FILENAME to the specified DIRECTORY (optional)\n"

while getopts ":helr" opt; do
    case ${opt} in
        h )
            echo -e $usage;;
        e )
            for file in $(ls ~/.trash); do
                rm -r ~/.trash/$file
            done;;
        l )
            ls -p ~/.trash;;
        r )
            if [ -e ~/.trash/$2 ]; then
                if [ -z $3 ]; then
                    mv ~/.trash/$2 "$2"
                else
                    if [ -d $3 ]; then
                        mv ~/.trash/$2 "$3/$2"
                    else
                        mkdir -p $3
                        mv ~/.trash/$2 "$3/$2"
                    fi
                fi
            fi
            exit 0;;
        \? ) echo "Invalid option";;
    esac
done

extra_arguments=${@:OPTIND}
for file in $extra_arguments; do
    if [ -e "$file" ]; then
        if [ -e ~/.trash/$file ]; then
            mv $file ~/.trash/$(date '+%Y%j%H%M%S')_"$file"
        else
            mv $file ~/.trash/$file
        fi
    else
        echo "$file not found"
    fi
done
exit 0

```



```

[dmity@localhost lab1]$ ls
backup.sh task1_1.sh task1_2.sh trash.sh
[dmity@localhost lab1]$ touch (1..5).txt
[dmity@localhost lab1]$ ls
1.txt 2.txt 3.txt 4.txt 5.txt backup.sh task1_1.sh task1_2.sh trash.sh
[dmity@localhost lab1]$ ./trash.sh *.txt
[dmity@localhost lab1]$ ls
backup.sh task1_1.sh task1_2.sh trash.sh
[dmity@localhost lab1]$ ./trash.sh -l
1.txt 2.txt 3.txt 4.txt 5.txt
[dmity@localhost lab1]$ touch 1.txt
[dmity@localhost lab1]$ ls
1.txt backup.sh task1_1.sh task1_2.sh trash.sh
[dmity@localhost lab1]$ ./trash.sh 1.txt
[dmity@localhost lab1]$ ./trash.sh -l
1.txt 2020260171153 1.txt 2.txt 3.txt 4.txt 5.txt
[dmity@localhost lab1]$ ./trash.sh -r 1.txt
[dmity@localhost lab1]$ ls
1.txt backup.sh task1_1.sh task1_2.sh trash.sh
[dmity@localhost lab1]$ ./trash.sh -r 2.txt restored
[dmity@localhost lab1]$ ls
1.txt backup.sh restored task1_1.sh task1_2.sh trash.sh
[dmity@localhost lab1]$ ls -R
.:
1.txt backup.sh restored task1_1.sh task1_2.sh trash.sh
./restored:
2.txt
[dmity@localhost lab1]$ ./trash.sh -e
[dmity@localhost lab1]$ ./trash.sh -l
[dmity@localhost lab1]$

```

## Ответы на вопросы из презентации

### 1. RPM vs DPKG

Это разные типы архивов. RPM используется в RedHat и CentOS, а DPKG в Debian дистрибутивах. Команда для RPM yum, для DPKG apt-get.

### 2. Объяснить смысл и назначение GPL v2, 3

В общем случае GPL - это copyleft лицензия, которая предполагает, что если код под лицензией GPL был модифицирован или встроен в другую программу, то эта программа также должна распространяться под GPL. GPLv3 отличается от GPLv2 более строгими правилами. Например, патентодержатель не имеет права распространять программы под GPLv3 на эксклюзивных условиях для патентов. Действует принцип "все или никому". Кроме того, GPLv3 разрешает добавить условия, согласно которым патентодержатель лишается прав на распространение программы в случае подания судебного иска против других пользователей программы по поводу нарушения своих патентов. Также, в GPLv3 новая независимая от US юридическая лексика, плюс добавлена совместимость с лицензией Apache. Ещё, в GPLv3 определена чёткая процедура добавления дополнительных разрешений и ограничений, это позволило переписать LGPLv3 как GPLv3 с дополнительными разрешениями.

#### 3.1 Аналогом каких команд являются команды "ll" и "dir"?

```
ll = ls -l dir = ls
```

#### 2. Что выводит команда "ls -thor"?

t - sort by modification time, newest first h - human readable file sizes o - like l, but without group information r - reversed sort

Получается, что эта команда выводит подробный список файлов без указания группы доступа, при этом первыми отображаются самые старые файлы, а размеры файлов и директорий указываются в удобочитаемом формате.



### 3.2 Что делает команда “!!”? А команда “!!:p”?

!! - вызывает последнюю команду !!:p - показывает, какая была последняя команда, но не вызывает ее

### 4. Что произойдет после выполнения команды “cd .” (зам.: одна точка)?

Из того, что удалось загуглить: "update the inode number of this directory"

Если по простому, то мы просто переходим в текущую директорию, то есть никуда. Но на внутреннем уровне обновляется какой-то "inode number", что может быть полезно, например, если директория была удалена, а потом создана с тем же именем.

### 5. Попробуйте выполнить “cd <Tab><Tab>” или “cd /h<Tab><Tab>”. Объясните результат?

По нажатию таба происходит автоподстановка возможных имен файлов или директорий, что дает возможность не прописывать все названия полностью. При первом нажатии произойдет попытка точной идентификации нужного файла/директории, а при втором — отображение всех возможных файлов/директорий, которые подходят под уже напечатанные символы.

**Заметка : Во время процесса вы видите 2 разных сервера, потому что мой компьютер недостаточно мощный, чтобы переключиться на Linux, поэтому я сделал это сам на другом сервере.!!!**

**Большое спасибо за просмотр !!!**

