

# Информатика

## Занятие 4

Грозов Владимир Андреевич

[va\\_groz@mail.ru](mailto:va_groz@mail.ru)

# Задание 1

## Задача:

Напишите программу на языке SASM, которая получает 16 случайных чисел, и записывает числа с номерами 0,2,4,...,14 в ячейки @0x88,@0x98,...,@0xF8, а числа с номерами 1,3,5,...,15 - в ячейки @0x8F,@0x9F,...,@0xFF.

# Задание 1

Ответ:

```
R00 <- 0
R01 <- 8
LOOP:  R02 <- ?
       R03 <- ?
       @0x88 <- R02
       @0x8F <- R03
       R03 <- @7
       R03 <- R03 + R01
       R03 <- R03 + R01
       @7 <- R03
       R02 <- @9
       R02 <- R02 + R01
       R02 <- R02 + R01
       @9 <- R02
       R00 <- R00++
       RF <- R00 ~ R01
       RC <- @LOOP(S)
END: RC <- @END
```

# Лабораторная работа №2

## Задачи

- Использование поразрядных операций
- Написание программы на языке C
- Написание программы на Assembler
- Дизассемблирование программы, написанной на языке C, и сравнение с программой, написанной на Assembler

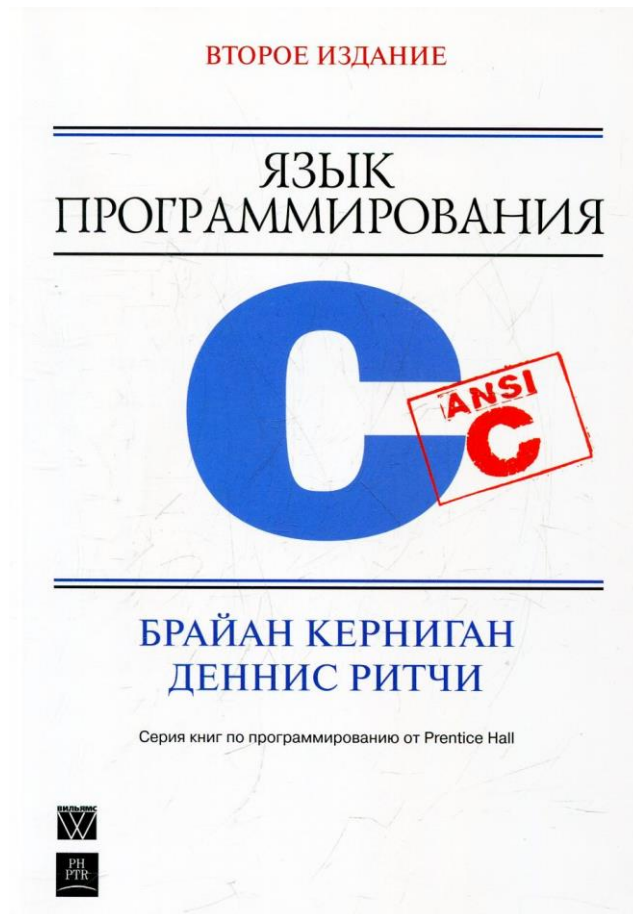
# Лабораторная работа №2

## Используемые средства

- C: компилятор gcc или clang
- Assembler: NASM
- ОС: Linux
- Дизассемблирование

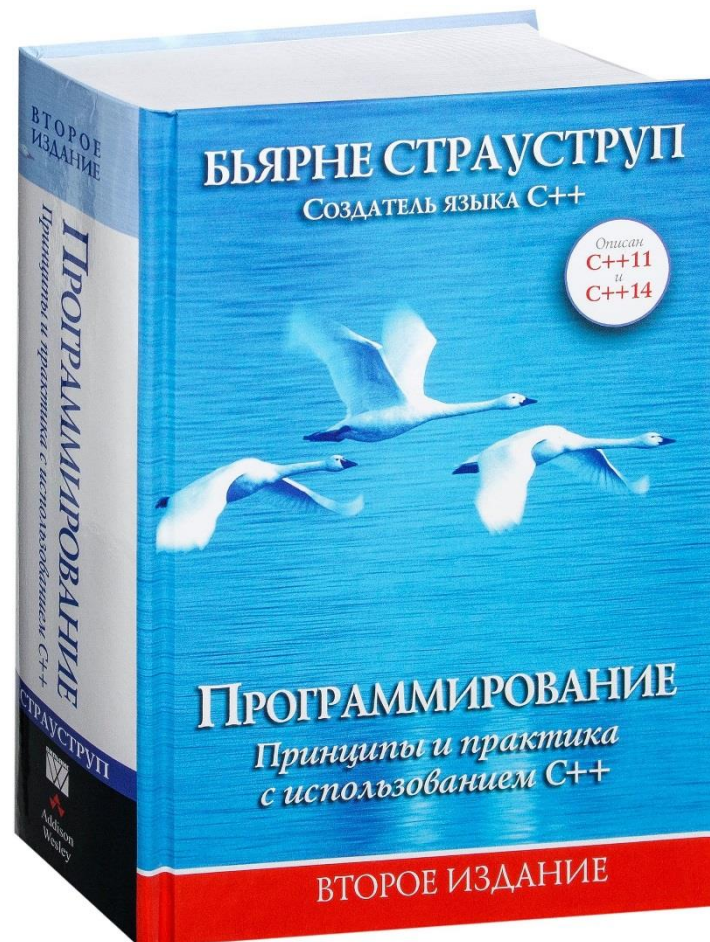
# Полезная литература по языку программирования С

1) Б. Керниган, Д. Ритчи. Язык программирования С.



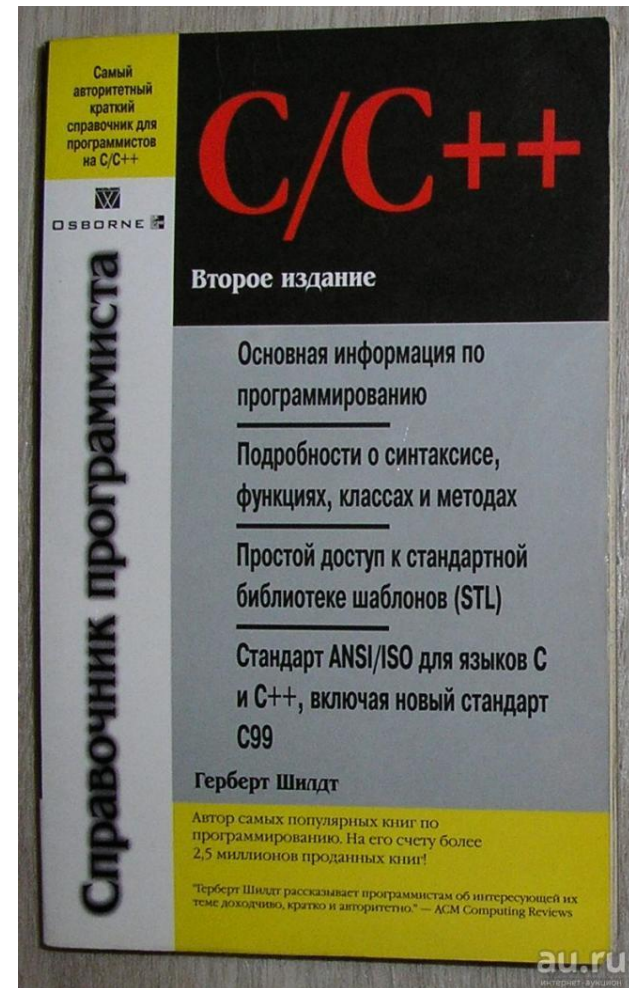
# Полезная литература по языку программирования С

## 2) Б. Страуструп. Язык программирования С++



# Полезная литература по языку программирования С

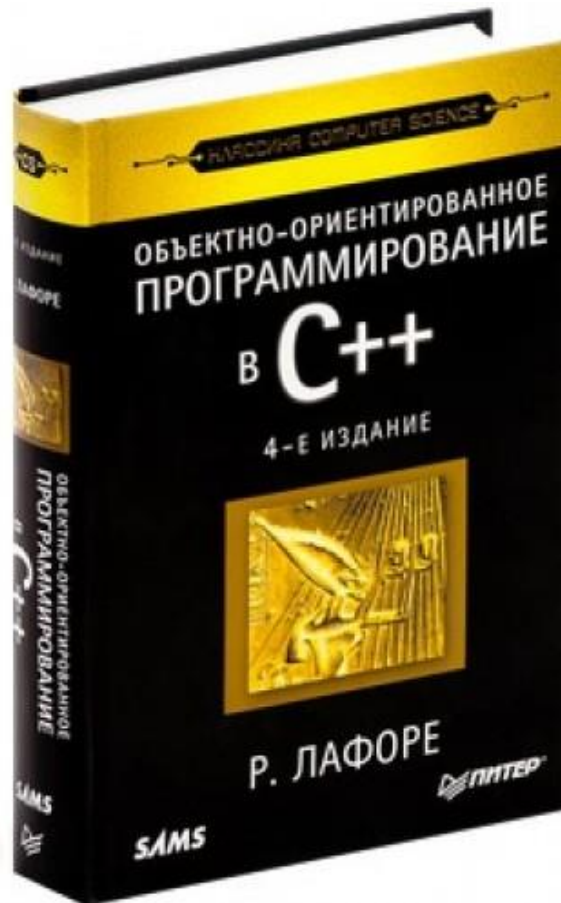
## 3) Герберт Шилдт





# Полезная литература по языку программирования С

4) Р. Лафоре. Объектно-ориентированное  
программирование в С++



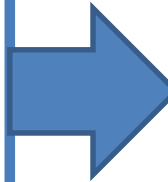
# Полезная литература по языку программирования С

Б. Керниган,  
Д. Ритчи

Б. Страуструп

Г. Шилдт

Р. Лафоре

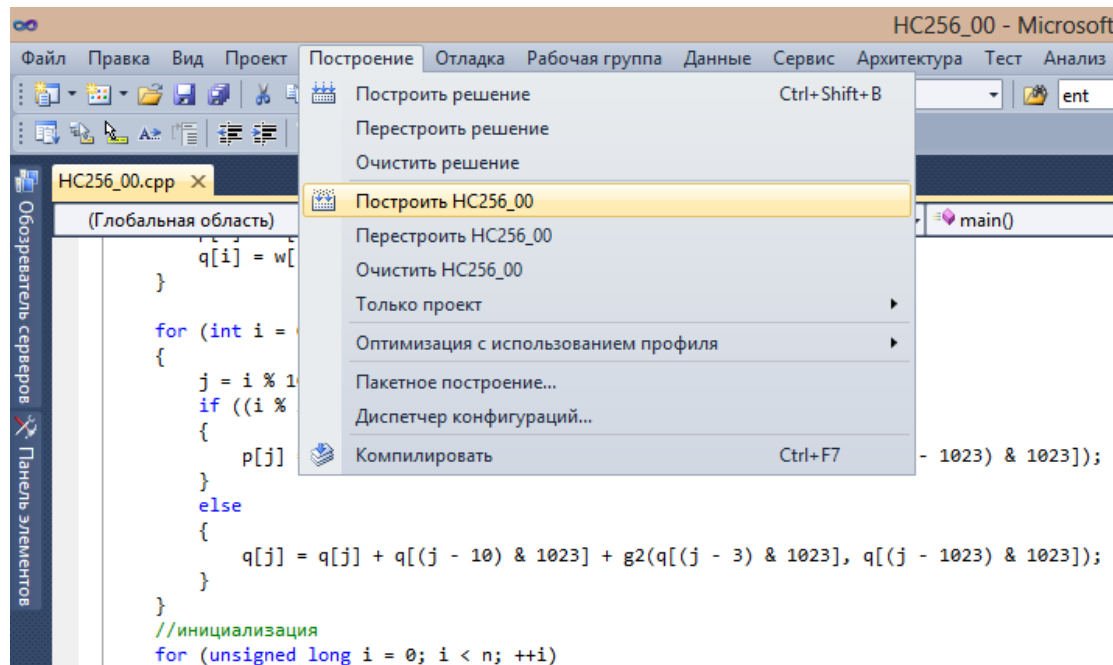


Г. Морган, Б. Робертс. Пиратский Кодекс

# С. Компиляция программ

## Среда разработки.

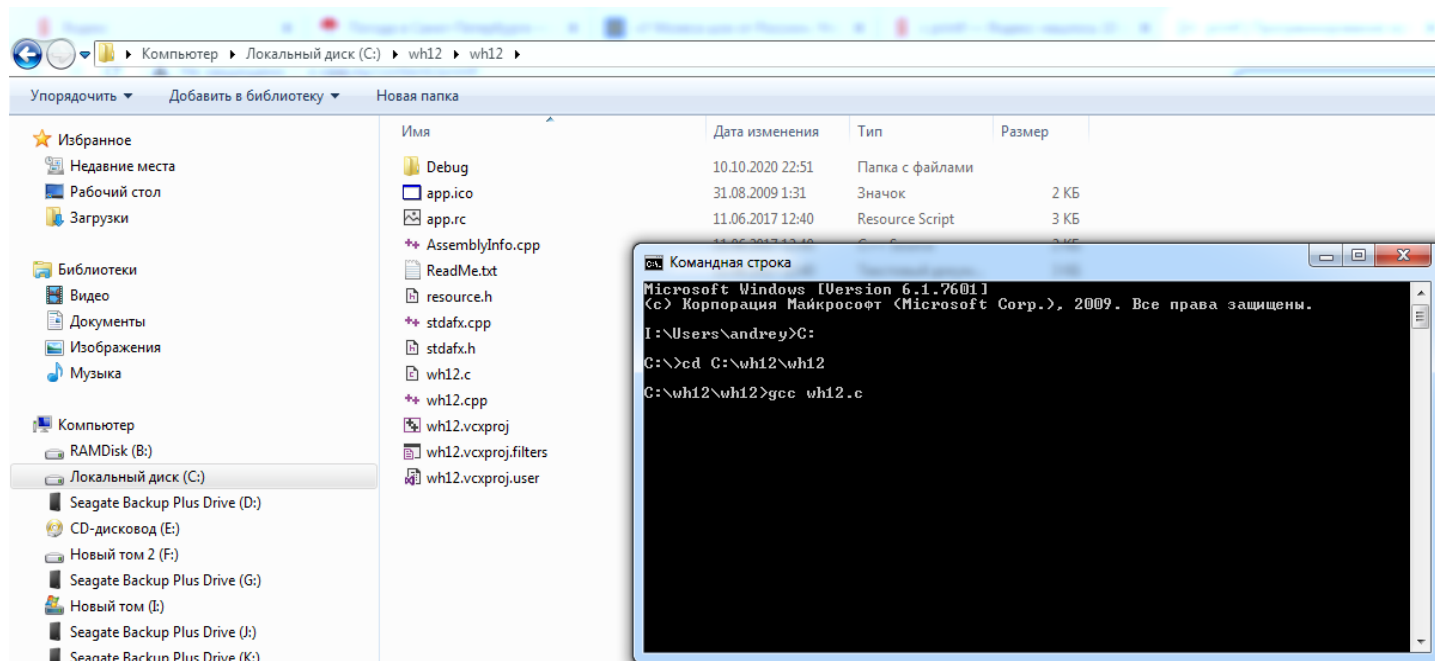
- Microsoft Visual Studio
  - Сохранить -> Построить -> Начать отладку/запуск без отладки



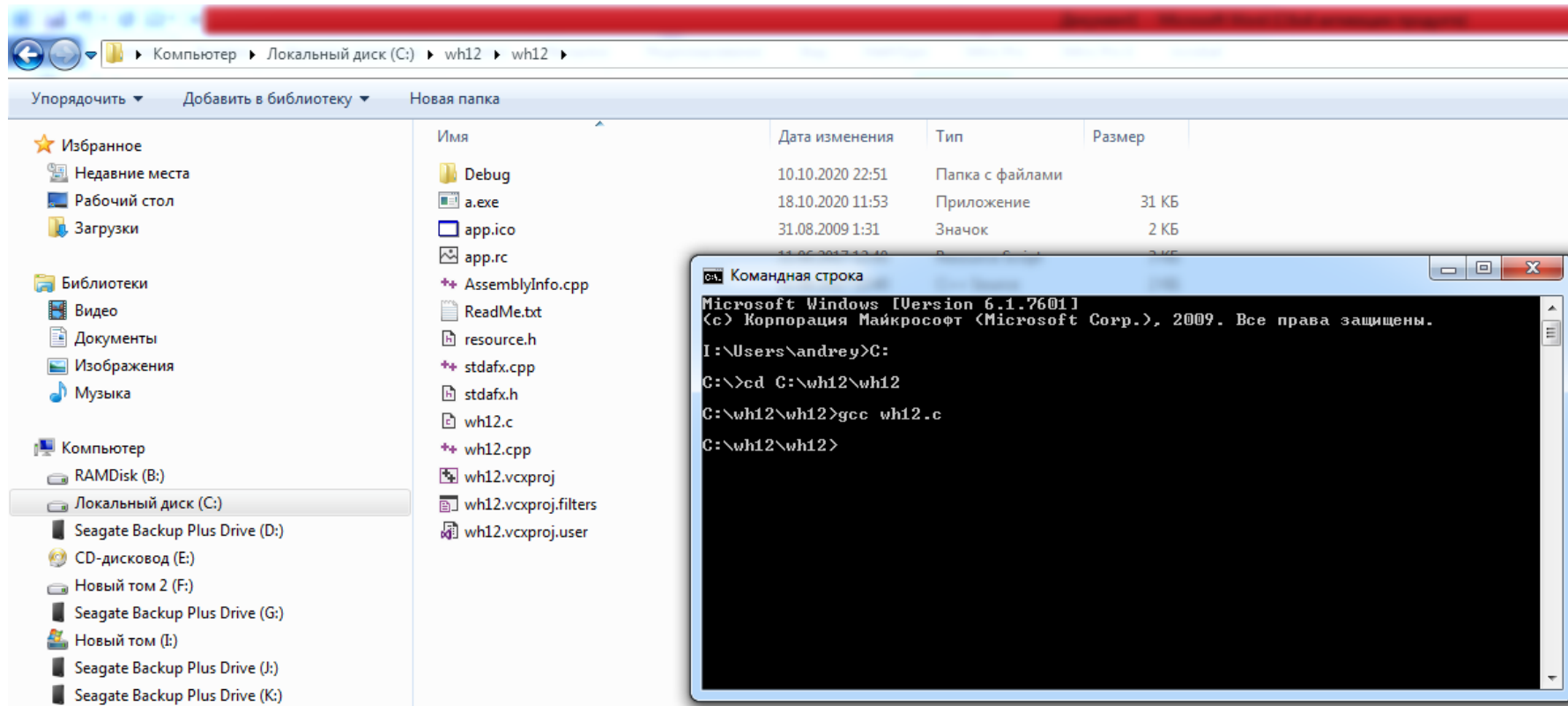
# С. Компиляция программ в Windows

## Использование командной строки

- Можно использовать любой компилятор
- Windows: необходимо самостоятельно установить нужный компилятор

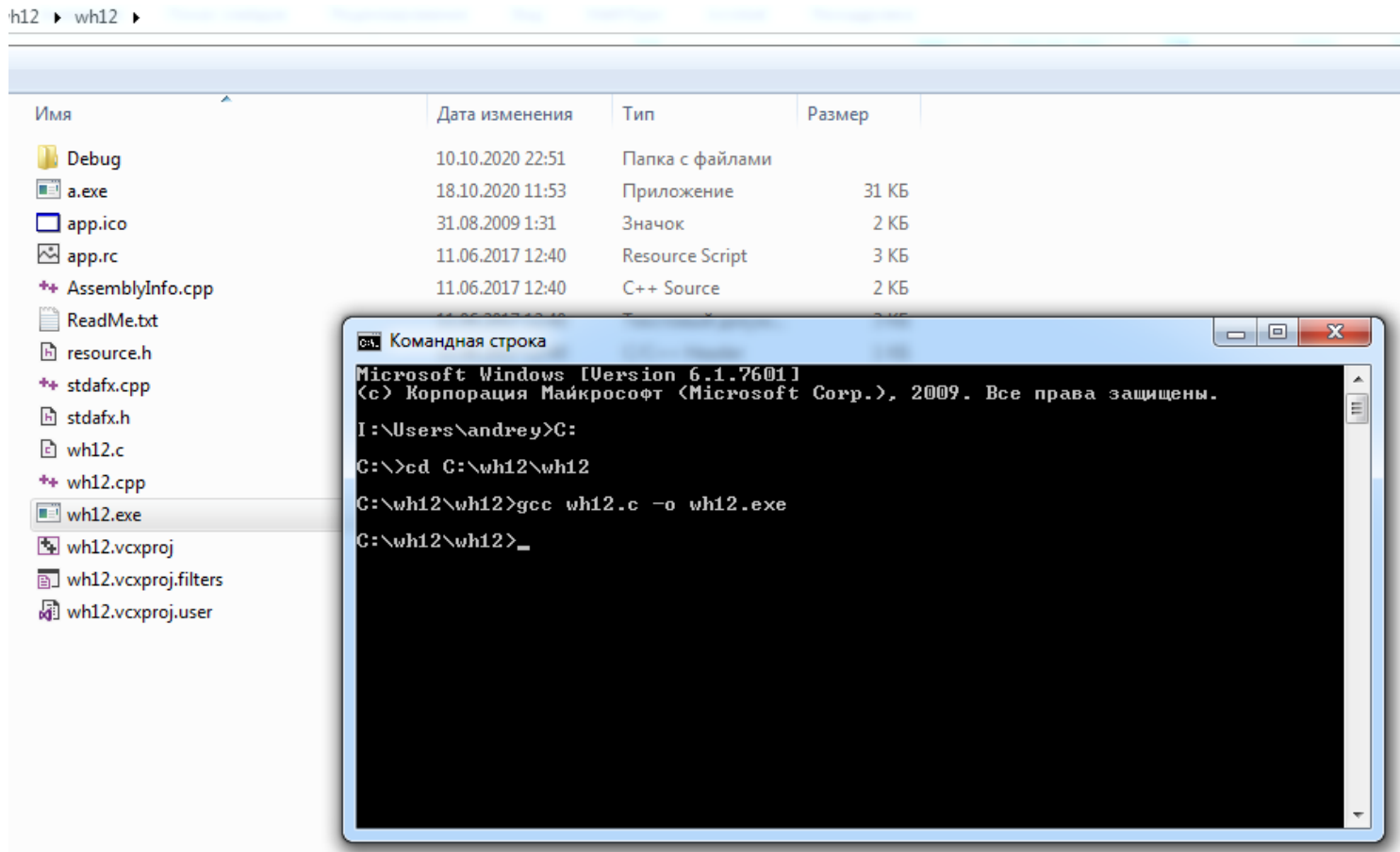


# С. Компиляция программ в ОС Windows

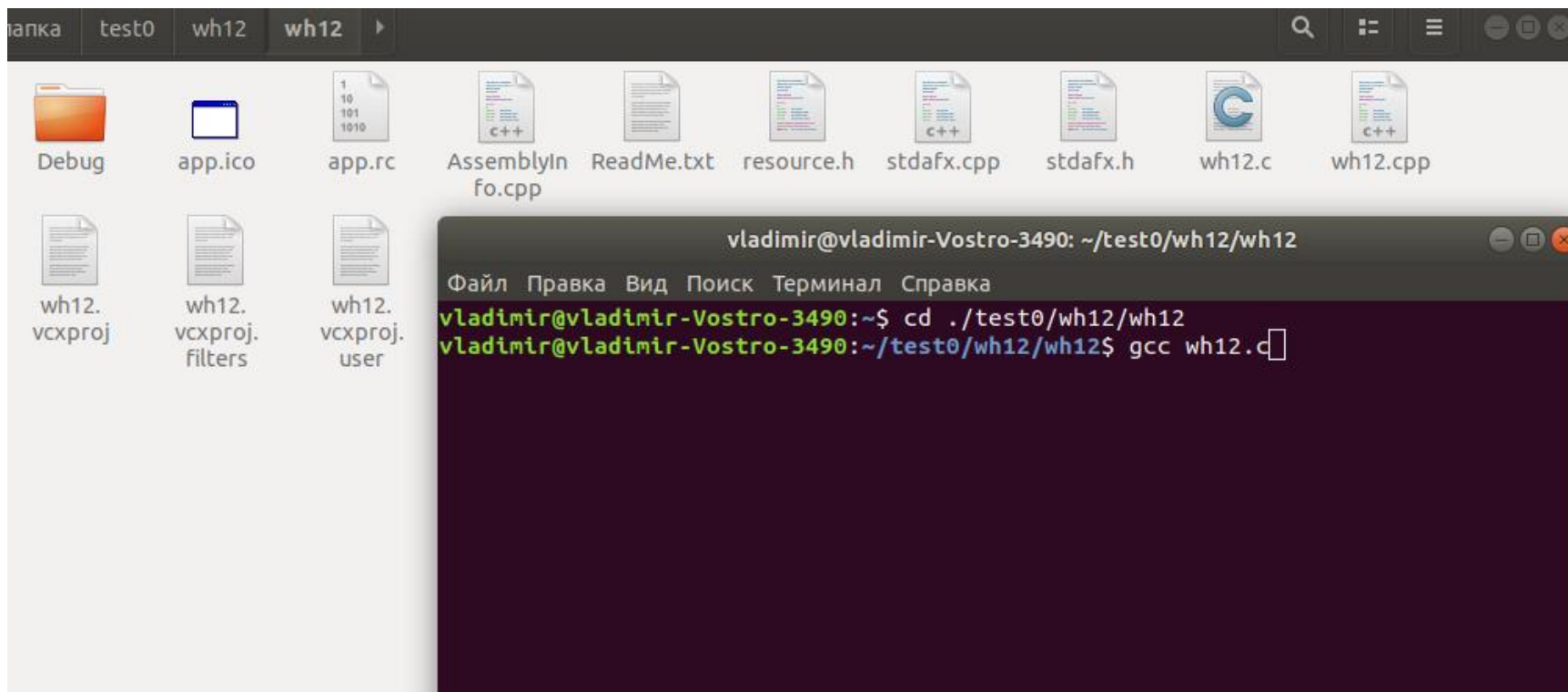


# С. Компиляция программ в ОС Windows

## Определение файла с конечной программой

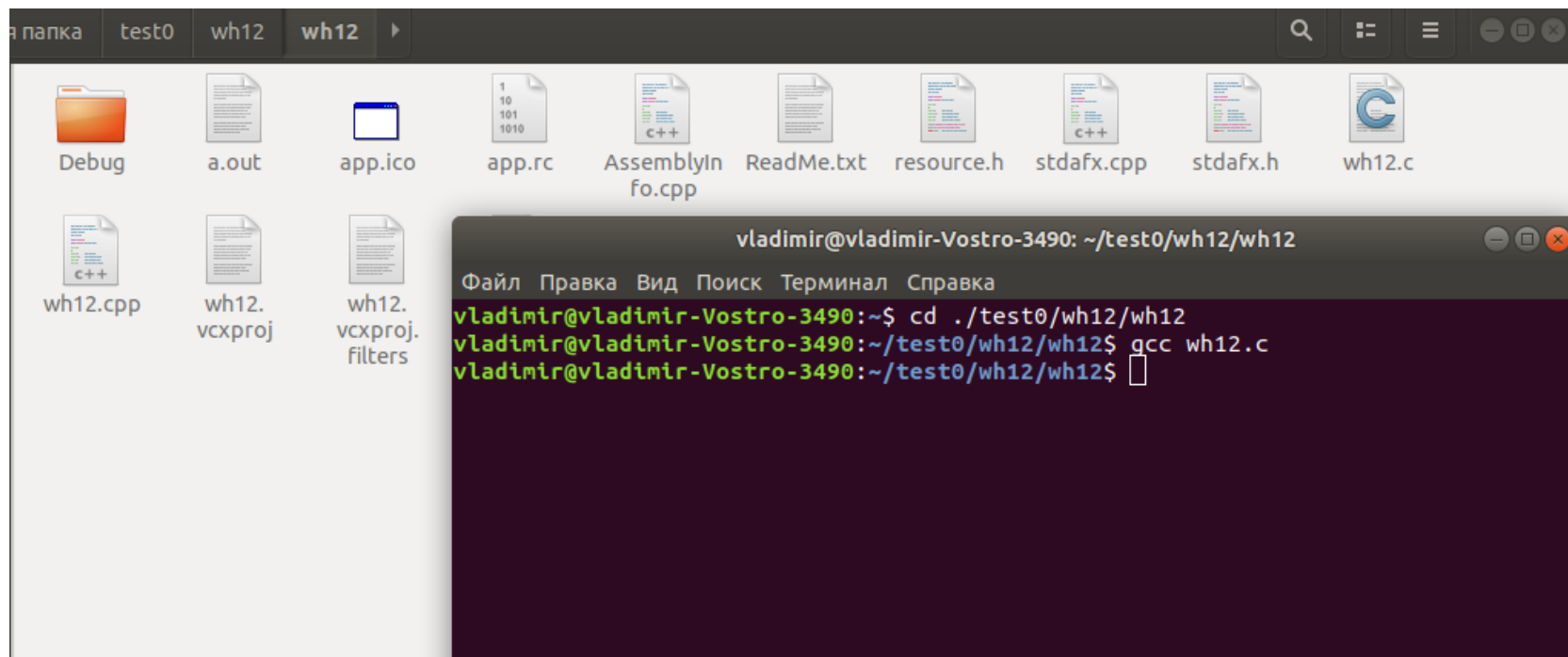


# С. Компиляция программ в ОС Linux



- Компилятор gcc установлен в ОС Linux по умолчанию.
- При работе с виртуальной машиной компилятор gcc необходимо устанавливать!

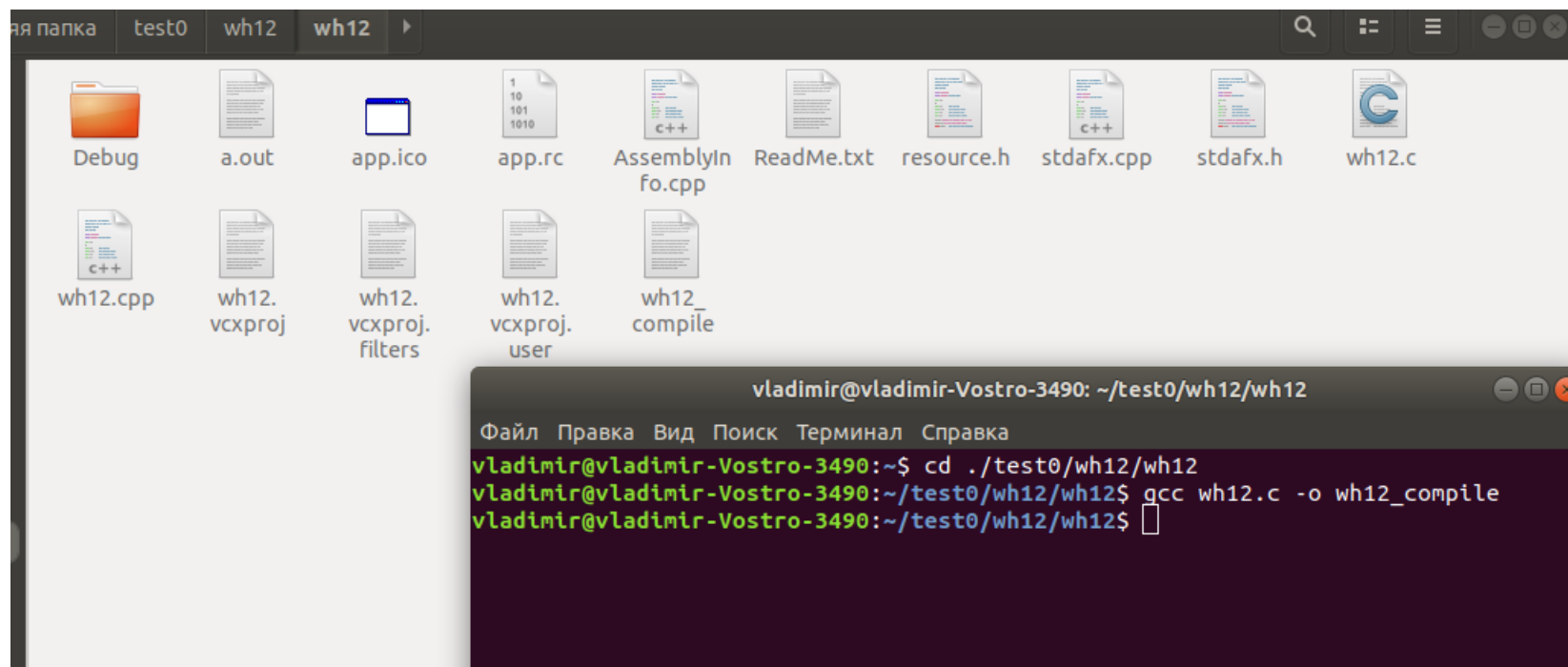
# С. Компиляция программ в ОС Linux





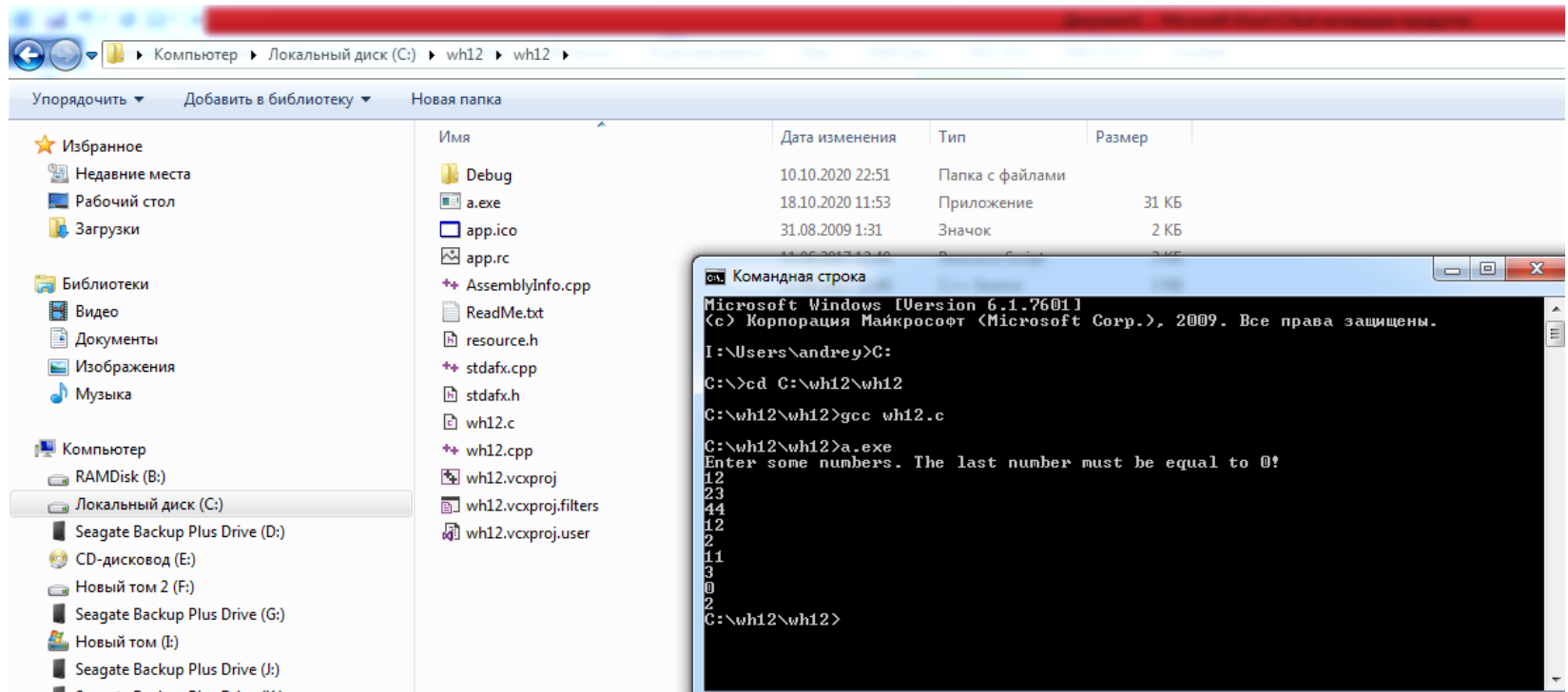
# С. Компиляция программ в ОС Linux

## Определение файла с конечной программой



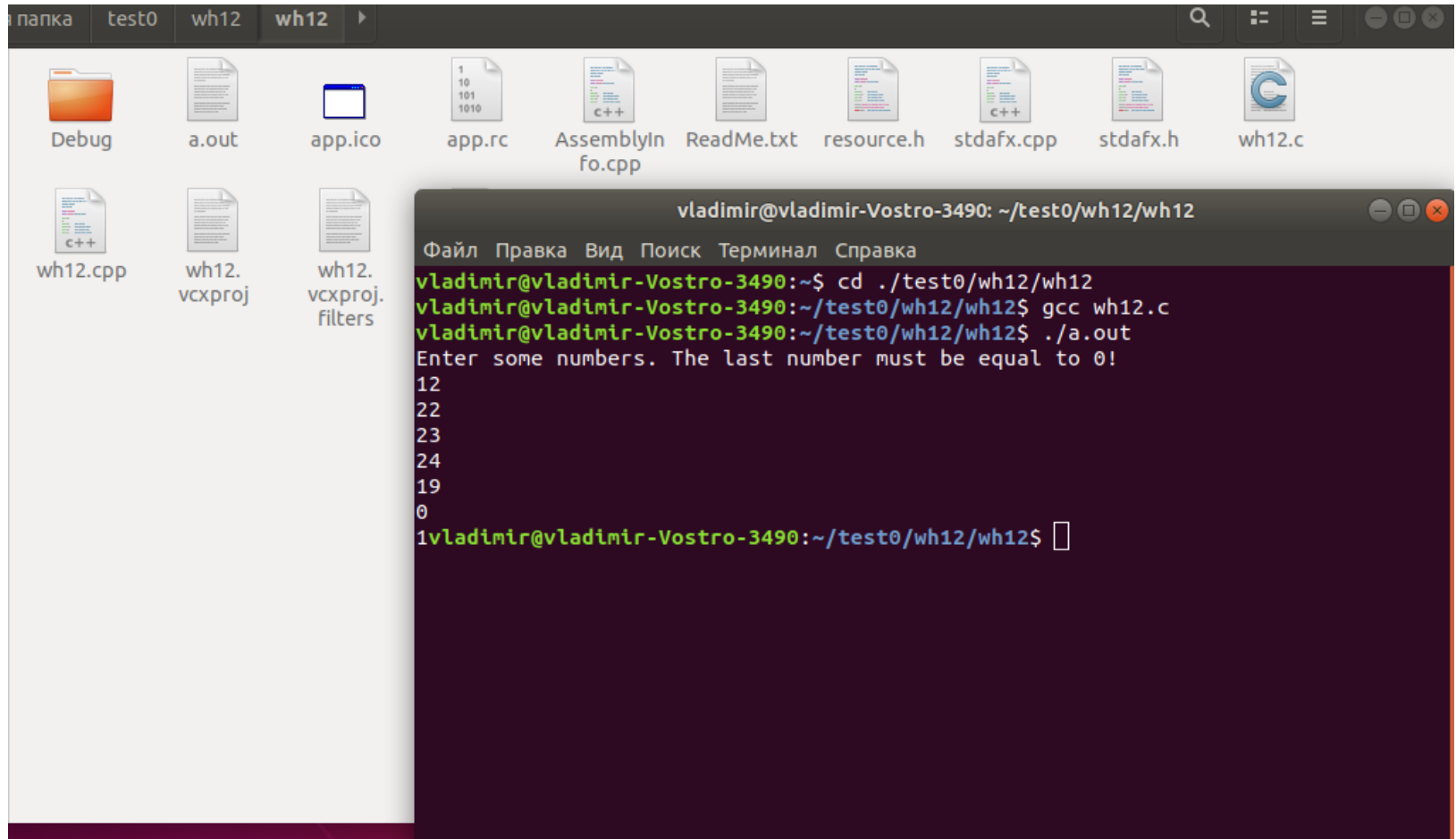
# С. Запуск скомпилированной программы

- ОС Windows



# С. Запуск скомпилированной программы

- ОС Linux



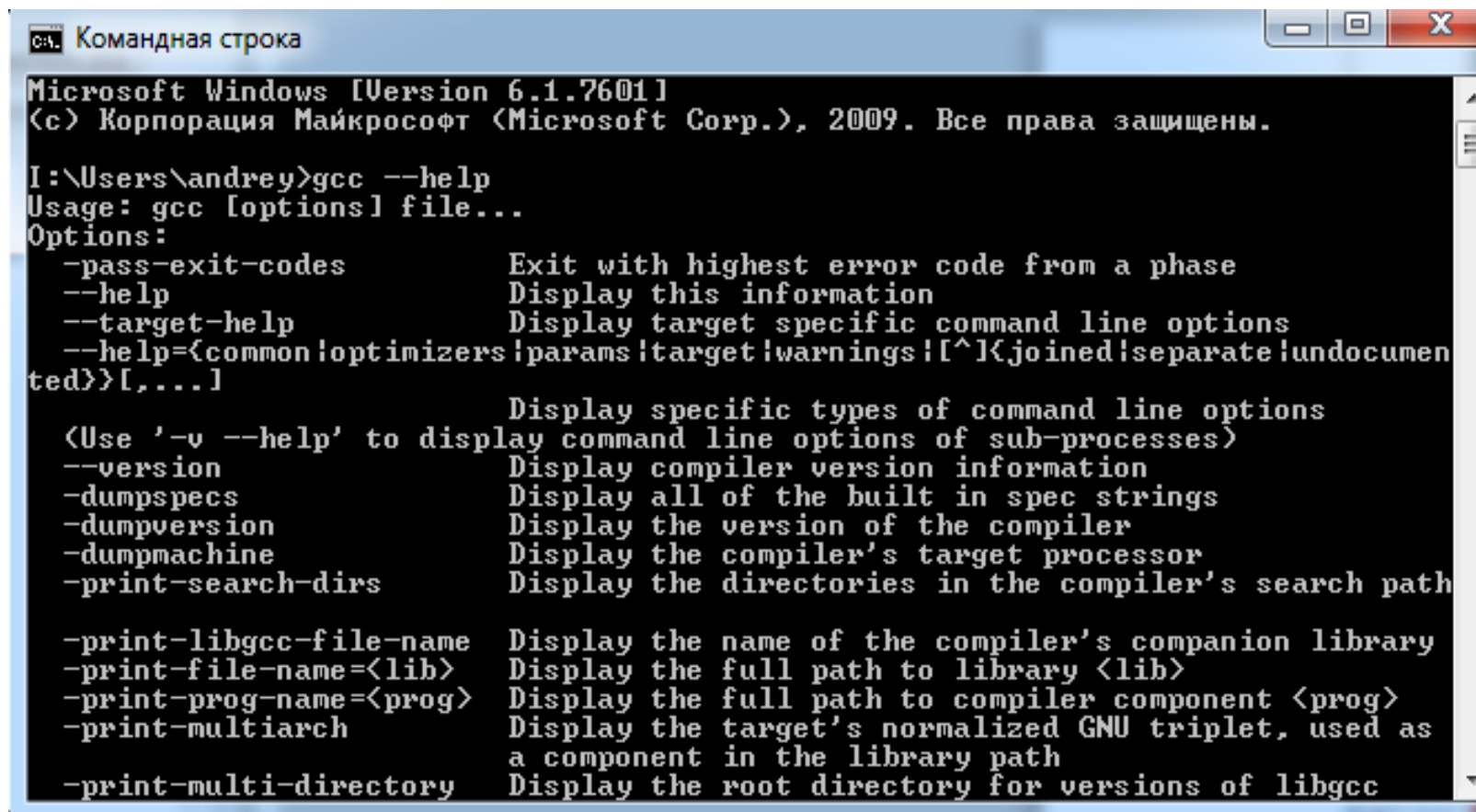
The screenshot shows a Linux desktop environment. In the background, a file manager window displays the contents of the directory `~/test0/wh12/wh12`. The files listed are: `Debug` (folder), `a.out` (executable), `app.ico` (image), `app.rc` (text), `AssemblyInFo.cpp` (C++ source), `ReadMe.txt` (text), `resource.h` (text), `stdafx.cpp` (C++ source), `stdafx.h` (C++ header), `wh12.c` (C source), `wh12.cpp` (C++ source), `wh12.vcxproj` (project file), and `wh12.filters` (text).

In the foreground, a terminal window titled `vladimir@vladimir-Vostro-3490: ~/test0/wh12/wh12` shows the following commands and output:

```
vladimir@vladimir-Vostro-3490:~$ cd ./test0/wh12/wh12
vladimir@vladimir-Vostro-3490:~/test0/wh12/wh12$ gcc wh12.c
vladimir@vladimir-Vostro-3490:~/test0/wh12/wh12$ ./a.out
Enter some numbers. The last number must be equal to 0!
12
22
23
24
19
0
1vladimir@vladimir-Vostro-3490:~/test0/wh12/wh12$
```

# С. Задание опций компиляции

- --help



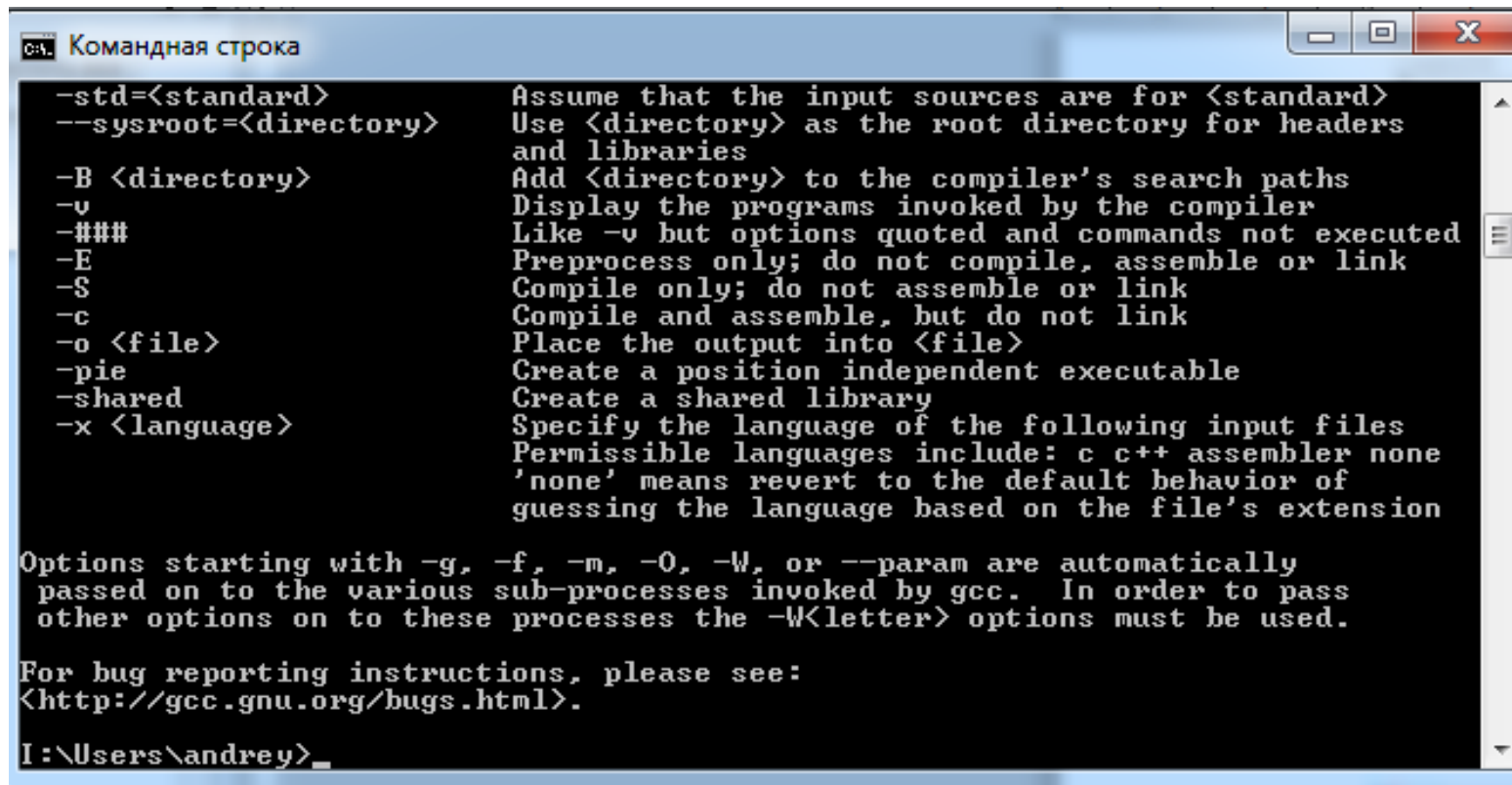
```
Командная строка
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

I:\Users\andrey>gcc --help
Usage: gcc [options] file...
Options:
  -pass-exit-codes      Exit with highest error code from a phase
  --help                Display this information
  --target-help         Display target specific command line options
  --help=<common!optimizers!params!target!warnings![^]<joined!separate!undocumen
ted>>[,...]
```

	Display specific types of command line options
(Use '-v --help' to display command line options of sub-processes)	
--version	Display compiler version information
-dumpspecs	Display all of the built in spec strings
-dumpversion	Display the version of the compiler
-dumpmachine	Display the compiler's target processor
-print-search-dirs	Display the directories in the compiler's search path
-print-libgcc-file-name	Display the name of the compiler's companion library
-print-file-name=<lib>	Display the full path to library <lib>
-print-prog-name=<prog>	Display the full path to compiler component <prog>
-print-multiarch	Display the target's normalized GNU triplet, used as a component in the library path
-print-multi-directory	Display the root directory for versions of libgcc

# С. Задание опций компиляции

- --help



```
C:\> Командная строка

-std=<standard>      Assume that the input sources are for <standard>
--sysroot=<directory> Use <directory> as the root directory for headers
                      and libraries
-B <directory>       Add <directory> to the compiler's search paths
-v                  Display the programs invoked by the compiler
-###                Like -v but options quoted and commands not executed
-E                  Preprocess only; do not compile, assemble or link
-S                  Compile only; do not assemble or link
-c                  Compile and assemble, but do not link
-o <file>            Place the output into <file>
-pie                Create a position independent executable
-shared             Create a shared library
-x <language>        Specify the language of the following input files
                      Permissible languages include: c c++ assembler none
                      'none' means revert to the default behavior of
                      guessing the language based on the file's extension

Options starting with -g, -f, -m, -O, -W, or --param are automatically
passed on to the various sub-processes invoked by gcc.  In order to pass
other options on to these processes the -W<letter> options must be used.

For bug reporting instructions, please see:
<http://gcc.gnu.org/bugs.html>.

I:\Users\andrey>
```

# С. Дизассемблирование

- Дизассемблирование – получение из программы её исходного кода на языке Assembler.

Исходная программа	Ассемблированный	и	Дизассемблированный текст
MOV BX, Label_1	BB0001		mov bx, 0100h
JMP BX	FFE3		jmp bx
Label_1:			

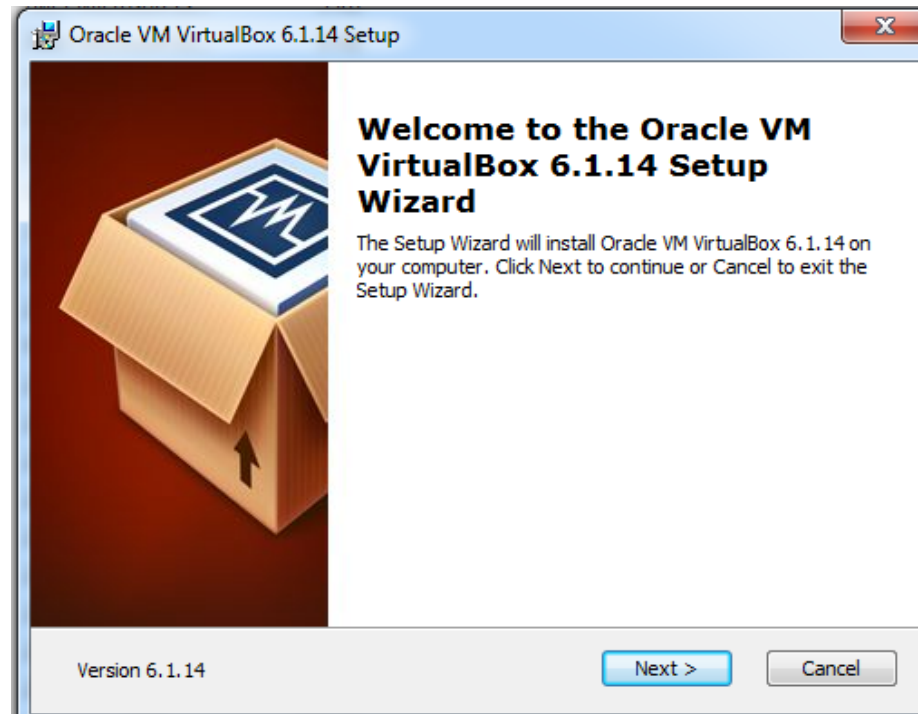
# С. Дизассемблирование

Исходная программа	Ассемблированный	и	Дизассемблированный текст
MOV AX,offset Table	B81000		mov ax,0010h
MOV BX,200h ;index	BB0002		mov bx,0200h
ADD AX,BX	01D8		add ax,bx
MOV AX,[BX]	8B07		mov ax,word ptr [bx]

Часто используемый инструмент дизассемблирования  
– IDA (Interactive DisAssembler)

# Установка виртуальной машины 1

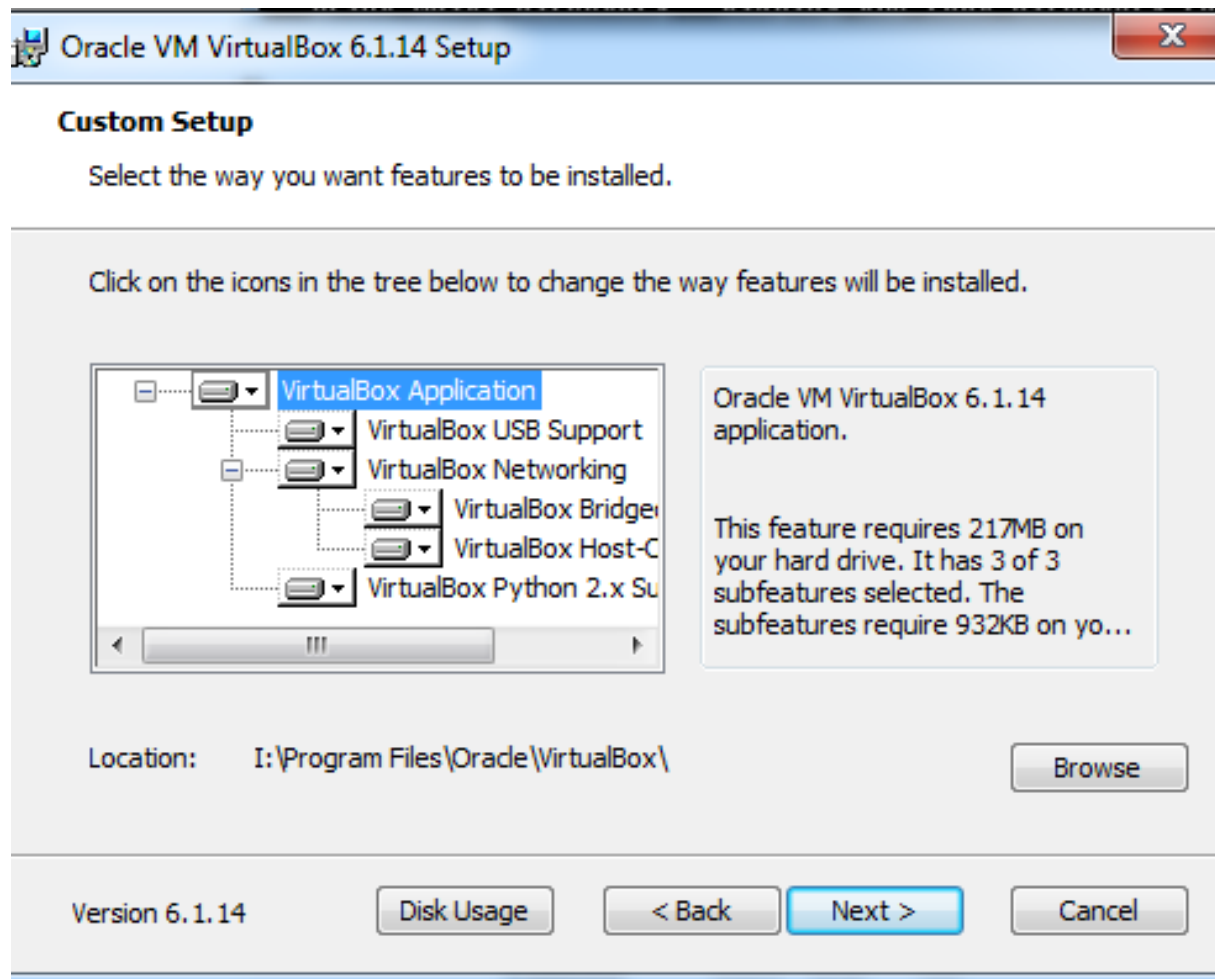
## Oracle VirtualBox





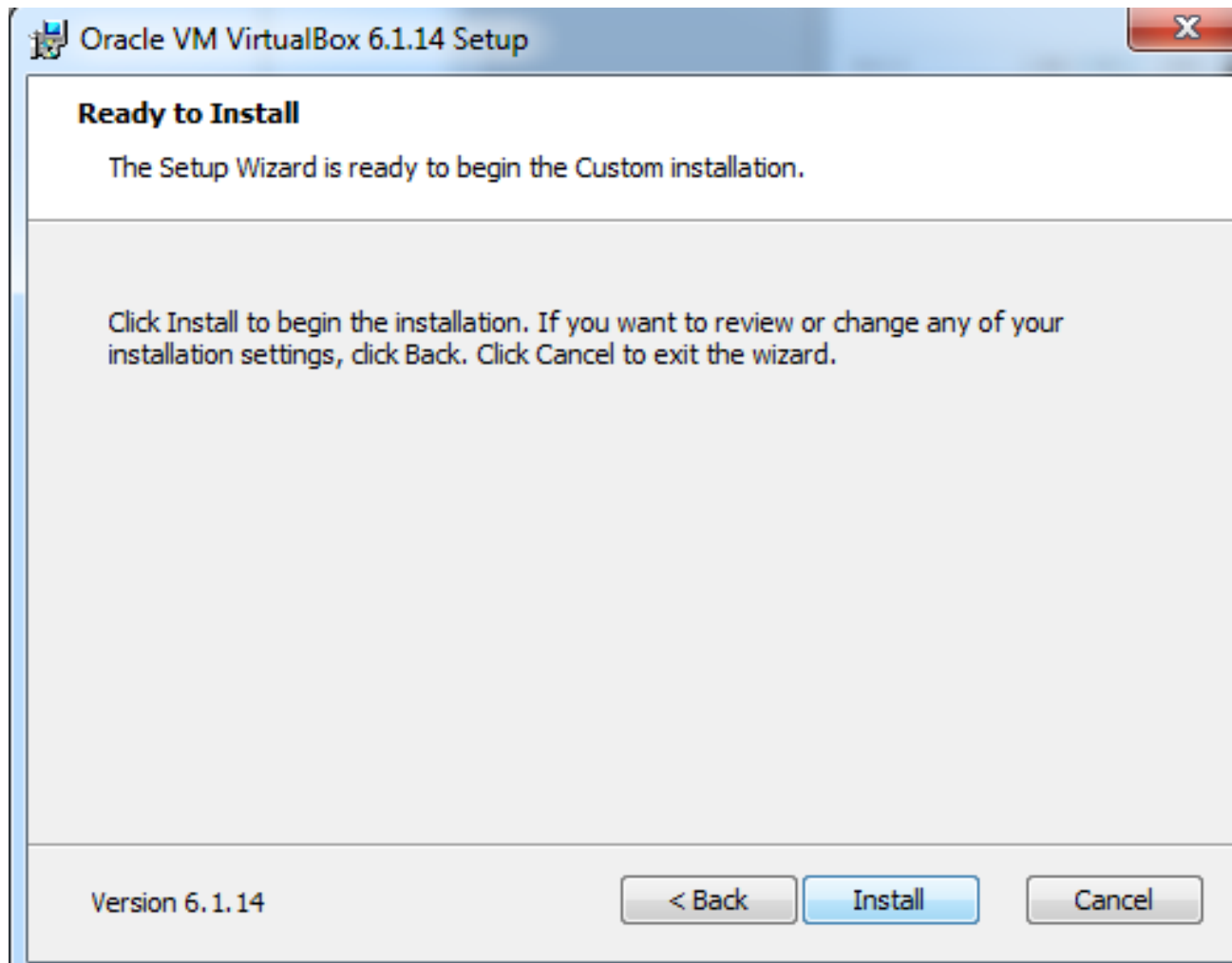
# Установка виртуальной машины 2

## Oracle VirtualBox

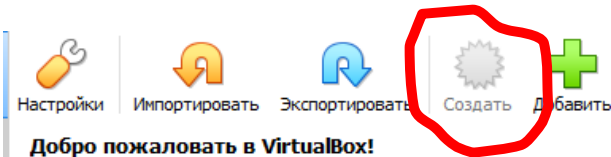


# Установка виртуальной машины 3

## Установка



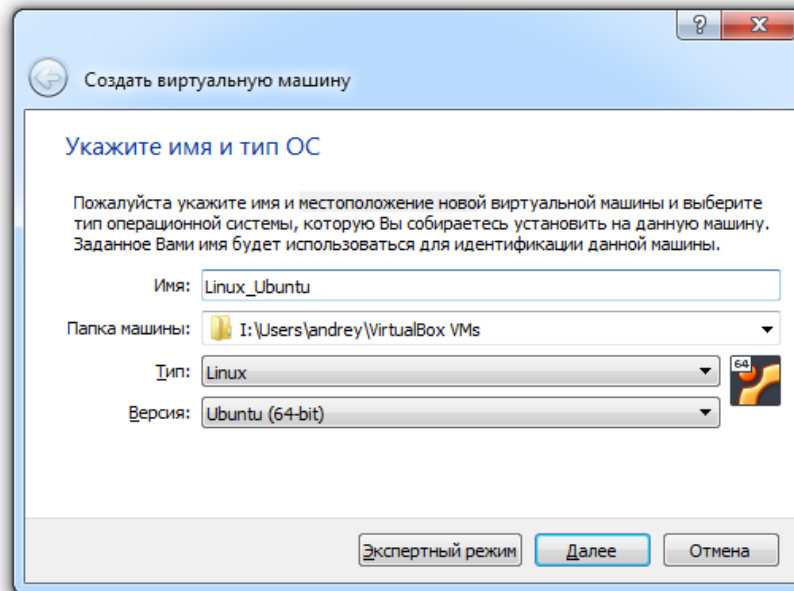
# Создание виртуальной машины 1



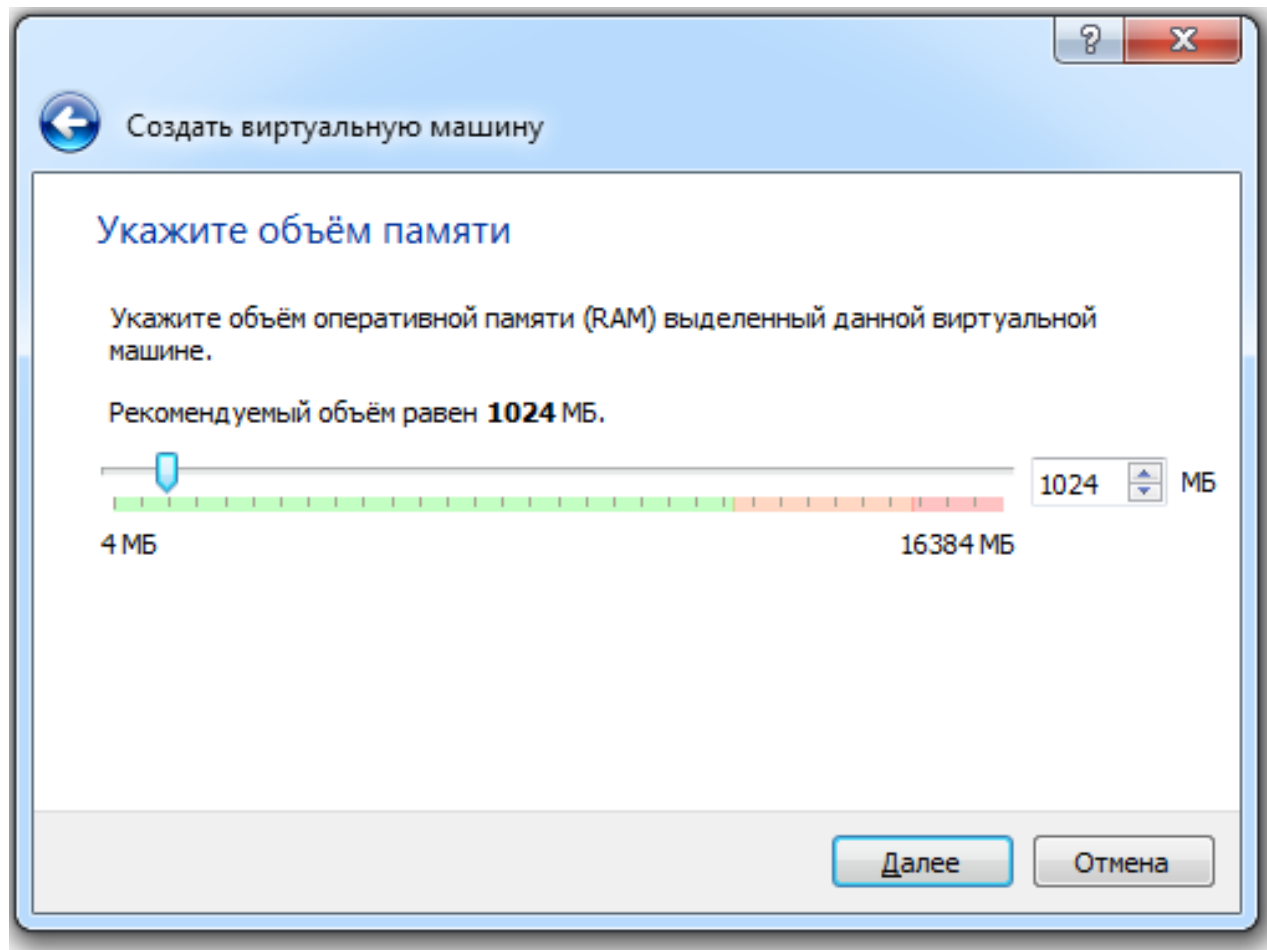
## Добро пожаловать в VirtualBox!

Левая часть окна приложения содержит глобальные инструменты, а также список всех виртуальных машин и их групп на Вашем компьютере. Вы можете соответствующие кнопки панели инструментов. Для получения списка инструментов текущего элемента, нажмите соответствующую кнопку элемента.

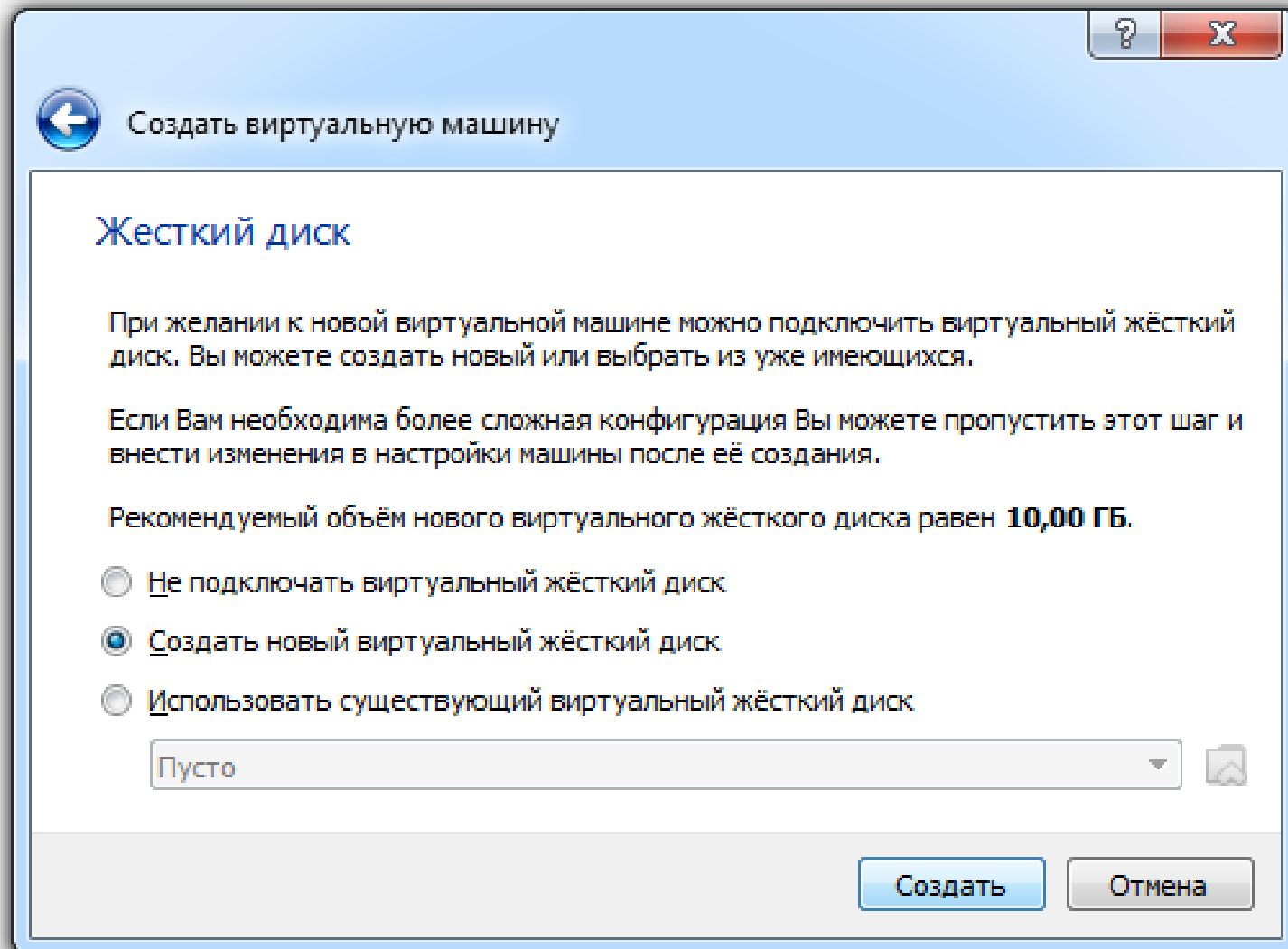
Вы можете нажать кнопку **F1** для получения помощи, или посетить [www.virtualbox.org](http://www.virtualbox.org) для дальнейшей информации и свежих новостей.



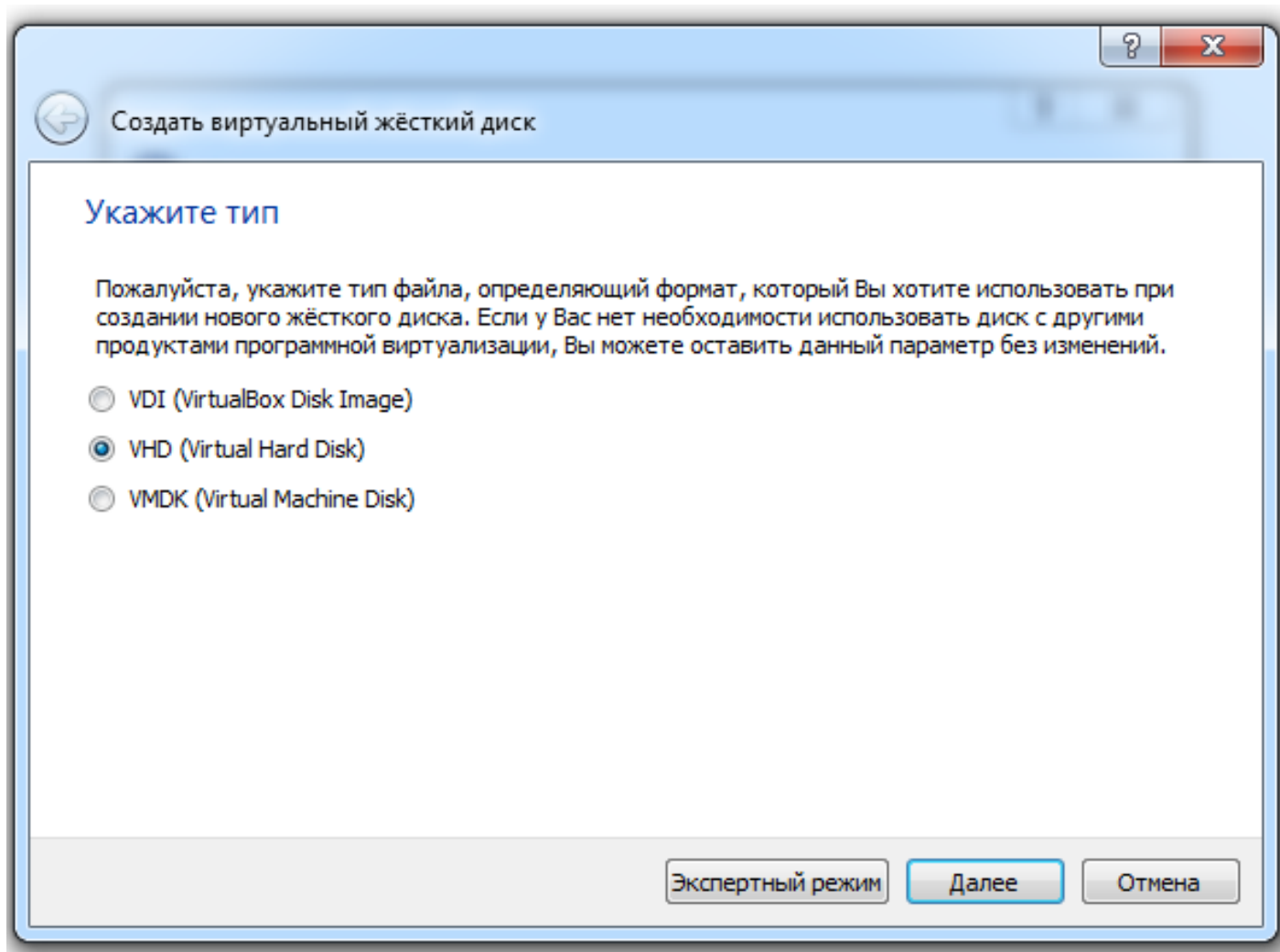
# Создание виртуальной машины 2



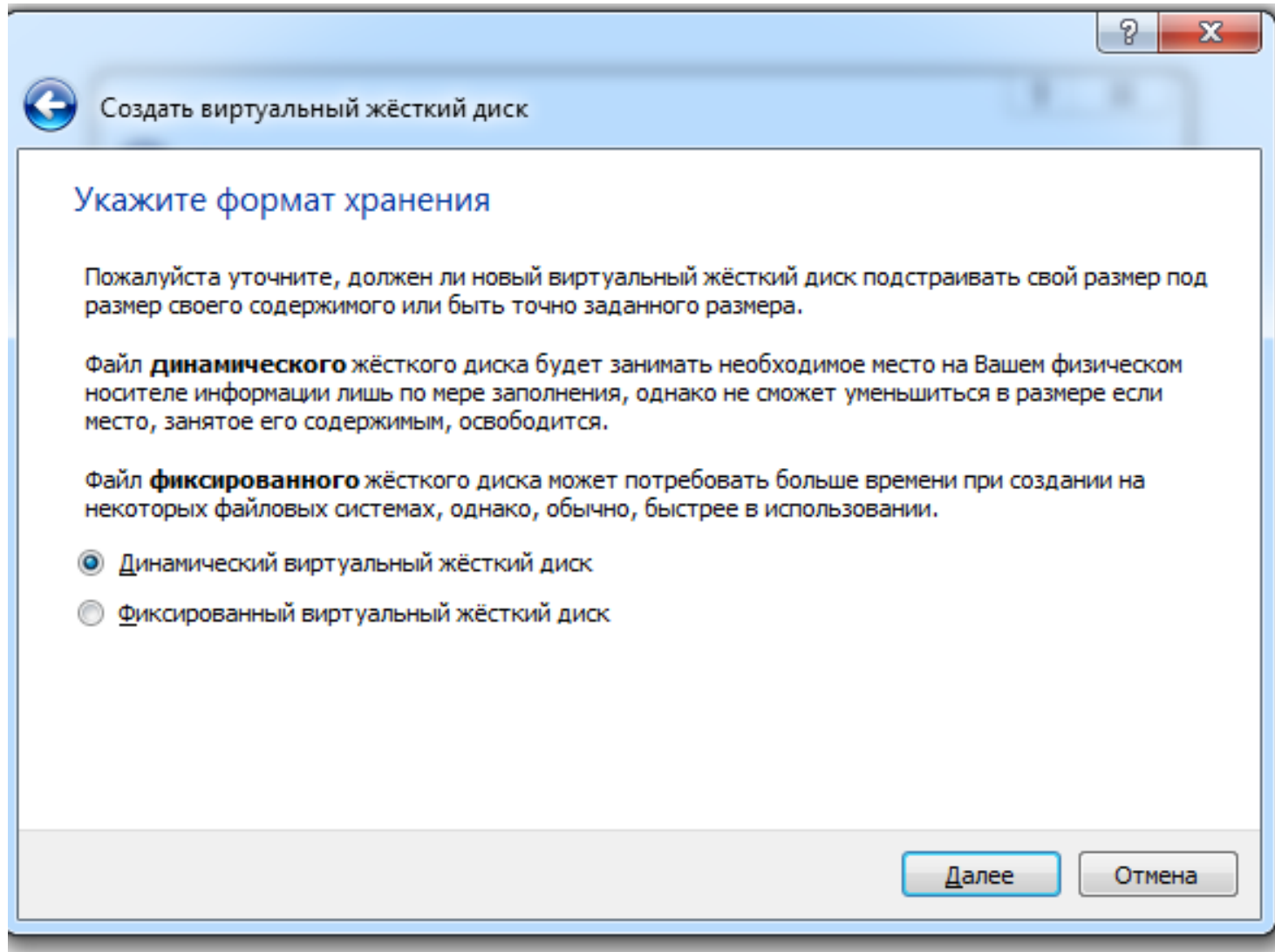
# Создание виртуальной машины 3



# Создание виртуальной машины 4

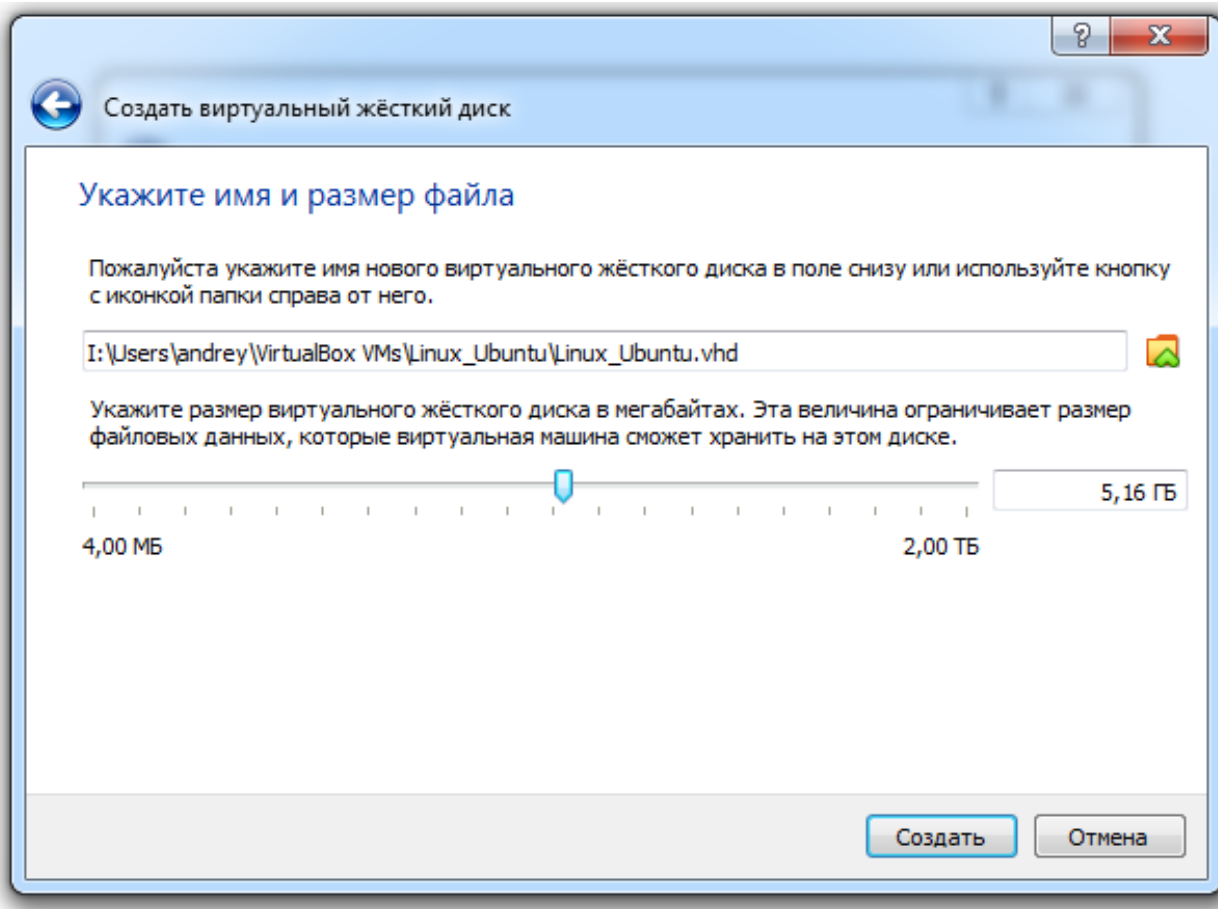


# Создание виртуальной машины 5



# Создание виртуальной машины 6

Лучше НЕ менять то, что задано по умолчанию!







# Создание виртуальной машины 8

Файл Машина Справка

Инструменты

Linux\_Ubuntu  
Выключена

Создать Настроить Сбросить **Запустить**

**Общие**  
Имя: Linux\_Ubuntu  
ОС: Ubuntu (64-bit)

**Система**  
Оперативная память: 1024 МБ  
Порядок загрузки: Гибкий диск, Оптический диск, Жёсткий диск  
Ускорение: VT-x/AMD-V, Nested Paging, Паравиртуализация KVM

**Дисплей**  
Видеопамять: 16 МБ  
Графический контроллер: VMSVGA  
Сервер удалённого дисплея: Выключен  
Запись: Выключена

**Носители**  
Контроллер: IDE  
Вторичный мастер IDE: [Оптический привод] Пусто  
Контроллер: SATA  
SATA порт 0: Linux\_Ubuntu.vhd (Обычный, 5,16 ГБ)

**Аудио**  
Аудиодрайвер: Windows DirectSound  
Аудиоконтроллер: ICH AC97

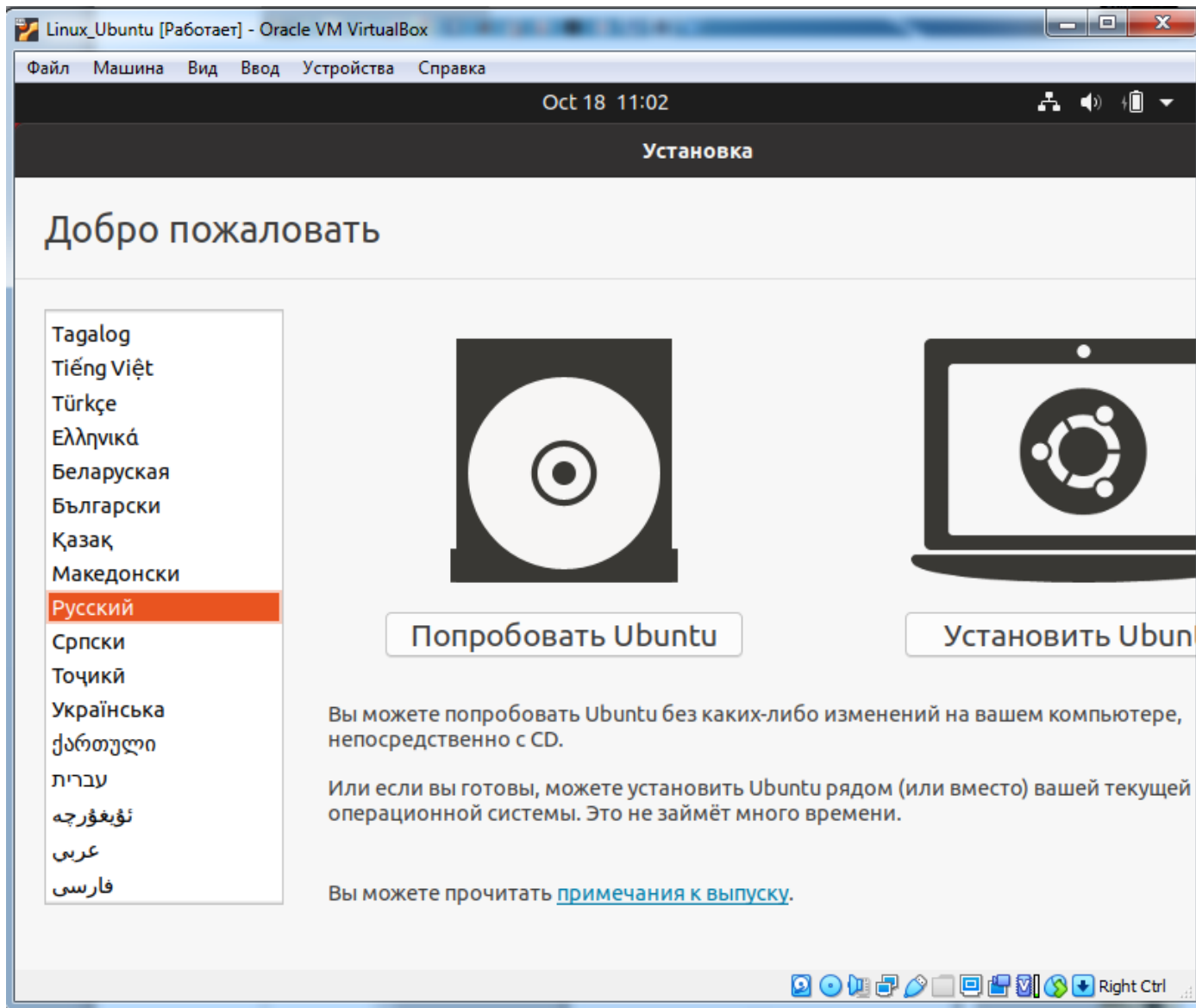
**Сеть**  
Адаптер 1: Intel PRO/1000 MT Desktop (NAT)

**USB**  
USB-контроллер: OHCI  
Фильтры устройств: 0 (0 активно)

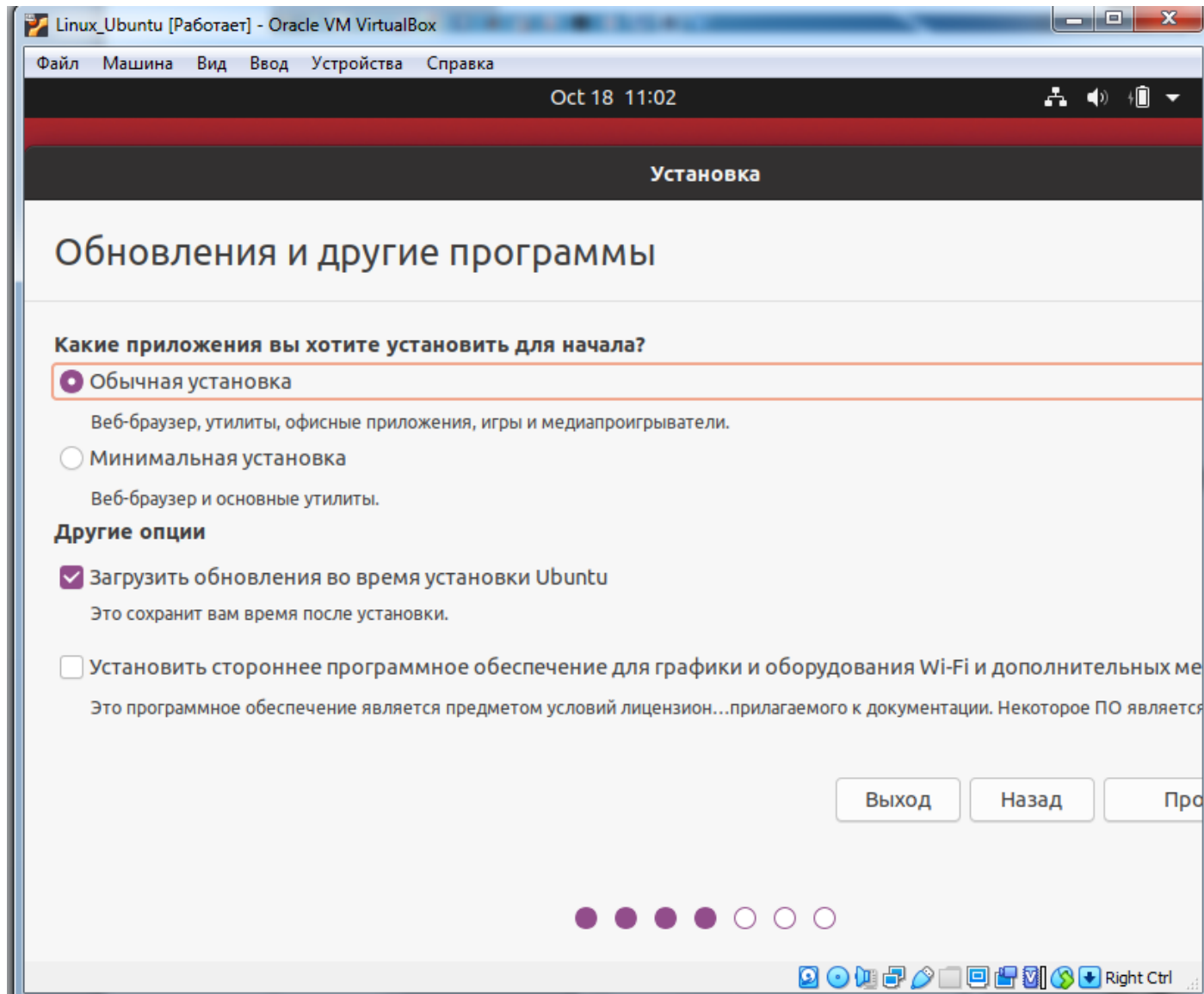
**Общие папки**  
Отсутствуют

**Описание**  
Отсутствует

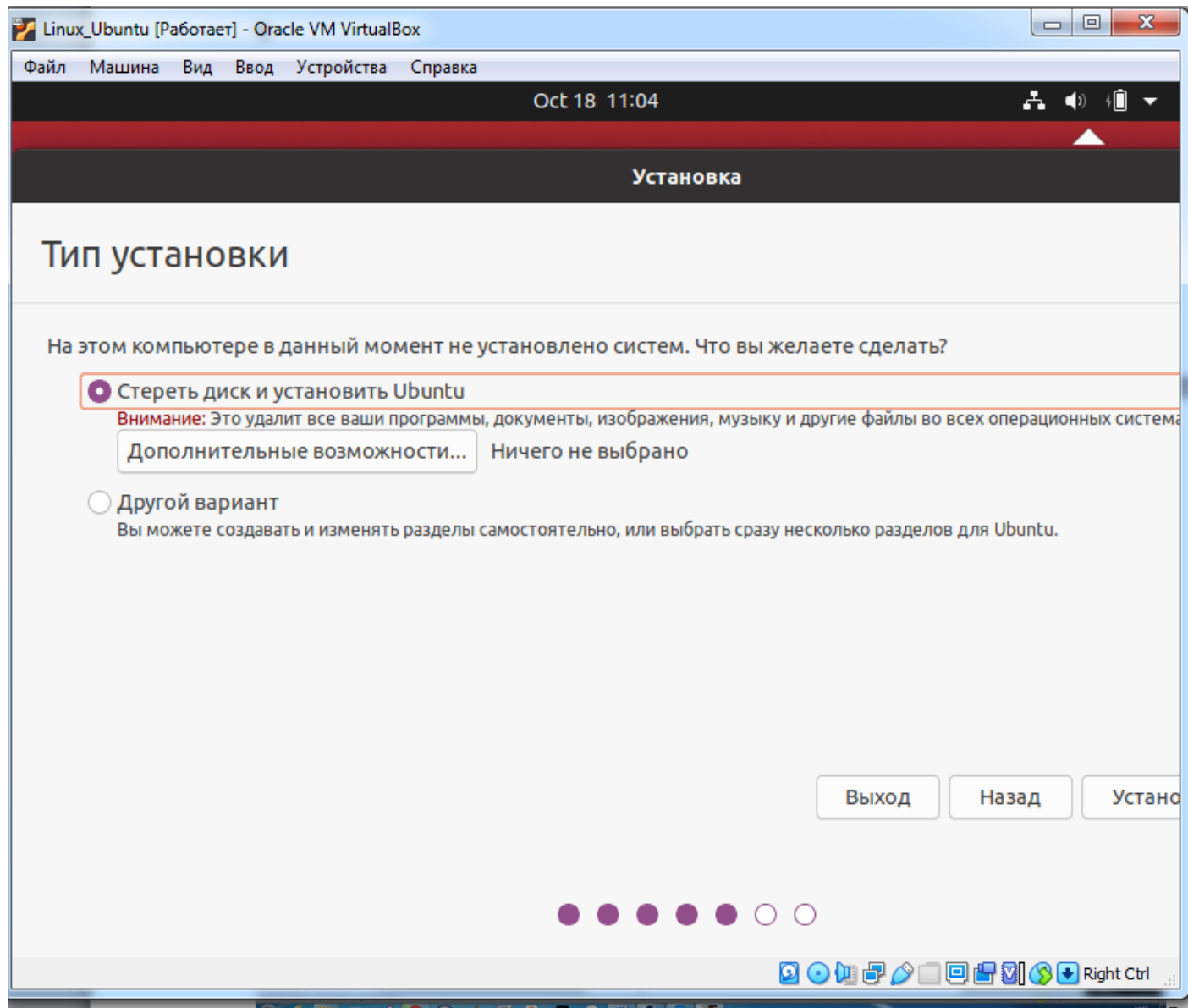
# Установка ОС Linux на ВМ 1



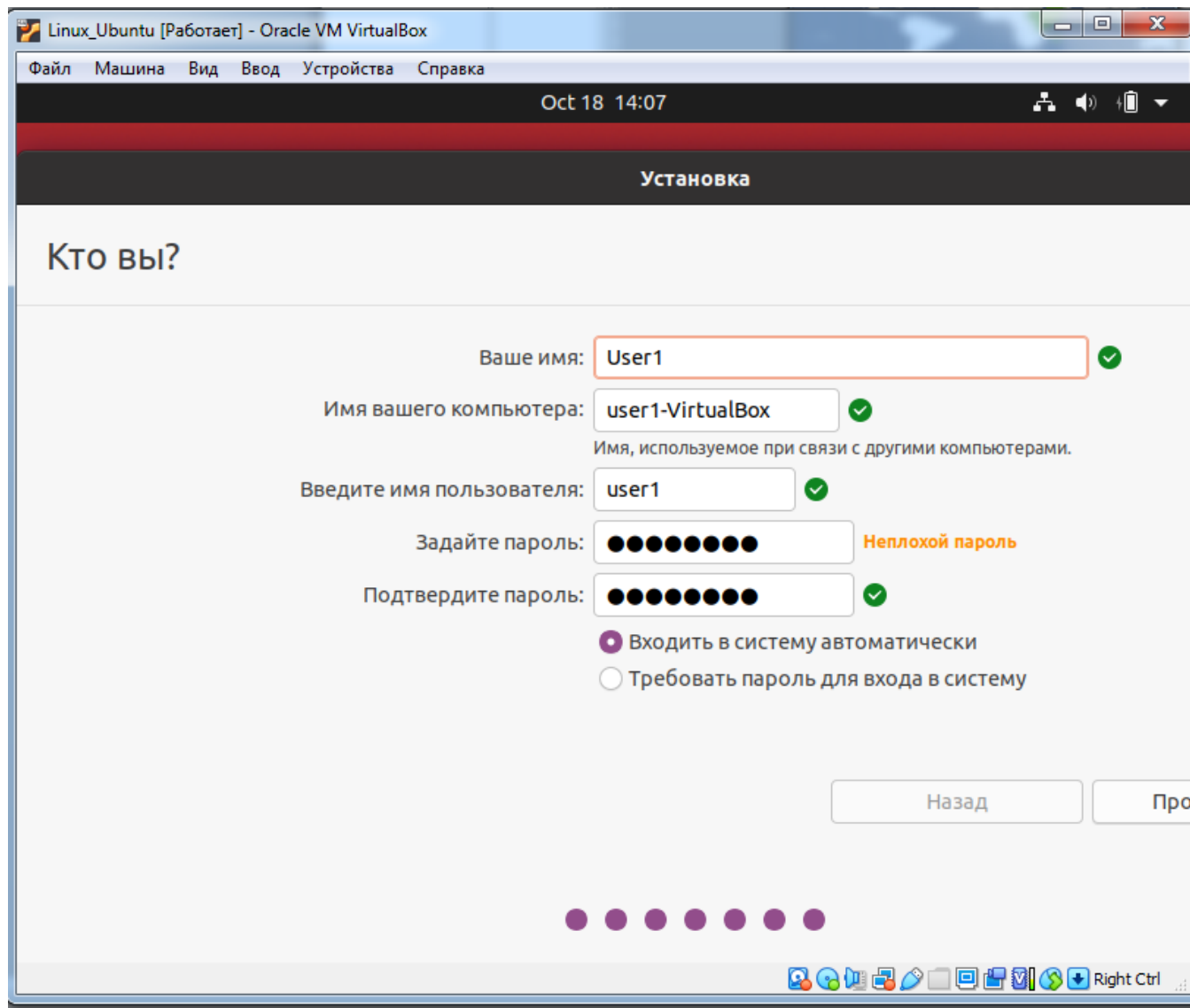
# Установка ОС Linux на ВМ 2



# Установка ОС Linux на ВМ 3



# Установка ОС Linux на ВМ 4



# Установка ОС Linux на ВМ 5

