

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ИНГУШСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра «Информационные системы и технологии»

СОГЛАСОВАНО

Руководитель образовательной программы

_____/М.Х. Мальсагов
«20» мая 2024г.

УТВЕРЖДАЮ

И.о. декана физико-математического
факультета

_____/Б.С.Кульбужев
«23» мая 2024г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ (МОДУЛЯ)
Б1.В.07 «Операционные системы»**

Направление подготовки

09.03.02 Информационные системы и технологии

Направленность (профиль подготовки)

Информационные системы и технологии

Квалификация выпускника

Бакалавр

Форма обучения

Очная, заочная, очно-заочная

Магас, 2024г

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Перечень формируемых компетенций

УК-6. Способен определить и реализовать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки

ОПК-7. Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений.

ПК-3. Способен предлагать и адаптировать методики оценки качества проводимых исследований, составлять отчеты о проделанной работе, подготавливать обзоры, готовить публикации.

Паспорт фонда оценочных средств

№ п/п	Контролируемые разделы (темы) дисциплины	Код Контролируемой компетенции (или ее части)	Наименование оценочного средства
1.	Тема 1. Назначение, классификация и структура операционных систем.	УК-6, ОПК-7, ПК-3	Тест
2.	Тема 2. Файловые системы. Системные вызовы для работы с файлами, каталогами и файловыми системами.	УК-6, ОПК-7, ПК-3	Тест
3.	Тема 3. Время в операционной системе. Системные вызовы и команды оболочки для работы со временем.	УК-6, ОПК-7, ПК-3	Коллоквиум
4.	Тема 4. Командные языки для пакетной обработки операционных систем. Оболочка bash. Встроенные и внешние команды. Условные операторы и операторы цикла. Скрипты.	УК-6, ОПК-7, ПК-3	Коллоквиум
5.	Тема 5. Процессы и система управления заданиями. Планирование процессов, обслуживания на этапе краткосрочного планирования	УК-6, ОПК-7, ПК-3	Тест, коллоквиум

6.	Тема 6. Межпроцессный обмен. Типы межпроцессного обмена (IPC). Сигнально-семафорный механизм. Каналы, именованные каналы, разделяемая память.	УК-6, ОПК-7, ПК-3	Тест, коллоквиум
7.	Тема 7. Структура драйвера в ОС. Специальные файлы ОС UNIX (Linux). Команды оболочки и системные вызовы для работы со специальными файлами и. Конфигурация ядра и его генерация. вызовы для работы со специальными файлами. Конфигурация ядра и его генерация.	УК-6, ОПК-7, ПК-3	Тест, коллоквиум
8.	Тема 8. Планирование памяти в ОС. Задачи планирования.	УК-6, ОПК-7, ПК-3	Тест
9.	Тема 9. Сетевая подсистем ОС. Настройки сети	УК-6, ОПК-7, ПК-3	Тест
10.	Тема 10. Виртуализация. Виды виртуализации, аппаратные и программные средства	УК-6, ОПК-7, ПК-3	Тест

Типовой тест промежуточной аттестации

Вопрос 1. Какие ОС называются мультипрограммными

1. обеспечивающие одновременную работу нескольких пользователей
2. поддерживающие сетевую работу компьютеров
- +3. обеспечивающие запуск одновременно нескольких программ
4. состоящие более чем из одной программы

Вопрос 2. Какие существуют способы реализации ядра системы?

- +1. многоуровневая (многослойная) организация
- +2. микроядерная организация
3. реализация распределенная
4. монолитная организация

Вопрос 3. Что обычно входит в состав ядра ОС

- +1. высокоуровневые диспетчеры ресурсов
- +2. аппаратная поддержка функций ОС процессором
- + 3. базовые исполнительные модули
- + 4. набор системных API-функций

Вопрос 4. Какие особенности характерны для современных универсальных операционных систем?

- + 1. поддержка многозадачности
- + 2. поддержка сетевых функций
- + 3. обеспечение безопасности и защиты данных

4. предоставление большого набора системных функций разработчикам приложений **Вопрос 5. Какие утверждения относительно понятия «API-функция» являются правильными?**

- + 1. API-функции определяют прикладной программный интерфейс
- + 2. API-функции используются при разработке приложений для доступа к ресурсам компьютера
- 3. API-функции реализуют самый нижний уровень ядра системы
- 4. API-функции — это набор аппаратно реализованных функций системы

Вопрос 6. Какие особенности характерны для ОС Unix

- + 1. открытость и доступность исходного кода
- 2. ориентация на использование оконного графического интерфейса
- + 3. использование языка высокого уровня C
- + 4. возможность достаточно легкого перехода на другие аппаратные платформы **Вопрос 7. Какие типы операционных систем используются наиболее часто в настоящее время?**

7. Какие типы операционных систем используются наиболее часто в настоящее время?

- + 1. системы семейства Windows
- + 2. системы семейства Unix/Linux
- 3. системы семейства MS DOS
- 4. системы семейства IBM OS 360/370

Вопрос 8. Какие задачи необходимо решать при создании мультипрограммных ОС

- + 1. защита кода и данных разных приложений, размещенных вместе в основной памяти
- + 2. централизованное управление ресурсами со стороны ОС
- + 3. переключение процессора с одного приложения на другое
- 4. необходимость размещения в основной памяти кода и данных сразу многих приложений

Вопрос 9. Какое соотношение между используемыми на СЕРВЕРАХ операционными системами сложилось в настоящее время?

- + 1. примерно поровну используются системы семейств Windows и Unix/Linux
- 2. около 10 % — системы семейства Windows, около 90 % — системы семейства Unix/Linux
- 3. около 90 % — системы семейства Windows, около 10 % — системы семейства Unix/Linux
- 4. около 30 % — системы семейства Windows, около 30 % — системы семейства Unix/Linux, около 40 % — другие системы

Вопрос 10. Какие утверждения относительно понятия «Ядро операционной системы» являются правильными?

- + 1. ядро реализует наиболее важные функции ОС
- + 2. подпрограммы ядра выполняются в привилегированном режиме работы процессора
- 3. ядро в сложных ОС может строиться по многоуровневому принципу
- 4. ядро всегда реализуется на аппаратном уровне

Вопрос 11. Какие сообщения возникают при нажатии на клавиатуре алфавитно-цифровой клавиши?

- + 1. WM_KeyDown
- + 2. WM_Char
- + 3. WM_KeyUp
- 4. WM_KeyPress

Вопрос 12. Какие шаги в алгоритме взаимодействия приложения с системой выполняются операционной системой

- 1. формирование сообщения и помещение его в системную очередь
- + 2. распределение сообщений по очередям приложений
- + 3. вызов оконной функции для обработки сообщения
- 4. извлечение сообщения из очереди приложения

Вопрос 13. Что представляет собой понятие “сообщение” (message)?

- 1. небольшую структуру данных, содержащую информацию о некотором событии

2. специальную API-функцию, вызываемую системой при возникновении события
3. однобайтовое поле с кодом происшедшего события

+ 4. небольшое окно, выводящее пользователю информацию о возникшем событии

Вопрос 14. Какие утверждения относительно иерархии окон являются справедливыми

- + 1. главное окно может содержать любое число подчиненных окон
- + 2. любое подчиненное окно может содержать свои подчиненные окна
3. подчиненные окна могут быть двух типов – дочерние и всплывающие
- + 4. приложение может иметь несколько главных окон

Вопрос 15. Как можно узнать координаты текущего положения мыши при нажатии левой кнопки

- + 1. с помощью события WM_LBUTTONDOWN и его поля LPARAM
2. с помощью события WM_LBUTTONDOWN и его поля WPARAM
3. с помощью события WM_LBUTTONDOWN и его полей WPARAM и LPARAM
4. с помощью события WM_LbuttonCoordinates

Вопрос 16. Какие функции можно использовать для получения контекста устройства?

- + 1. GetDC
- + 2. BeginPaint
3. ReleaseDC
4. CreateContext

Вопрос 17. Какая инструкция (оператор) является основной при написании оконной функции?

- + 1. инструкция множественного выбора типа Case — Of
2. условная инструкция if – then
3. инструкция цикла с известным числом повторений
4. инструкция цикла с неизвестным числом повторений

Вопрос 18. Какой вызов позволяет добавить строку в элемент-список?

- + 1. SendMessage (MyEdit, lb_AddString, 0, строка)
2. SendMessage (“Edit”, lb_AddString, 0, строка)
3. SendMessage (MyEdit, AddString, 0, строка)
4. SendMessage (MyEdit, строка, lb_AddString, 0)

Вопрос 19. Какие утверждения относительно оконной функции являются правильными

- + 1. оконная функция принимает 4 входных параметра
- + 2. тело оконной функции – это инструкция выбора с обработчиками событий
- + 3. оконная функция обязательно должна обрабатывать сообщение wm_Destroy
- + 4. оконная функция явно вызывается из основной функции приложения

Вопрос 20. Какие сообщения возникают при нажатии на клавиатуре функциональной клавиши?

- + 1. WM_KeyDown
- + 2. WM_KeyUp
3. WM_KeyPress
4. WM_Char

Вопрос 21. Что может быть причиной появления внутреннего прерывания

- + 1. попытка деления на ноль
2. попытка выполнения запрещенной команды
- + 3. попытка обращения по несуществующему адресу
4. щелчок кнопкой мыши

Вопрос 22. Какие операции определяют взаимодействие драйвера с контроллером

- + 1. проверка состояния устройства
- + 2. запись данных в регистры контроллера

+ 3. чтение данных из регистров контроллера
4. обработка прерываний от устройства

Вопрос 23. Какие операции включает в себя вызов обработчика нового прерывания

+ 1. обращение к таблице векторов прерываний для определения адреса первой команды вызываемого обработчика

2. сохранение контекста для прерываемого программного кода

+ 3. занесение в счетчик команд начального адреса вызываемого обработчика

+ 4. внесение необходимых изменений в таблицу векторов прерываний

Вопрос 24. Что входит в программный уровень подсистемы ввода/вывода

+ 1. драйверы

2. диспетчер ввода/вывода

+ 3. системные вызовы

4. контроллеры

Вопрос 25. Что определяет понятие “порт ввода/вывода”

+ 1. порядковый номер или адрес регистра контроллера

2. машинную команду ввода/вывода

3. устройство ввода/вывода

4. контроллер устройства ввода/вывода

Вопрос 26. Какие существуют типы прерываний

+ 1. внешние или аппаратные прерывания

+ 2. внутренние прерывания или исключения

+ 3. программные псевдопрерывания

4. системные прерывания

Вопрос 27. Какие утверждения относительно понятия прерывания являются правильными

+ 1. прерывания — это механизм реагирования вычислительной системы на происходящие в ней события

2. прерывания используются для синхронизации работы основных устройств вычислительной системы

+ 3. прерывания возникают в непредсказуемые моменты времени

4. прерывания — это основной механизм планирования потоков

Вопрос 28. Какую информацию могут содержать регистры контроллеров устройства

+ 1. текущее состояние устройства

+ 2. текущую выполняемую устройством команду

3. данные, передаваемые от устройства системе

4. данные, передаваемые системой устройству

Вопрос 29. Как выстраиваются аппаратные прерывания в зависимости от их приоритета

1. сбой аппаратуры > таймер > дисковые устройства > сетевые устройства > клавиатура и мышь

2. сбой аппаратуры > таймер > дисковые устройства > клавиатура и мышь > сетевые устройства

+ 3. таймер > сбой аппаратуры > дисковые устройства > сетевые устройства > клавиатура и мышь

4. сбой аппаратуры > дисковые устройства > таймер > сетевые устройства > клавиатура и мышь

Вопрос 30. Что может быть причиной появления внешнего прерывания

+ 1. нажатие клавиши на клавиатуре

+ 2. завершение дисковой операции обращение выполняемой процессором команды по несуществующему адресу попытка выполнения запрещенной команды

Типовой вариант задания на контрольную работу

Изучить особенности приоритетного планирования процессов.

Рассмотреть алгоритм работы планировщика для не вытесняющего приоритетного планирования и для вытесняющего приоритетного планирования.

Рассчитать среднее время ожидания и среднее время исполнения процессов для каждого варианта работы планировщика.

Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

1. Итоговый контрольный тест доступен студенту только во время тестирования, согласно расписания занятий или в установленное деканатом время.
2. Студент информируется о результатах текущей успеваемости.
3. Студент получает информацию о текущей успеваемости, начислении бонусных баллов и допуске к процедуре итогового тестирования от преподавателя или в ЭИОС.
4. Производится идентификация личности студента.
5. Студентам, допущенным к промежуточной аттестации, открывается итоговый контрольный тест.
6. Тест закрывается студентом лично по завершении тестирования или автоматически по истечении времени тестирования.

Экзаменационные вопросы по дисциплине «Операционные системы»

1. Основные этапы развития ОС.
2. Основные задачи, решаемые ОС при управлении вычислительными процессами.
3. Каковы причины появления ОС пакетной обработки информации?
4. Сетевые ОС. Основные отличия от несетевых ОС.
5. Основные ресурсы ВС, которыми управляет ОС.
6. Задачи управления ресурсами ВС.
7. Основные типы ОС.
8. Особенности функционирования распределенных ОС.
9. ОС как совокупность средств для управления ресурсами ВС.
10. Монолитная структура ОС.
11. Слоистая структура ОС.
12. Микроядерная структура ОС.
13. Пользовательский режим работы.
14. Основные средства обеспечения переносимости ОС.
15. Понятие процесса.

- 16 Диаграмма состояний процесса.
- 17 Понятие потока (нити).
- 18 Способы реализации многопоточковых вычислений.
- 19 Планирование процессов.
- 20 Основные задачи планирования процессов.
- 21 Краткосрочное и долгосрочное планирование.
- 22 Основные задачи управления памятью.
- 23 Виртуальные адреса.
- 24 Преобразование виртуальных адресов в физические.
- 25 Задача распределения памяти.
- 26 Страничная организация памяти.
- 27 Таблица страниц.
- 28 Управление страничным обменом.
- 29 Сегментная организация памяти.
- 30 Сегментно -страничная организация памяти

Типовая лабораторная работа

Создание и настройка виртуальной машины VirtualBox на ПК с операционной системой Ubuntu

Цель работы:

1. Установка платформы виртуализации VirtualBox.
2. Создание и настройка виртуальной машины VirtualBox с ОС Ubuntu.
3. Практическое знакомство с ОС Ubuntu-desktop x32.
4. Компиляция программ на C/C++ в терминале Ubuntu-desktop x32.

Введение

С помощью бесплатной программы VirtualBox можно создать на своем компьютере виртуальную машину с другой гостевой операционной системой. Платформа виртуализации VirtualBox создает виртуальные машины, в которые можно будет установить разные операционные системы: Windows, Linux, Mac OS X и т. д.

VirtualBox – платформа виртуализации, имитирующее работу ПК. Позволяет устанавливать и запускать операционные системы как обыкновенные приложения. Создает на ПК изолированное окружение, состоящее из: жесткого диска, видеокарты, памяти, контроллеров устройств. Может кому-то потребоваться включить виртуализацию. Дело в том, что по умолчанию в настройках BIOS большинства материнских плат виртуализация отключена. Ее необходимо включить, зайдя в BIOS в соответствующий раздел, который называется у каждого производителя по-своему, например, «Virtualization Technology», изменив значение опции с «Disabled» на «Enabled».

Популярные способы применения:

1. Знакомство с другими ОС: Linux, FreeBSD, MacOS, любая из версий Windows, Android. Система работает изолированно. Можно экспериментировать, не боясь, что

нарушится работа реальной системы.

2. Запуск программных продуктов, несовместимых с основной ОС.
3. Использование старых приложения.
4. Тестирование потенциально опасных приложений.

Ubuntu – дистрибутив Linux, основанный на ядре Linux. Основным разработчиком и спонсором является компания Canonical. В настоящее время проект активно развивается и поддерживается свободным сообществом. Ubuntu – это операционная система, которая идеально подходит для использования на персональных компьютерах, ноутбуках и серверах. Она содержит все необходимые программы, которые нужны всем: программу просмотра Интернет, офисный пакет для работы с текстами, электронными таблицами и презентациями, программы для общения в Интернет и много других.

Ссылки на скачивание VirtualBox и Ubuntu:

VirtualBox-6.0.24-139119-Win.exe

<https://file.tpu.ru/index.php/s/v5PnrEHucLBuOIf>

Ubuntu-14.04.6-desktop-i386.iso

<https://file.tpu.ru/index.php/s/CQ7G6WPyhhi4cSF>

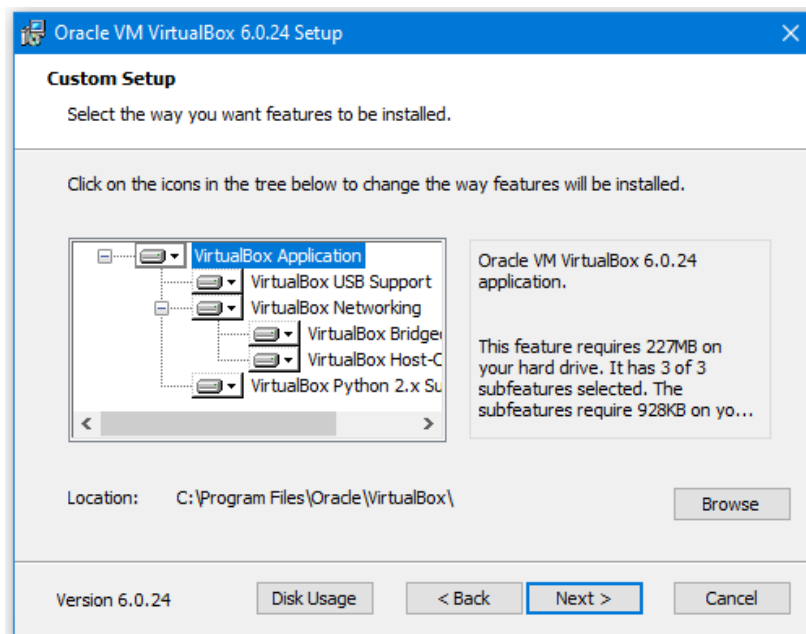
1. Установка платформы виртуализации VirtualBox

Дистрибутив представлен как один инсталляционный файл расширения «exe». Нажимаем по нему два раза мышкой. Откроется окно помощника. Нажимаем «Next».



Нажимаем «Next».

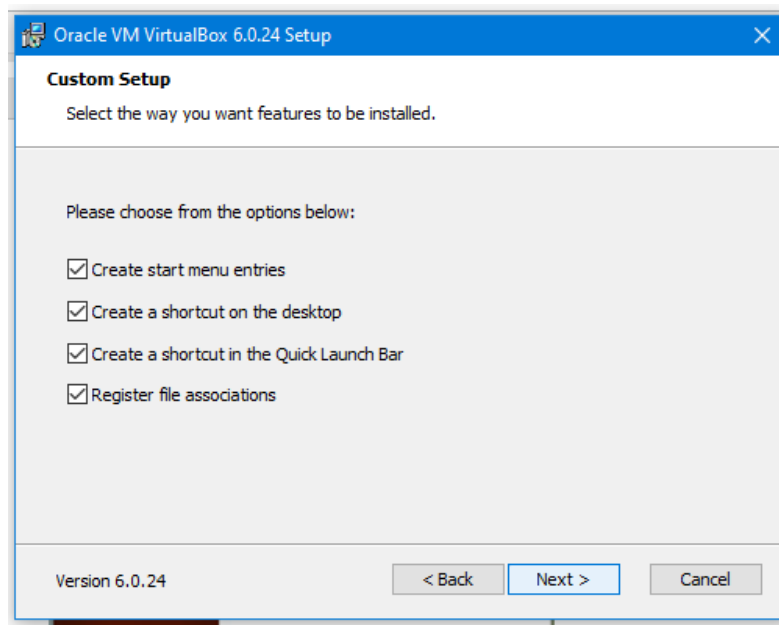
Выбираем место для инсталляции. Будет предложено установить все компоненты. Не рекомендуются без необходимости отключать их. Они нужны даже при минимальном использовании. Нажимаем «Next».



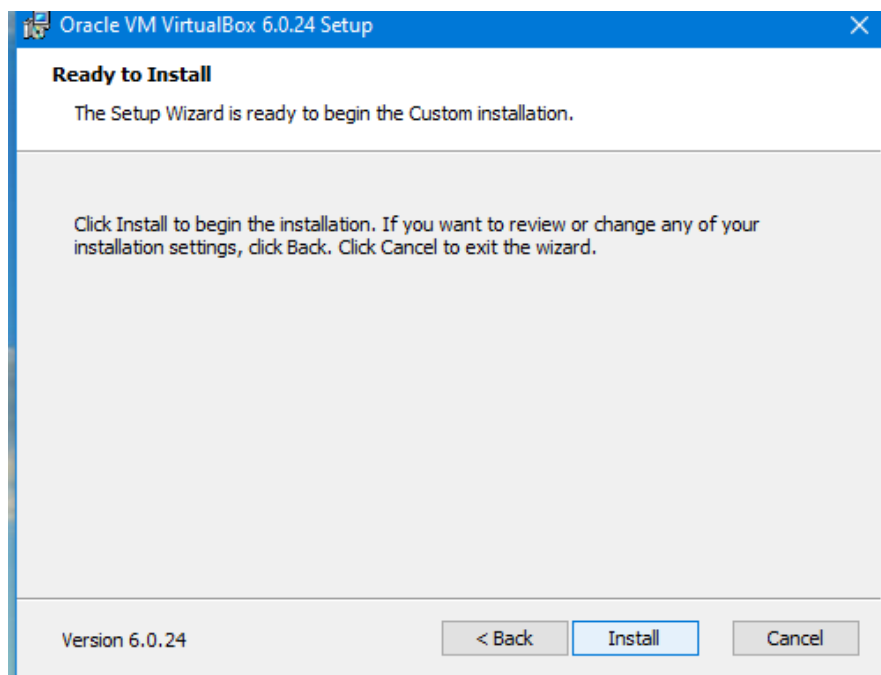
В новом окне расположены такие настройки запуска:

1. Создание пунктов меню "Пуск";
2. Создание ярлыка на рабочем столе;
3. Ярлык на панели быстрого запуска;
4. Зарегистрировать расширения файлов программы в ОС.

Нажимаем «Next».



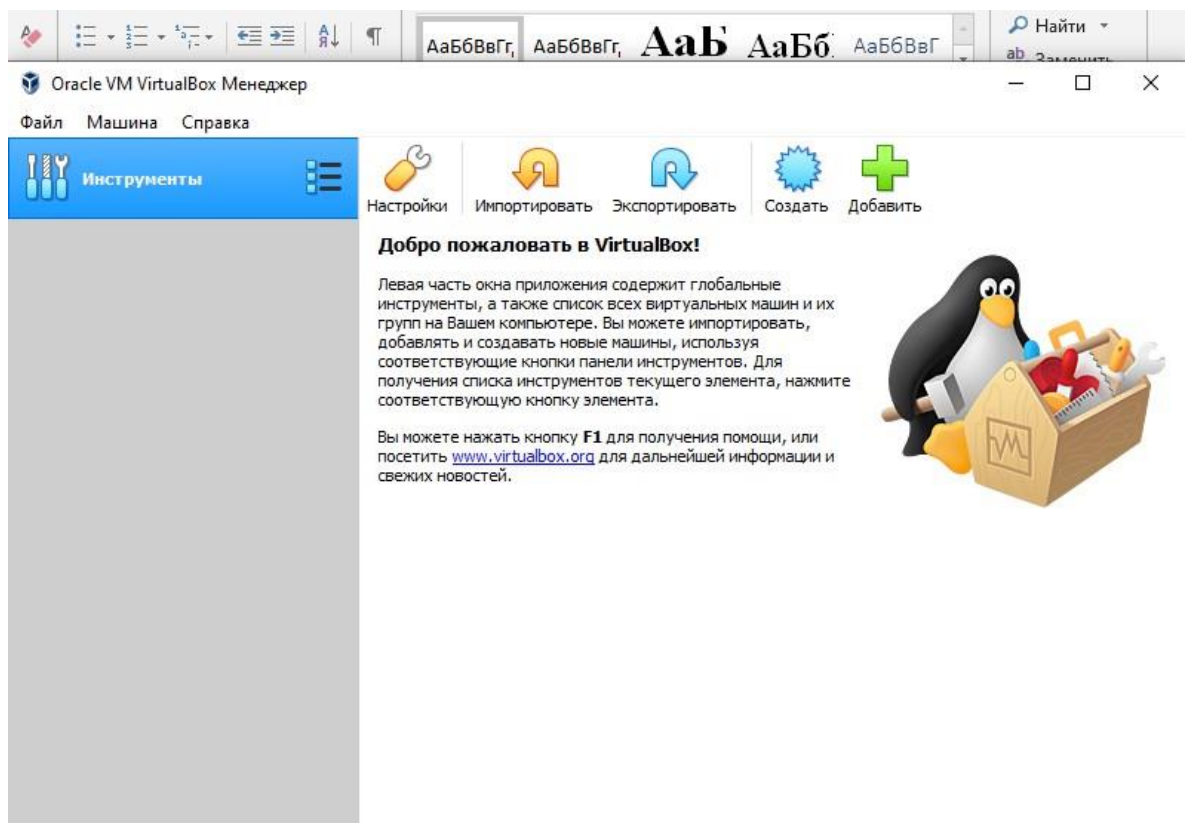
Будет предложено запустить процесс загрузки программы. Нажимаем «Install».



Далее нажимаем на «Finish».

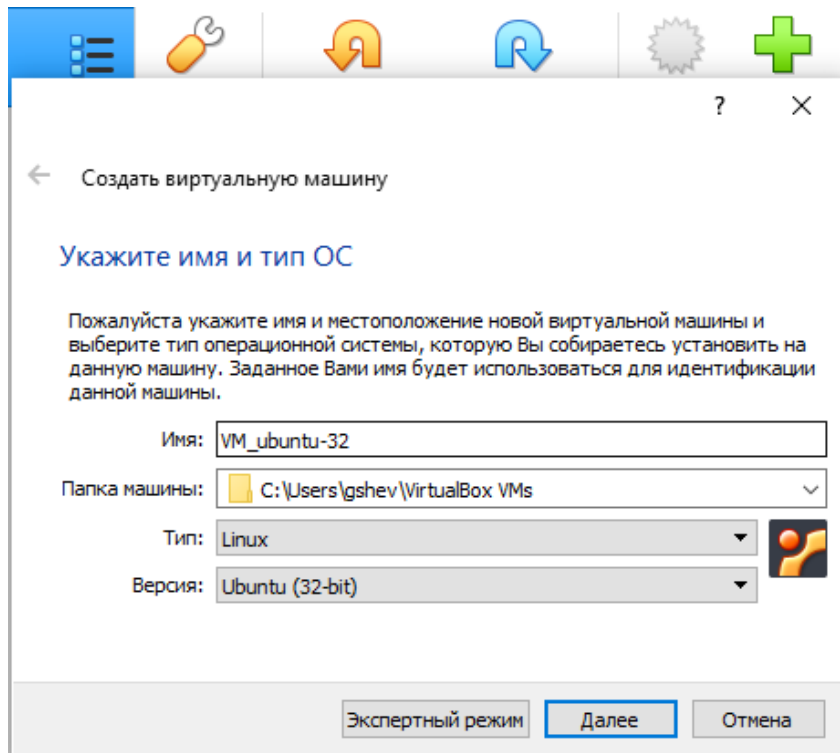


Платформа виртуализации установлена.

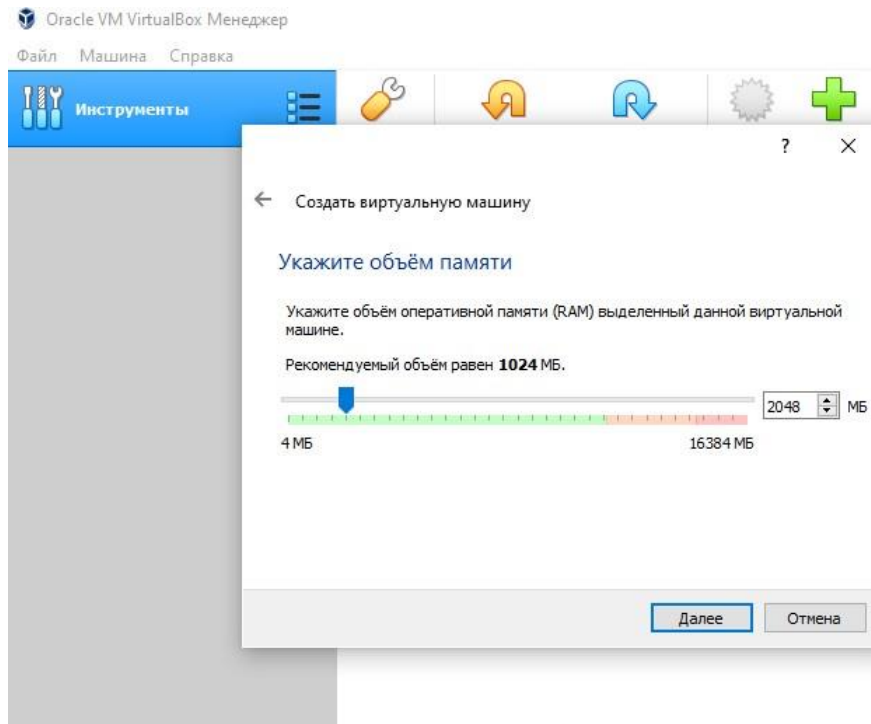


2. Создание и настройка виртуальной машины VirtualBox с ОС Ubuntu

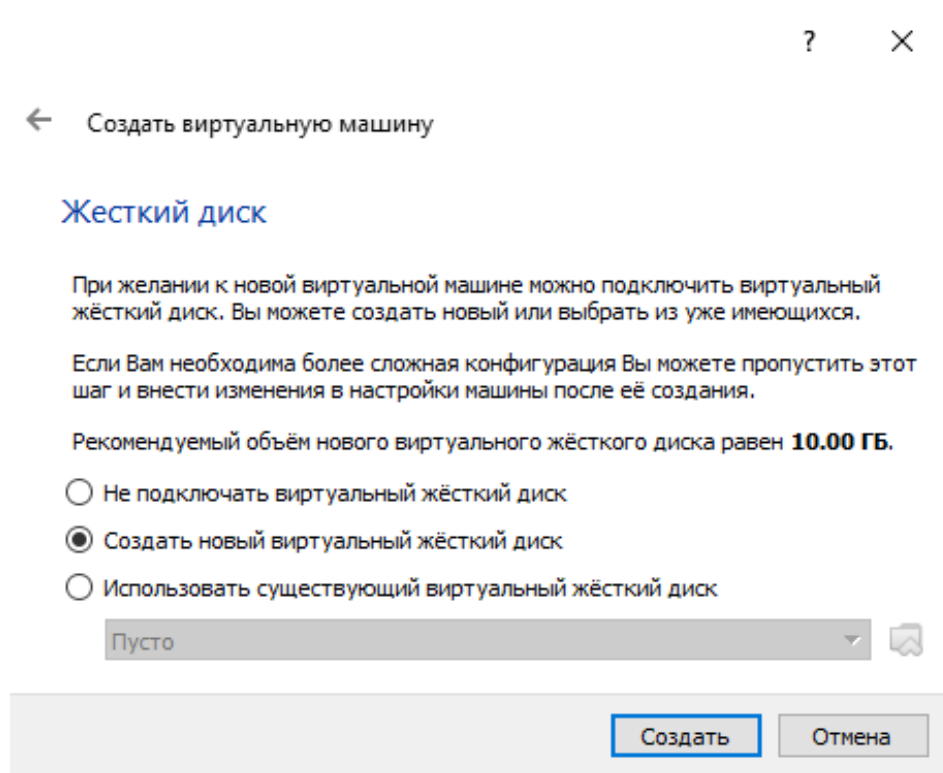
В окне программы нажимаем «Создать».



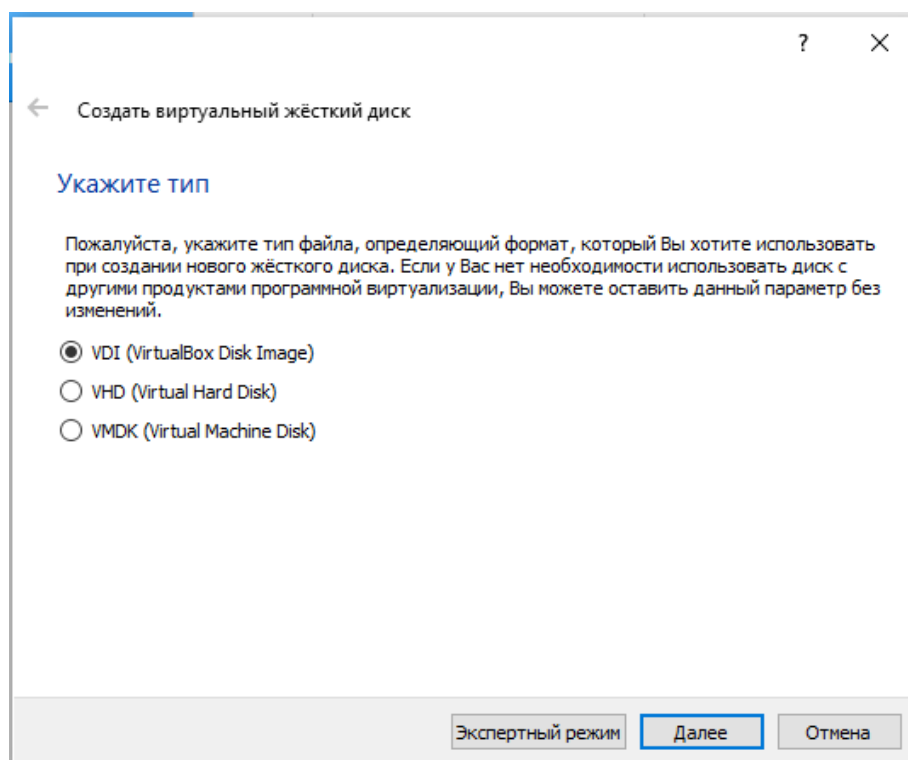
Пропишите имя, тип и версию. Например, VM_Ubuntu-32. По нему вы будете идентифицировать систему. Поэтому создавайте его информативным. Будем устанавливать 32-битную версию Ubuntu desktop. Следовательно, версия: Ubuntu (32-бит).



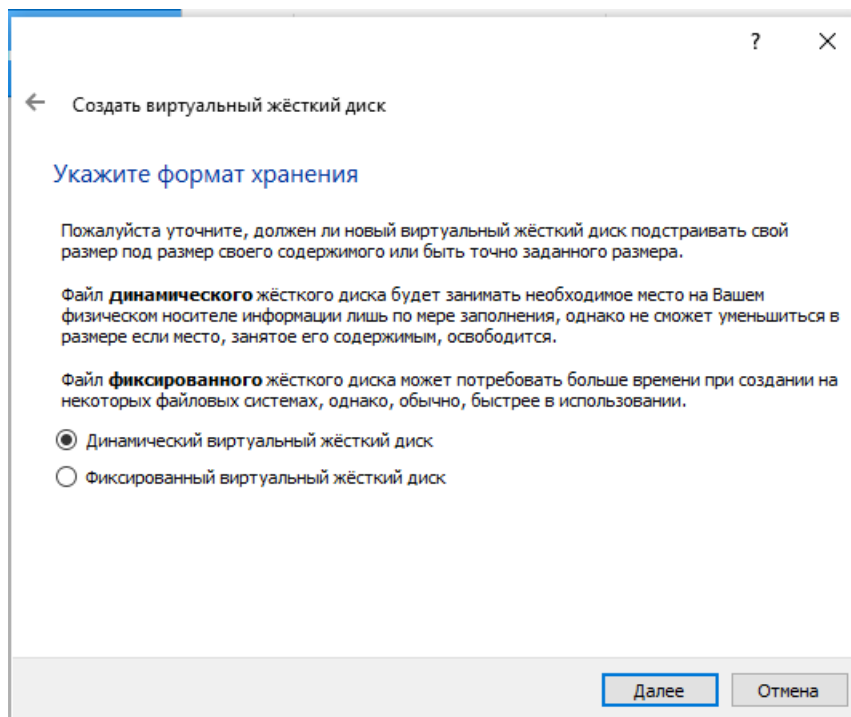
Нажимаем «Далее». Пропишите объем ОЗУ, который вы отдадите виртуальной машине. Например, хотя бы 1 ГБ. Укажите объем, перемещая специальный ползунок, или пропишите вручную. После выбора нажимаем «Далее».



Создаем жесткий диск. Если рассматривать его физически, это файл, хранящийся на одном из разделов на вашем HDD. Программа предлагает три варианта. Выбираем 2-й вариант: создать новый виртуальный жесткий диск.

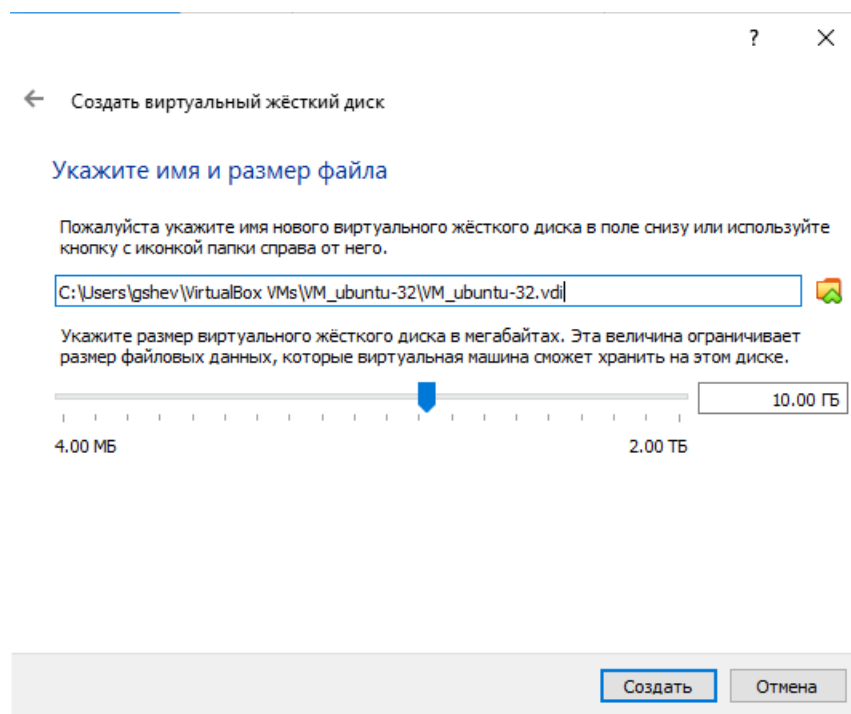


Нажимаем «Создать». Определяемся с типом виртуального HDD. Выбираем тип VDI, т.е. первый вариант. Затем выбираем формат хранения: динамический виртуальный жесткий диск.



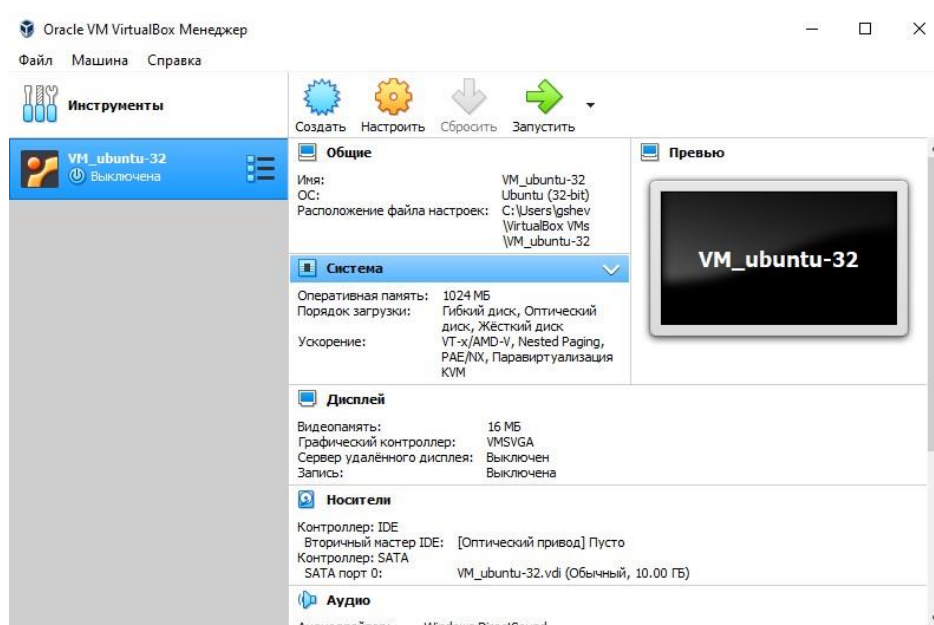
Укажите имя и объем виртуального HDD, например, VM_ubuntu-32. При необходимости укажите другое место хранения диска. Для этого надо кликнуть кнопку справа от поля ввода.

На шкале укажите объем виртуального жесткого диска в мегабайтах. Для этого перетащите ползунок на нужное место, соответствующее вашим потребностям, например, 10 ГБ. После выбора настроек, нажмите на кнопку «Создать».



Виртуальная машина создана. После этого откроется главное окно «Oracle VM

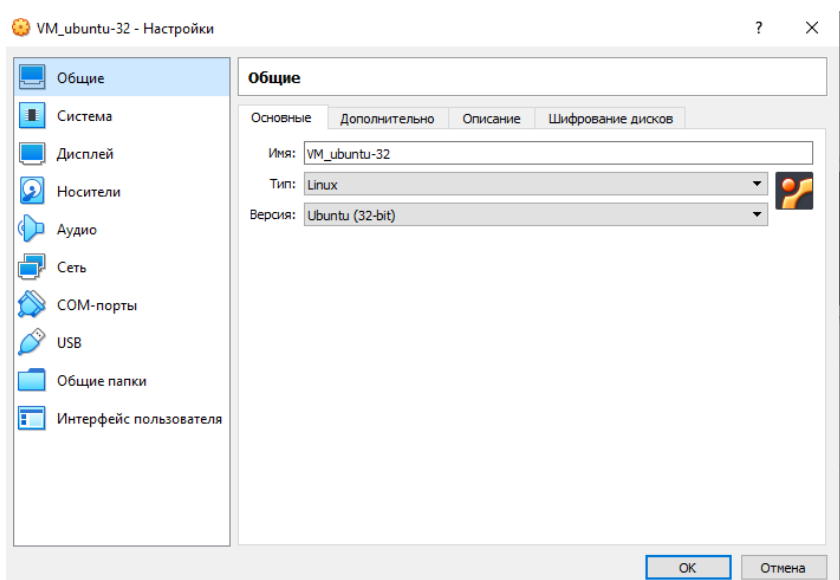
VirtualBox Менеджер» с вновь созданной виртуальной машиной. В правой части окна вы можете ознакомиться с некоторыми параметрами виртуальной машины.



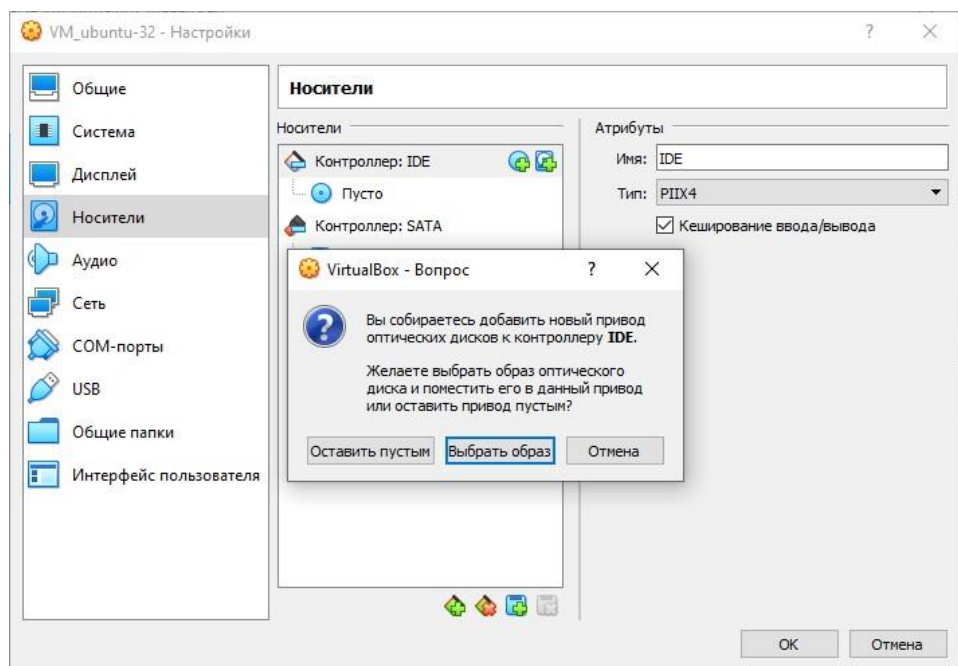
Теперь необходимо будет сделать дополнительные настройки перед установкой операционной системы Ubuntu на виртуальную машину.

В главном окне VirtualBox нажмите на кнопку «Настроить» для входа в настройки этой виртуальной машины:

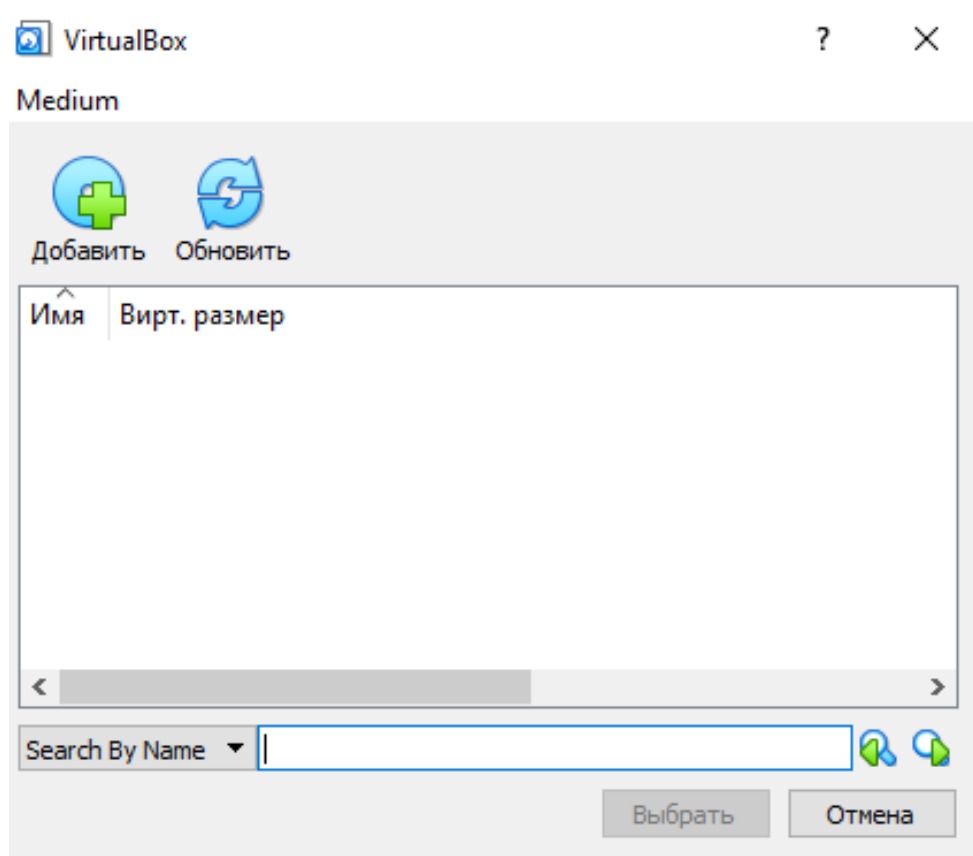
1. Для установки операционной системы на виртуальной машине потребуется загрузка с установочного диска. В среде VirtualBox имеется возможность выполнения загрузки с использованием виртуального привода, создаваемого на основе образа загрузочного диска. При первом запуске виртуальной машины, когда еще нет установленной гостевой операционной системы, VirtualBox предложит выбрать устройство загрузки.



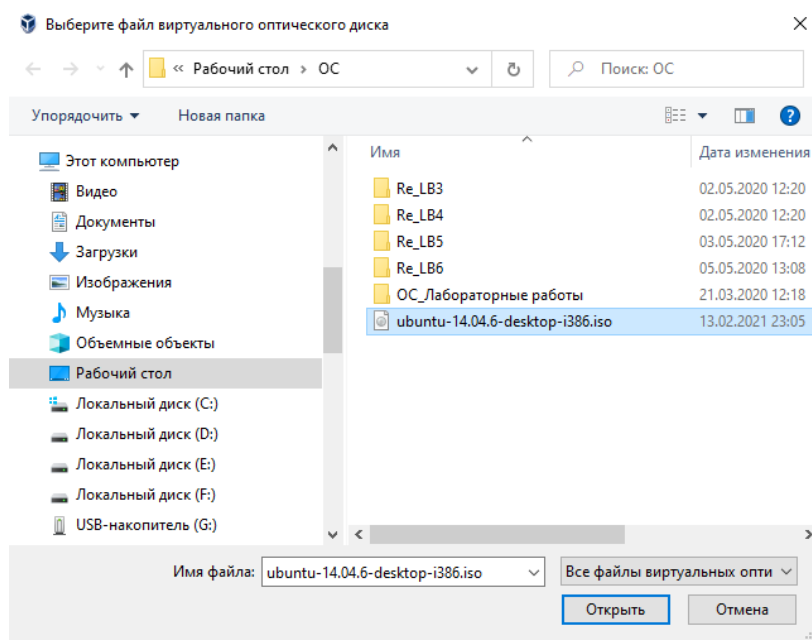
Выбрать пункт «Носители»



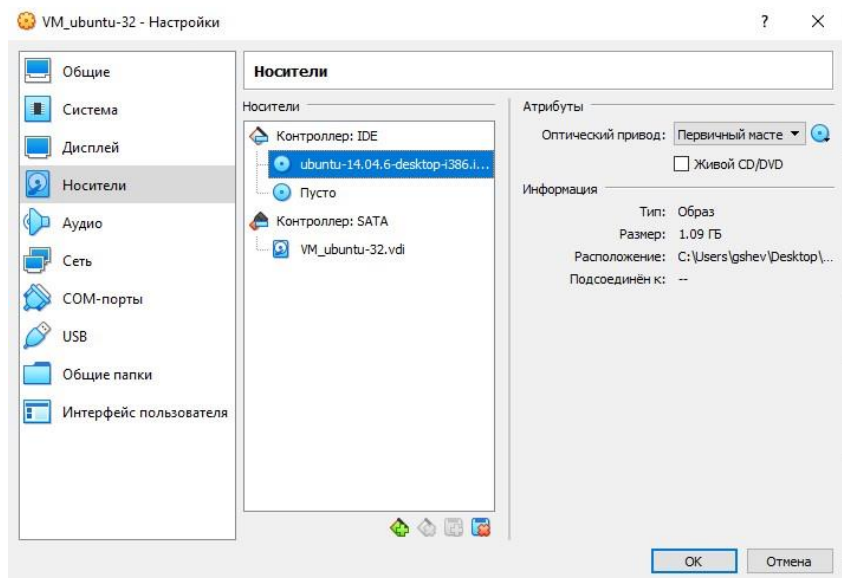
Добавить привод оптического диска (нажать на 1-ю кнопку справа от контроллера: IDE). Затем «Выбрать образ»



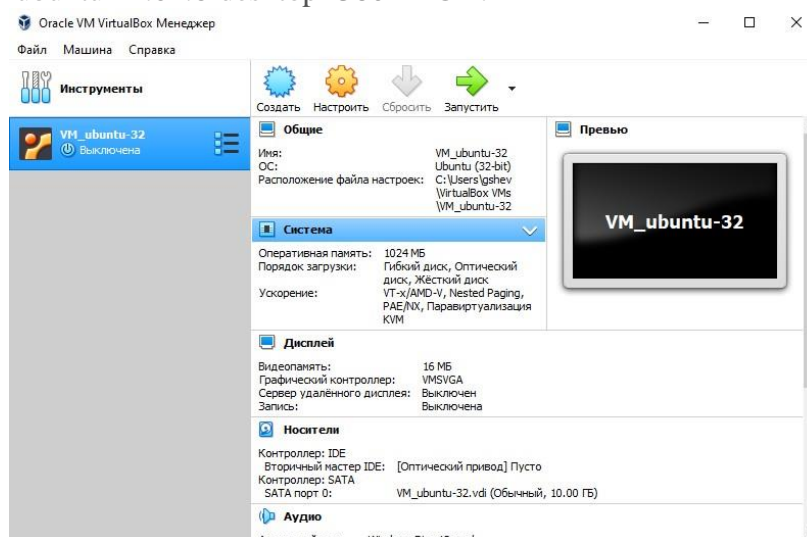
Выбрать кнопку «Добавить».
Указать путь на своем компьютере к образу Ubuntu.



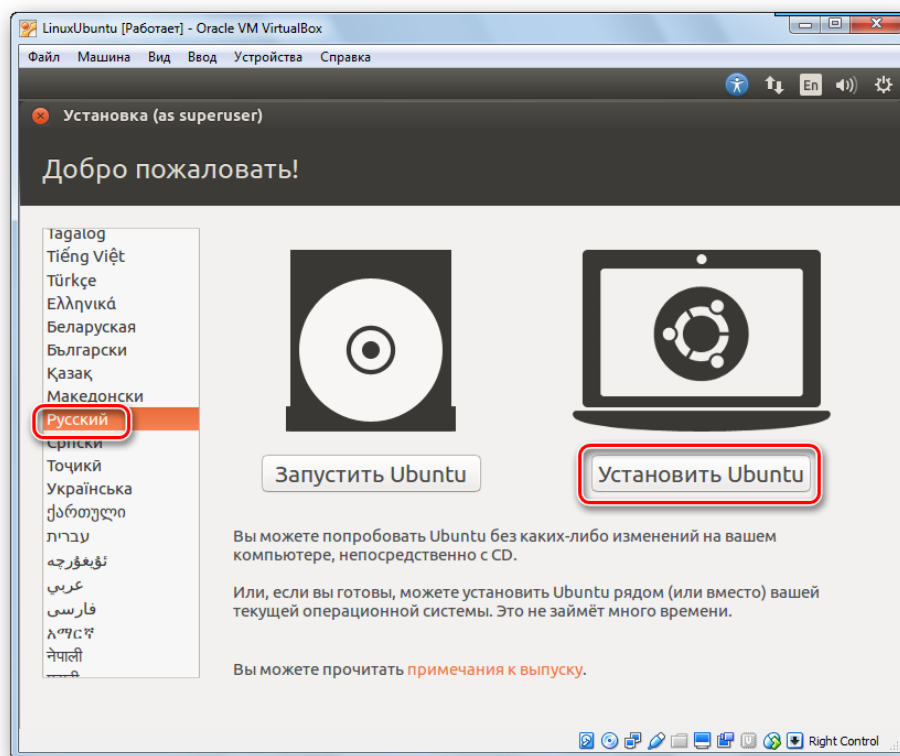
«Открыть»



Выбрать ubuntu 14.04.6-desktop-i386... ОК.



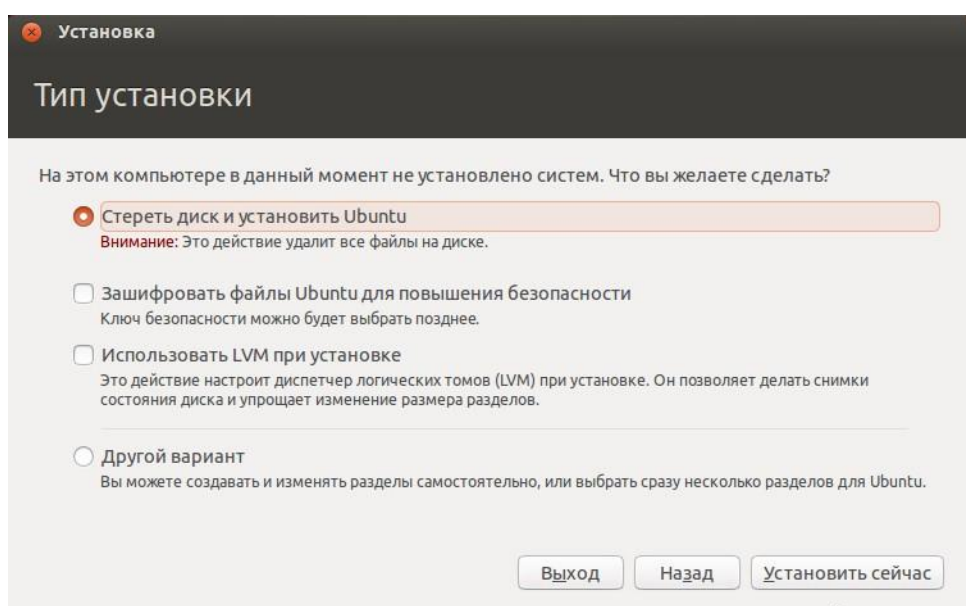
«Запустить» Oracle VM VirtualBox



Выберите язык из списка в левой части окна. Нажмите «Установить Ubuntu».

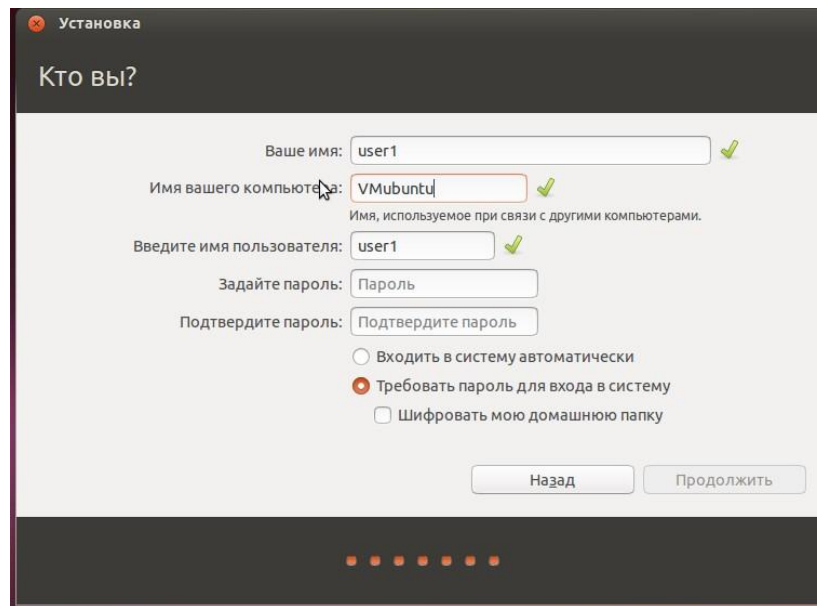
Начнется загрузка ОС.

Процесс установки гостевой ОС ничем не отличается от установки на реальной машине. Можно выбрать язык для устанавливаемой системы (обычно Русский), часовой пояс, раскладку клавиатуры и т. п. Большинство параметров можно оставить по умолчанию, в том числе и Тип установки.



Установщик Linux предостерегает вас от ошибочных действий. Ознакомьтесь с представленной вам информацией и смело жмите «Установить сейчас».

В процессе установки необходимо задать имя компьютера, пользователя, пароль и режим входа в систему (лучше выбрать «Входить в систему автоматически»).

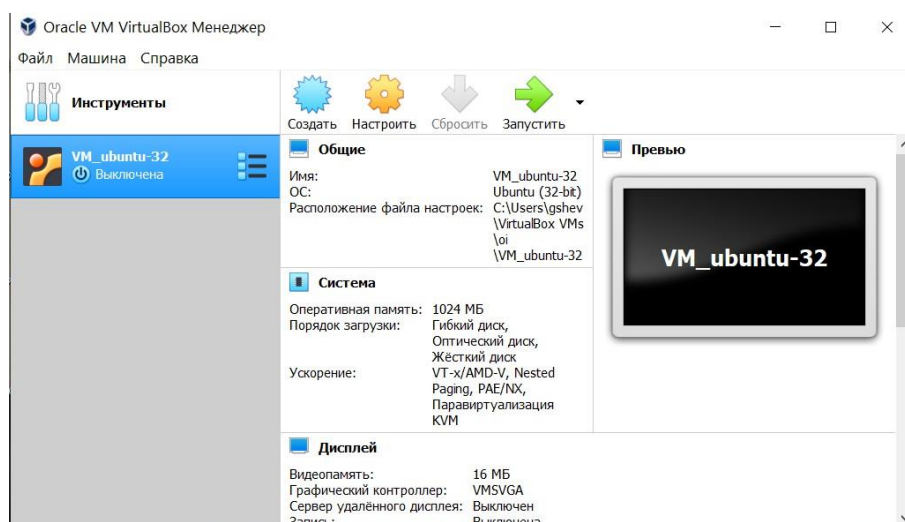


Дальнейшая установка Ubuntu выполняется без какого-либо вмешательства пользователя и завершается предложением перезагрузить компьютер. По сравнению с установкой системы на реальном компьютерном оборудовании, установка на виртуальной машине выполняется медленнее, что вполне ожидаемо. Степень снижения производительности в основном, зависит от быстродействия оборудования реального компьютера.

При первой загрузке вновь установленной операционной системы, диспетчер VirtualBox автоматически отключит виртуальный привод на основе образа диска с дистрибутивом Ubuntu, загрузка будет выполнена с виртуального жесткого диска и по ее завершению, на экране отобразится приглашение ко входу в систему.

2. Настроим VirtualBox так, что в ней можно будет работать с флешкой.

Подключаем флешку (надо знать ее имя) к компьютеру и запускаем VirtualBox, затем жмём на кнопку «Настроить».

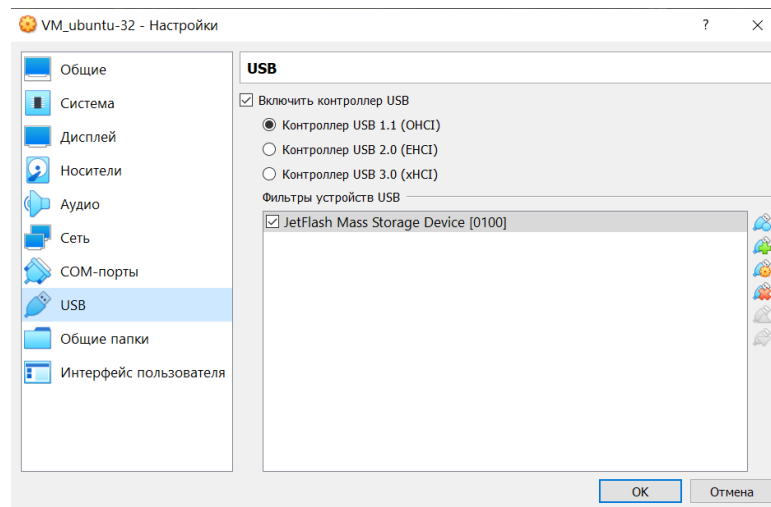


Выбираем Категорию USB.

Отмечаем пункты Включить контроллер USB

Включить контроллер USB 1.1 (OHCI)

В окне Фильтры устройств USB жмём на + (справа второй сверху) и выбираем нашу флешку (в данном случае JetFlash...), отмечаем её.



3. Практическое знакомство с ОС Ubuntu-desktop x32

Начнём знакомство с Ubuntu с описания пользовательского интерфейса.

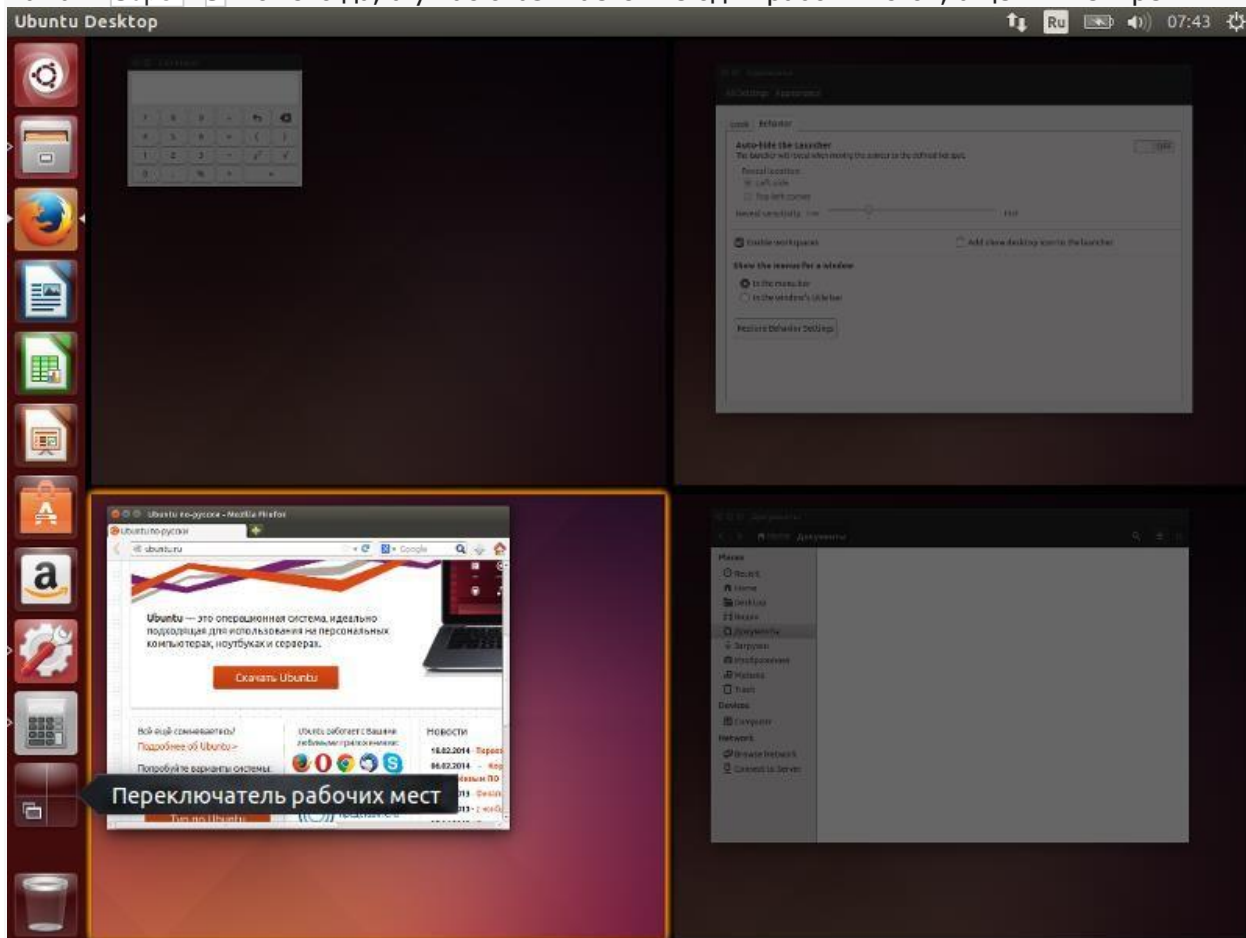
В дистрибутивах Linux (таких, как Ubuntu) используется несколько графических сред рабочего стола, самые распространённые из которых [GNOME](#), [KDE](#), [Xfce](#)¹⁾, [LXDE](#). Так вот, в Ubuntu среда рабочего стола — GNOME, но среда очень хитрая, — сверху над ней находится оболочка (по-английски — Shell) под названием **Unity**.

Unity

Если вы работали с другими операционными системами, то вы легко разберётесь в интерфейсе Ubuntu. Поэтому остановимся, в основном, только на особенностях графической оболочки Unity. Итак, вы только что ввели свой пароль на экране входа и вот вы в Ubuntu.

Рабочий стол

Ну стол как стол, что в нём особенного? Обои на месте, мышь можно подвигать, окна пооткрывать – всё как в других операционных системах. А вот не торопитесь, попробуйте нажать **Super+S**. Вот это да, а у вас оказывается не один рабочий стол, а целых четыре!



Чтобы функция работала, нужно включить опцию «задействовать рабочие места», которая по умолчанию отключена. Включается в меню «Все параметры → Оформление → Режим → Задействовать рабочие места».

Эти виртуальные рабочие столы — рабочие места, с их помощью можно очень эффективно использовать рабочее пространство. Например, можно собрать все окна коммуникационных приложений, таких как электронная почта и чаты, на одном рабочем месте, а выполняемую работу – на другом. На третьем рабочем месте может быть ваш музыкальный проигрыватель.

Конечно, как и в любой другой операционной системе, на рабочем столе можно создавать файлы и папки, перемещать существующие элементы в удобные для вас места, менять обои, размер значков, и т. п. Можно также настроить, чтобы на рабочем столе отображались значки подключённых устройств, домашняя папка, корзина.

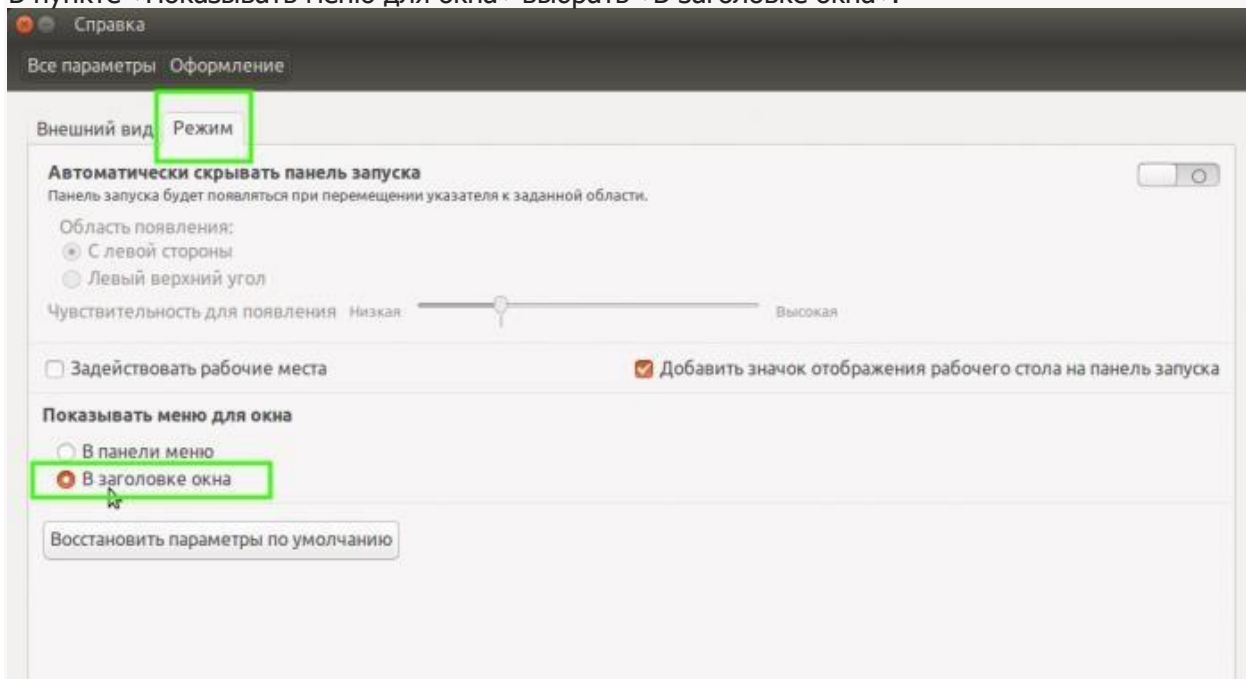
Да, кстати. Как вы уже, наверное, заметили, очень многие действия с интерфейсом проще и быстрее выполнять с помощью различных комбинаций клавиш. Как же их запомнить? Нажмите и подержите пару секунд клавишу **Super**. Запомнили?

Панель меню

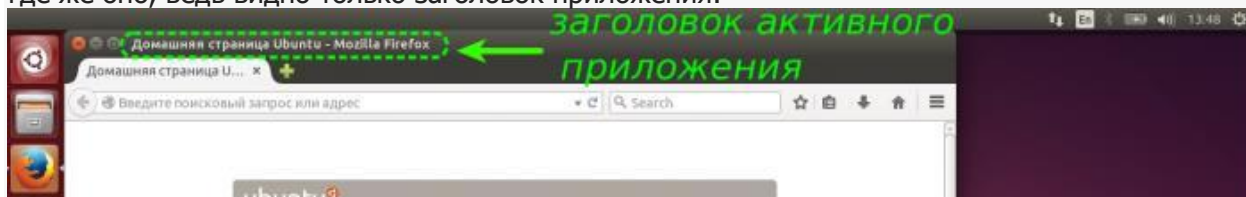
Вы уже знаете, что тёмная полоска вверху экрана называется панелью меню. На ней располагаются кнопки управления окном, меню приложения и меню состояния:

Кнопки управления окном находятся в верхнем левом углу окна. А вот когда окно развёрнуто на весь экран, то кнопки располагаются в верхнем левом углу экрана. Наверное, вы и так знаете, что кнопки нужны для закрытия, сворачивания, разворачивания и восстановления размера окна.


Стоит отметить, что в Ubuntu 14.04 был добавлен новый опциональный режим размещения меню и кнопок управления в локальных окнах приложений при котором меню размещено в заголовке окна. Включается он просто. Нужно кликнуть ПКМ в свободном месте рабочего стола выбрать из контекстного меню «Изменить фон рабочего стола» и перейти на вкладку «Режим». В пункте «Показывать меню для окна» выбрать «В заголовке окна».



Меню приложения находится справа от кнопок управления окном. Подождите-подождите, а где же оно, ведь видно только заголовки приложения:










Дело в том, что Unity скрывает меню приложения и кнопки управления окном до тех пор, пока не переместить указатель мыши в верхнюю левую часть экрана или не нажать **Alt+F10** в случае режима по умолчанию (глобального меню). Если активирован режим размещения меню в локальных окнах, то, чтобы увидеть это меню, просто переместите указатель мыши к заголовку



меню приложения

Меню состояния — так сейчас принято называть старую добрую область уведомлений из Gnome 2. А ещё эти меню часто называют просто индикаторами. Они расположены в правой части панели и очень удобны для проверки и изменения состояния как вашего компьютера, так и различных приложений. Давайте пробежимся по основным из этих меню и покажем, что они умеют:

- **Индикатор сети**  — неизвестный вам [NetworkManager](#): с его помощью очень просто подключаться практически к любым сетям.
- **Индикатор раскладки**  — кажется, статье выше что-то говорилось про [локализацию](#), хотя, попробуйте навести указатель мыши на индикатор и покрутить колёсиком.
- **Меню Bluetooth**  — с его помощью можно отправлять и получать файлы по Bluetooth, разумеется, если устройство Bluetooth обнаружено.
- **Меню сообщений**  в нём собрано всё, что нужно для быстрого запуска и получения входящих уведомлений от электронной почты, социальных сетей и чатов.
- **Индикатор батареи**  он показывает состояние заряда вашего аккумулятора. Понятно, что если аккумулятора нет, то и индикатора не будет.
- **Звуковое меню**  — настройка звука и управление медиапроигрывателем. Попробуйте навести на индикатор указатель мыши и покрутить колёсиком — посмотрите, что произойдёт.
- **Часы** — ну, тут всё понятно из названия, кроме одного: здесь могут отображаться события из календаря Evolution.
- **Системное меню**  пожалуй, самое важное меню. Мало того, что оно расскажет о вашем компьютере, выведет справку и позволит настроить систему, так оно ещё и позволит переключать пользователей, блокировать экран, завершать сеанс, переходить в ждущий режим, выключать и перезагружать компьютер! Крутое меню!

Большинство меню состояний изменяют свой значок в зависимости от состояния компьютера или приложения, наверное, поэтому их и называют индикаторами. Кстати, очень многие приложения сами добавляют своё меню на панель, например, BitTorrent-клиент Transmission, да тот же Skype, в конце концов.

Панель запуска

Вот вы и подобрались к главной особенности Unity, можно даже сказать, ключевому элементу интерфейса – панели запуска (Launcher по-английски), она всегда появляется вдоль левой стороны экрана после входа в систему. С её помощью в системе организован быстрый доступ к приложениям, рабочим местам, съёмным носителям, корзине. Потренируемся?

Щёлкните значок домашней папки, кажется, это второй значок сверху – откроется окно файлового менеджера **Nautilus**. А теперь присмотритесь, как реагирует значок на запуск файлового менеджера: правильно, значок пульсирует – так Ubuntu сообщает, что приложение начинает работать. Как только домашняя папка откроется, значок наполнится цветом, а слева от его рамки появится маленький белый треугольник. Два треугольника означают, что открыто два окна этого приложения, а если три треугольника – то три окна или более. Что, и справа

треугольник появился, да ещё в другую сторону смотрит? Не волнуйтесь, просто окно приложения в настоящий момент активно.

А теперь поговорим о самой важной кнопке на панели запуска — о кнопке с логотипом Ubuntu:

Главное меню

Это кнопка запуска главного меню (Dash – приборная панель по-английски) — важнейший элемент интерфейса Unity.

Многим, пользовавшимся Windows или MacOS, главное меню покажется аналогом кнопки «Пуск» в Windows или «Launchpad» в MacOS. Это не совсем так. Общее в них только одно – с помощью этих кнопок пользователь действительно получает быстрый доступ к приложениям и файлам на компьютере. А дальше начинаются различия. И самое главное из них — это принцип действия, механизм, если хотите, работы главного меню.

Это мощнейший поисковый инструмент, тщательно продуманный и организованный. Что такое поиск? Очень сложный процесс, основанный на анализе огромного количества поисковых запросов и выдаче релевантных ответов. Почему сложный? Задумайтесь хотя бы над двумя вопросами: что мы ищем и где. Первый — что. Контент? А какой: текстовый, графический, мультимедийный, а может быть, сообщения из социальных сетей или прогноз погоды? Приложения — опять же, какие: предназначенные для работы (работы над чем и с чем?) или для отдыха (для какого отдыха: музыку послушать или в игру сыграть, а может, в магазин сходить?), недавно использовавшиеся, установленные на локальном компьютере или доступные для загрузки из сети? Такой же непростой ответ будет и на второй вопрос, ведь область поиска практически ничем не ограничена.

Так вот, в главном меню этот механизм реализован очень элегантно и эффективно. Нажмите на клавишу **Super** или на самую верхнюю кнопку панели запуска — на кнопку с логотипом Ubuntu. Откроется полупрозрачное окно – это и есть главное меню:



Начнём обзор главного меню с нижнего ряда значков – это линзы (*Lenses* по-английски), основные крупные элементы в глобальном механизме поиска, — так Unity группирует наиболее общие поисковые результаты. Линз по умолчанию семь¹²⁾. Главное меню всегда открывается на **домашней** линзе – в ней собираются результаты поиска из других линз: **приложений, файлов, видео, музыки, фотографий** и области¹³⁾ **друзей**. Понятно, что линзы сужают («фокусируют») выдачу поисковых результатов в соответствии со своими названиями, так что останавливаться подробно на каждой линзе большого смысла нет. Переключаться между линзами можно или мышкой, или с помощью **Ctrl+Tab**.

Домашняя линза сразу показывает недавно использовавшиеся приложения и файлы. И для приложений, и для файлов отображается только один ряд результатов. Если результатов больше, то для их просмотра можно щёлкнуть «Показать ещё ... результатов». А теперь начните печатать в строке поиска что-нибудь, имеющее отношение к вашему запросу, и совершенно не важно, что вы ищете: приложение, файл или папку, — подходящие результаты поиска из других установленных линз будут появляться автоматически. Нашли что искали? Щёлкайте мышью на результате, а если он первый в ряду, то нажимайте **Enter**.

Нет, всё-таки вернёмся к линзам, и вот с какой целью. Во всех линзах есть система очень удобных фильтров, каждый элемент которых – это своеобразное узкоспециальное¹⁴⁾ направление поиска. Эти элементы называются областями или сферами, они могут быть как встроенными в систему, обрабатывающими в основном локальные запросы, так и внешними, анонимно¹⁵⁾ направляющими ваш поисковый запрос на удалённый сервер для обработки. Их единственная цель — как можно более точно предложить вам релевантный ответ. Активировать их, или наоборот выключить, можно с помощью мыши, нажав на соответствующую кнопку. Кстати, многие называют области поисковыми плагинами – разумеется, они правы, но что тогда делать с областью «Поисковые плагины» в линзе приложений?

Ну как, похоже главное меню на кнопку «Пуск» в Windows? Ответ очевиден.

И в заключение, как всегда, о комбинациях клавиш – выбрать сразу нужную линзу можно с помощью клавиши **Super** в комбинации с: **A (Application)** для выбора линзы приложений, **F** для файлов, **V** для видео, **M** для музыки, **C (Camera)** для фотографий и **G (Gwibber)** для области друзей. Вы, наверное, уже забыли, что простое нажатие **Super** вызывает главное меню на домашней линзе.

4. Компиляция программ на C/C++ в терминале Ubuntu

Имеется текст программы с русскими символами на языке C, записанный в формате Word (кодировка Windows – 1251) в файл hello.docx.

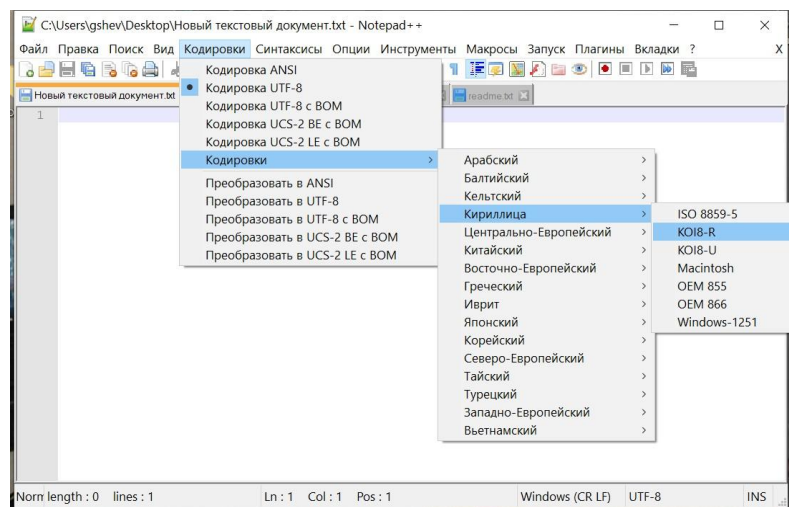
```
#include <stdio.h>

int main(int argc, char **argv)
{
    puts("Привет, мир!");
    return 0;
}
```

Этот файл надо перекодировать в KOI8-R, записать с расширением hello.c на флешку в хостовом компьютере, затем запустить виртуальную машину с настроенной флешкой и скопировать файл hello.c с этой флешки на рабочий стол терминала Ubuntu. Если в меню нет иконки Terminal, то надо нажать кнопку запуска главного меню Dash (1-я сверху) и в открывшемся окне набрать слово Terminal, появившуюся иконку перетащить в меню. Открыть терминал можно также комбинацией клавиш «Ctrl+Alt+T». Для вызова Рабочий стол нажать кнопку Nautilus (2-я сверху). Появится список команд.

- Работаем в хостовой машине с ОС Windows.

Запускаем текстовый редактор Notepad++ (он бесплатный и есть в интернете). Надо заменить кодировку в редакторе.



После смены кодировки надо сбросить в этот редактор текст из файла hello.docx и записать его на флешку с расширением hello.c.

- Работаем в виртуальной машине с ОС Ubuntu.



Находим иконку доступа к флешке с записанным в ней файлом hello.c, и файл hello.c сбрасываем на рабочий стол терминала Ubuntu.

Теперь с помощью горячих клавиш <ctrl> + <Alt> + <T> запускаем терминал. Здесь в начале установим инструменты сборки, куда входят необходимые компиляторы gcc для языка C и g++ для языка C++:

```
sudo apt install build-essential
```

Для установки требуется ввести пароль, при вводе которого может сложиться впечатление, что ничего не происходит, но на самом деле терминал просто в целях безопасности не отображает символы. Далее в терминале нам необходимо перейти в директорию, куда сохранили файл с текстом программы. Перемещение выполняется командой cd (англ. change directory – изменить каталог). Чтобы воспользоваться командой в начале пишется cd, затем через пробел путь, куда нужно перейти. Для перехода на рабочий стол, команда будет следующей:

```
cd Рабочий\ стол
```

Обратите внимание на символ обратной косой черты \ в имени директории Рабочий стол. Обратная косая экранирует пробел, и сообщает команде cd, что пробел и следующие за ним символы являются частью имени. Для просмотра содержимого директории применяется команда ls (сокращение от англ. list).

Команда компиляции для программы на C выглядит следующим образом:

```
gcc -pthread -Wall -o hello hello.c
```

где:

gcc – компилятор для языка программирования C;

-Wall – ключ вывода всех предупреждений компилятора;

-o hello – с помощью ключа -o указывается имя выходного файла;

hello.c – имя нашего исходного файла, который компилируем.

Выполнив команду ls, увидим, что появилась наша скомпилированная программа hello, отмеченная цветом, обозначающим исполняемые программы. В завершение запустим hello, вводом имени программы с префиксом ./:

```
./hello
```

Префикс ./ сообщает терминалу о необходимости выполнить программу с заданным именем в текущем каталоге. (Точка – это условное название текущего каталога.)

Программы на C++ компилируются аналогично, как и программы на C.’

```
#include <iostream>
```

```
int main(int argc, char **argv)
{
    std::cout << "Привет, мир!" << std::endl;
    return 0;
}
```

Сохраняем текст программы в KOI8-R в файл под именем hello1.cpp. Команда компилирования будет иметь вид:

```
g++ -pthread -Wall -o hello1 hello.cpp
```

Для запуска результата вводим в терминале:

```
./hello1
```

Контрольные вопросы

1. Что такое виртуальная машина?
2. Зачем нужна виртуальная машина и виртуализация?
3. Преимущества и недостатки использования виртуальной машины.
4. Популярные виртуальные машины.
5. Как создать виртуальную машину VirtualBox?
6. Как установить операционную систему в VirtualBox?
7. Какие возможны настройки Ubuntu-desktop после установки?
8. Как запускать программы C/C++ в ubuntu-desktop?