

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №1

по курсу «ОС Linux»

Вариант 5

Студент

Морозов Д.С.

Группа ПИ-21-1

Руководитель

Кургасов В.В.

Липецк 2023 г.

Задание кафедры

- Установить гипервизор, изучить понятия: Виртуализация, гипервизоры, изучить создание виртуальных машин.
- Установить 64 серверную ОС на выбор (Ubuntu, Debian)
- Дано: одномерный массив из N целочисленных элементов (тип `int`). Все отрицательные элементы массива заменить их модулем, подсчитать их количество.

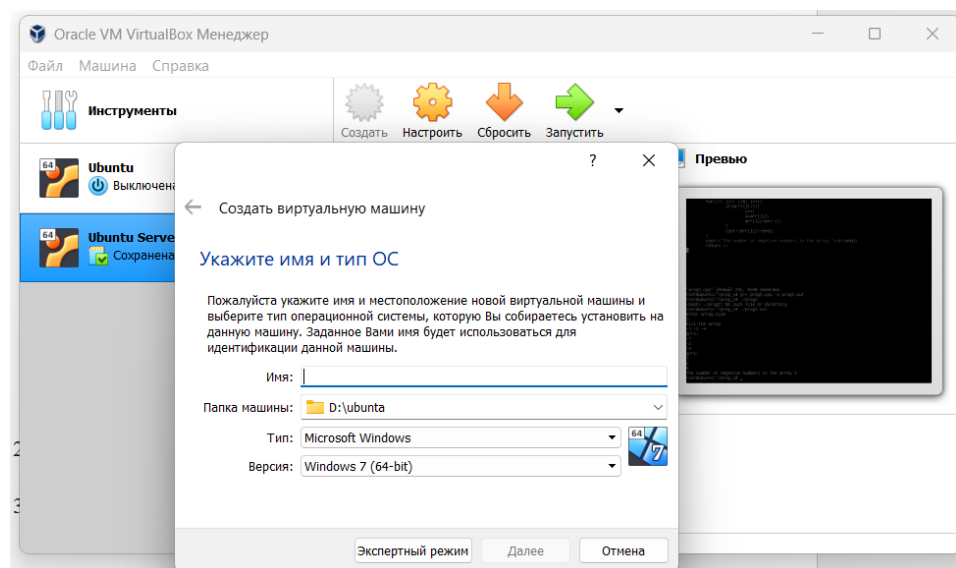
Ход работы

1) Установка гипервизора VirtualBox:

- Скачиваем VirtualBox с официального сайта <https://www.virtualbox.org/>
- После чего совершаем установку

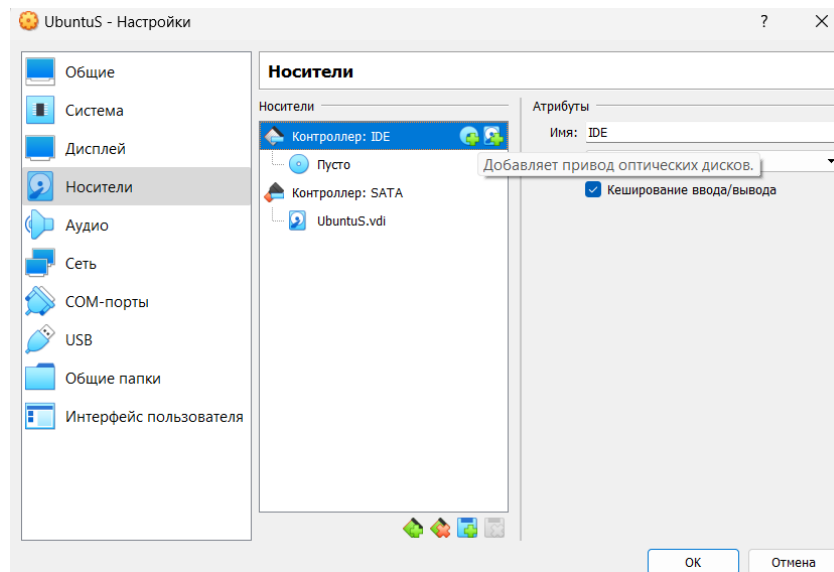


- 2) После чего нам необходимо скачать образ Ubuntu Server с официального сайта <https://ubuntu.com/>
- 3) Далее создаем виртуальную машину
 - Открываем VirtualBox и нажимаем кнопку создать

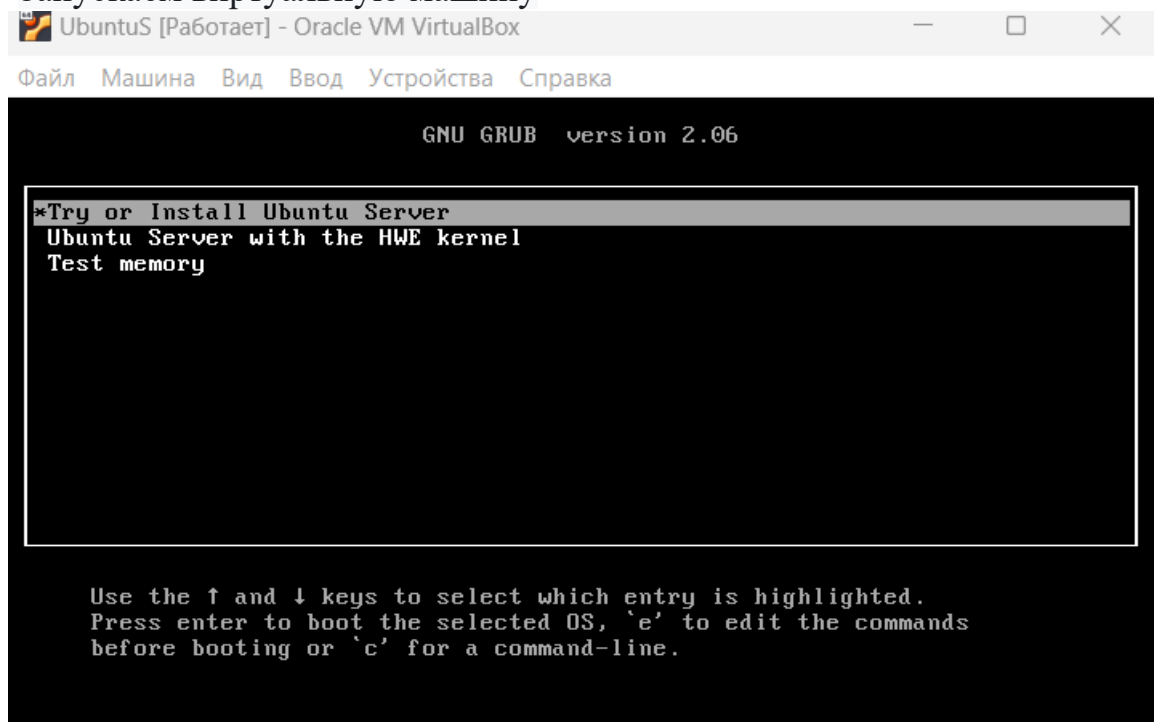


- Далее следуем инструкциям и выполняем создание

- После создания виртуальной машины, заходим в ее настройки, пункт Носители и добавляем привод оптических дисков, то есть ранее скаченный образ Ubuntu Server и жмем OK



4) Запускаем виртуальную машину



Жмем Enter для установки операционной системы

```
Файл  Машина  Вид  Ввод  Устройства  Справка
[ OK ] Started Wait until snapd is fully seeded.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
        Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.
Ubuntu 18.04.4 LTS ubuntu-server tty1
```

Идет установка, следуем шагам до завершения установки.

5) Установка завершена

```
Ubuntu 22.04 LTS ubuntu tty1
ubuntu login: dikamen
Password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-47-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

6) Понятия Виртуализация и Гипервизоры:

- Виртуализация — предоставление набора вычислительных ресурсов или их логического объединения, абстрагированное от аппаратной реализации, и обеспечивающее при этом логическую изоляцию друг от друга вычислительных процессов, выполняемых на одном физическом ресурсе.

- Гипервизор или монитор виртуальных машин — программа или аппаратная схема, обеспечивающая или позволяющая одновременное, параллельное выполнение нескольких операционных систем на одном и том же хост-компьютере.

7) Для установки gcc, g++, make вводим команду

```
daniil@ubuntu:~$ sudo apt install build-essential
```

Теперь мы можем компилировать программы на Си/C++

8) После чего создаем файл с разрешением cpp в редакторе nano с помощью команды:

```
nano prog1.cpp
```

И записываем код на C++:

```
GNU nano 6.2 prog1.cpp
#include <iostream>
using namespace std;
int main(){
    int N;
    cout<<"Enter array size"<<endl;
    cin>>N;
    int *arr1=new int[N];
    cout<<"Fill the array"<<endl;
    for(int i=0; i<N; i++){
        cin>>arr1[i];
    }
    cout<<"arr1:"<<endl;
    for(int i=0; i<N; i++){
        cout<<arr1[i]<<endl;
    }
    int a;
    int s=0;
    cout<<"arr1:"<<endl;
    for(int i=0; i<N; i++){
        if(arr1[i]<0){
            s++;
            a=arr1[i];
            arr1[i]=a*(-1);
        }
        cout<<arr1[i]<<endl;
    }
    cout<<"The number of negative numbers in the array "<<s<<endl;
    return 0;
}
```

[Wrote 29 lines]

Help Write Out Where Is Cut Execute Location M-U Undo
Exit Read File Replace Paste Justify Go To Line M-E Redo

Сохраняем содержимое файла Ctrl+S и выходим из него Ctrl+X

9) Компилируем программу с помощью команды:

```
root@ubuntu:~/prog_c# gcc prog1.cpp -o prog1
```

10) После чего ее запускаем командой ./prog1:

```

root@ubuntu:~/prog_c# ./prog1
Enter array size
5
Fill the array
-1 7 -3 -4 2
arr1:
-1
7
-3
-4
2
arr1:
1
7
3
4
2
The number of negative numbers in the array 3
root@ubuntu:~/prog_c#

```

11) Теперь запустим ее с помощью команды make:

- Создаем командой makefile

```

root@ubuntu:~/prog_c# touch makefile_

```

- Затем открываем его в редакторе nano и записываем следующие команды:

```

Файл  Машина  Вид  Ввод  Устройс
GNU nano 6.2
all:
    g++ -o prog1 prog1.cpp
run: all
    ./prog1

```

Теперь наша программа в файле prog1.cpp будет сама компилироваться и запускаться после команды make run:

```

root@ubuntu:~/prog_c# make run
g++ -o prog1 prog1.cpp
./prog1
Enter array size
7
Fill the array
5 1 2 -8 -9 1 2
arr1:
5
1
2
-8
-9
1
2
arr1:
5
1
2
-8
-9
1
2
The number of negative numbers in the array 2
root@ubuntu:~/prog_c#

```

12) Скачаем GDB для того, чтобы запустить нашу программу там:

- Для скачивания вводим команду:

```

root@ubuntu:~/prog_c# sudo apt install gdb

```

- После установки запускаем с помощью gdb скомпилированный файл:

```

(gdb) # gdb ./prog1

```

- С помощью команды **b** имя можно поставить точку останова, а командой **r** запускаем программу:

```

(gdb) b main
Breakpoint 1 at 0x1251
(gdb) r
Starting program: /root/prog_c/prog1
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, 0x0000555555555251 in main ()
(gdb)

```


- Выполнение программы остановится на ранее поставленной точки остановки, для продолжения выполнения введем команду **c**:

```
breakpoint 1, 0x0000555555555251 in main ()
(gdb) c
Continuing.
Enter array size
9
Fill the array
7 8 9 -6 -6 1 2 3 -1
arr1:
7
8
9
-6
-6
1
2
3
-1
arr1:
7
8
9
6
6
1
2
3
1
The number of negative numbers in the array 3
[Inferior 1 (process 3946) exited normally]
(gdb)
```

- Командой **q** выходим из GDB:

```
(gdb) q
root@ubuntu:~/prog_c#
```

13) Теперь установим редактор **vim**, создадим с помощью него файл с программой и запустим ее:

- Установка:

```
root@ubuntu:~/prog_c# sudo apt install vim_
```

- Создание файла:

```
root@ubuntu:~/prog_c# vim prog2.cpp
```

- Ввод программы:

```

Файл  Машина  Вид  Ввод  Устройства  Справка
#include <iostream>
using namespace std;
int main()
{
    int N;
    cout<<"Enter array size"<<endl;
    cin>>N;
    int *arr1=new int[N];
    cout<<"Fill the array"<<endl;
    for(int i=0; i<N; i++){
        cin>>arr1[i];
    }
    cout<<"arr1:"<<endl;
    for(int i=0; i<N; i++){
        cout<<arr1[i]<<endl;
    }
    int a;
    int s=0;
    cout<<"arr1:"<<endl;
    for(int i=0; i<N; i++){
        if(arr1[i]<0){
            s++;
            a=arr1[i];
            arr1[i]=a*(-1);
        }
        cout<<arr1[i]<<endl;
    }
    cout<<"The number of negative numbers in the array "<<s<<endl;
    return 0;
}

```

- Сохраняем и выходим(жмем Esc, : и в командной строке вводим wq)
- Далее компилируем и запускаем файл:

Компиляция:

```
root@ubuntu:~/prog_c# g++ prog2.cpp -o prog2.out
```

Запуск:

```

root@ubuntu:~/prog_c# ./prog2.out
Enter array size
3
Fill the array
-7 -2 -4
arr1:
-7
-2
-4
arr1:
7
2
4
The number of negative numbers in the array 3
root@ubuntu:~/prog_c# _

```

Вывод:

Для работы в другой ОС нужно проделать несколько небольших действий, для этого не нужно менять свою основную ОС. Так же терминал Ubuntu Server обладает большими возможностями, на нем можно очень легко запускать программы несколькими способами.

Контрольные вопросы:

1. Что такое IDE?

Интегрированная среда разработки, ИСР, также единая среда разработки, ЕСР — комплекс программных средств, используемый программистами для разработки программного обеспечения.

2. Что такое API?

API — описание способов взаимодействия одной компьютерной программы с другими. Обычно входит в описание какого-либо интернет-протокола, программного каркаса или стандарта вызовов функций операционной системы. Часто реализуется отдельной программной библиотекой или сервисом операционной системы.

3. Что такое библиотека в программировании?

Библиотека в программировании — сборник подпрограмм или объектов, используемых для разработки программного обеспечения. С точки зрения операционной системы и прикладного ПО, библиотеки разделяются на динамические и статические.

4. Понятия Статической и Динамической библиотек.

Динамическая библиотека — файл, содержащий машинный код. Загружается в память процесса загрузчиком программ операционной системы либо при создании процесса, либо по запросу уже работающего процесса, то есть динамически.

Статическая библиотека — объектный файл в виде файла (нередко может быть поставлен вместе с исходным кодом), код из которого выборочно или полностью вставляется в программу на этапе компоновки.

5. Что такое плагин?

Плагин — независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей. Плагины обычно выполняются в виде библиотек общего пользования.

6. Назовите несколько консольных текстовых редакторов для Linux.

Nano — консольный текстовый редактор для UNIX и Unix-подобных операционных систем, основанный на библиотеке curses и распространяемый под лицензией GNU GPL.

Vim — свободный текстовый редактор, созданный на основе более старого vi. Ныне это мощный текстовый редактор с полной свободой настройки и автоматизации, возможными благодаря расширениям и надстройкам.

Emacs — семейство многофункциональных расширяемых текстовых редакторов. Оригинальный EMACS был написан в 1976 году Ричардом Столлманом и Гаем Стиллом как набор макросов для редактирования для редактора TECO.

Spacemacs — это среда конфигурации для GNU Emacs. Он может использовать все возможности GNU Emacs, включая как графический интерфейс пользователя, так и пользовательский интерфейс командной строки, и может выполняться в системе X Window и в терминале оболочки Unix.

Leafpad — простой, легковесный, быстрый GTK+ текстовый редактор для Unix-подобных систем с небольшим количеством зависимостей от внешних библиотек. Его достоинством является малое время запуска на большинстве современного оборудования.

7. Что делает команда gcc?

Команда gcc предназначена для компиляции с помощью компилятора GCC кода на языке C.

8. Что делает команда make?

Утилита, автоматизирующая процесс преобразования файлов из одной формы в другую. Чаще всего это компиляция исходного кода в объектные файлы и последующая компоновка в исполняемые файлы или библиотеки.

9. Что делает команда gdb?

Позволяет подробно изучить ход выполнения программы, исполняя ее пошагово и просматривая значения переменных.

10. Дайте определение заголовочного файла и файла реализации. Приведите пример.

В программировании заголовочный файл или подключаемый файл — файл, содержимое которого автоматически добавляется препроцессором в исходный текст в том месте, где располагается некоторая директива.

Пример: `#include <iostream>` - подключает стандартный библиотечный заголовочный файл для работы с потоками ввода-вывода.

11. Что означает единица трансляции? В чем особенность разработки программ из нескольких единиц трансляции?

В языках программирования единица трансляции — максимальный блок исходного текста, который физически можно оттранслировать (преобразовать во внутреннее машинное представление; в частности, откомпилировать). При компиляции большого количества исходных файлов компилятор, по факту, выполняет множество раз одну и ту же работу по разбору тяжеловесных заголовков.

12. Дайте краткую характеристику каждому этапу трансляции программ, написанных на Си.

На первом этапе происходит препроцессорная обработка текста.

На втором этапе создается промежуточный (объектный) файл.

На третьем этапе несколько объектных файлов komponуются в единый исполняемый файл, который может быть загружен в память компьютера и выполнен.