ТЕХНИЧЕСКОЕ ОПИСАНИЕ КОМПЕТЕНЦИИ ОБЛАЧНОГО ПРОГРАММИРОВАНИЯ

Главный эксперт **Галипанова А.С..**

1. Введение
1.1.Наименование и описание компетенции
1.2.Наименование компетенции
2. Программирование в "облаке"
2.1."Облачные" сервисы хранения данных
3. Описание компетенции

1. Введение.

В облачных вычислениях выделяют следующие ключевые характеристики:

- **Самообслуживание по требованию.** Потребитель самостоятельно выбирает, каким набором вычислительных возможностей и ресурсов он будет пользоваться (например, сетевые хранилища, базы данных, процессорное время, объем оперативной памяти). Также потребитель может при необходимости изменять этот набор без согласования с провайдером в автоматическом режиме.
- **Высокая эластичность (гибкость) сервисов.** Вычислительную мощность можно легко уменьшить или увеличить, исходя из потребностей пользователя. В случае высокой нагрузки на сервис количество ресурсов оперативно повышается, в случае уменьшения нагрузки – ресурсы освобождаются. Если образовательному учреждению потребуется срочно увеличить объем вычислительных ресурсов, то руководству учреждения не придется тратить средства и время на закупку и настройку дополнительного оборудования и программного обеспечения, которое впоследствии может использоваться достаточно редко.
- **Возможность объединение ресурсов.** Вычислительные ресурсы "облачного" провайдера группируются в пулы с возможностью динамического перераспределения физических и виртуальных ресурсов между конечными потребителями. С применением современных технологий виртуализации это позволяет "облачному" провайдеру легко наращивать мощности и заменять вышедшее из строя оборудование без снижения уровня производительности и надежности.
- **Учет потребления ресурсов и оплата по факту использования.** Потребители платят только за фактически потребленные услуги (например, за объем переданной информации, пропускную способность и т.д.).
- **Технологичность.** Можно смело утверждать, что в дата-центрах поставщиков облачных услуг используются более современные инновационные технологии, чем в большинстве учебных заведений. Эти технологии позволяют автоматически оптимизировать использование вычислительных ресурсов и сократить издержки на обслуживание оборудования по сравнению аналогичными издержками в учебных заведениях.
- **Отказоустойчивость и высокий уровень доступности.** Дата-центры для облачных вычислений представляют собой надежную распределенную сеть, узлы которой могут располагаться в различных уголках мира. Отказоустойчивость у такой сети как правило заведомо выше любой пользовательской локальной сети, т.к. обеспечивается многократным резервированием и квалифицированным обслуживанием технического персонала. В итоге, такая распределенная сеть позволяет получить услуги с высоким уровнем доступности. Позволить себе организовать подобную сеть дата-центров может далеко не каждое образовательное учреждение. Кроме того, дата-центры как правило строят вблизи дешевых источников электроэнергии, что является экономически более целесообразным, чем поддержание работоспособности ИТ-инфраструктуры при работе по обычным для небольших потребителей тарифам на электроэнергию.

1.1. Классификация облачных вычислений

В облачных вычислениях традиционно выделяют три типа (уровня):

- Инфраструктура как услуга.
- Платформа как услуга.
- Программное обеспечение как услуга.

Рассмотрим подробно каждый из этих типов, т.к. каждый из них имеет свою целевую аудиторию и цели, о которых нужно иметь чёткое *представление* при переходе с традиционной парадигмы организации вычислений на "облачную".

На рис. 1.1 в виде обобщенной схемы представлен каждый из перечисленных выше видов облачных услуг, названия которых приведены в центре в виде перевернутой пирамиды. Больший размер блока пирамиды означает, что он включает в себя всю инфраструктуру более маленького блока. Например, для предоставления сервиса "Платформа как услуга" с точки зрения поставщика услуг необходимо также иметь возможность обеспечить сервис "Инфрастуктура как услуга".

В левой части рисунка в указано, каким видом ИТ-ресурсов необходимо обладать, чтобы предоставить соответствующие услуги. В правой части рисунка перечислены виды целевой аудитории предоставляемых облачных услуг.

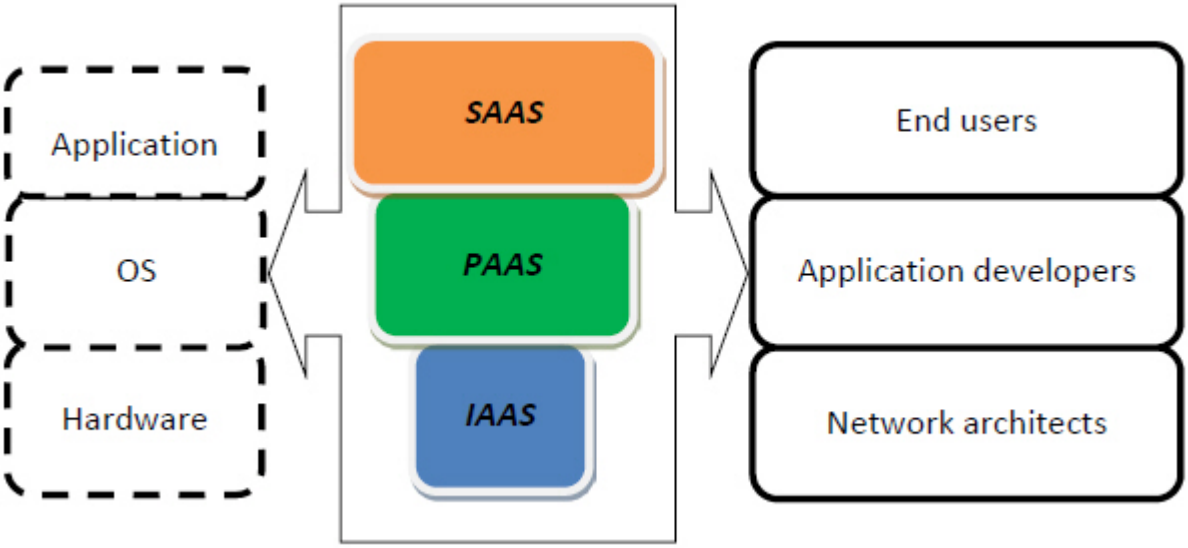


Рис. 1.1. Виды "облачных" услуг

Инфраструктура как услуга (IaaS, *Infrastructure as* a *Service*). На этом уровне потребитель может самостоятельно конструировать свою IT-инфраструктуру в облаке и управлять ей. Например, создавать виртуальные сети, добавлять *виртуальное* оборудование (серверы, хранилища, *базы данных*), устанавливать необходимое для работы прикладное *программное обеспечение* и операционные системы, т.е. использовать облако так, как если бы это была реальная IT-*инфраструктура* образовательного учреждения. Самые известные IaaS-решения: Amazon CloudFormation, Google *Compute Engine*, *Windows* Azure.

Платформа как услуга (PaaS, *Platform as* a *Service*). На этом уровне *провайдер* облачных услуг предоставляет пользователю *доступ* к операционным системам, системам управления базами данными, средствам разработки и тестирования. Таким образом, потребитель облачных услуг получает возможность и средства для самостоятельного создания, тестирования и эксплуатации программного обеспечения. При этом вся

информационная *инфраструктура* (вычислительные сети, серверы и системы хранения) управляется провайдером. Вот перечень наиболее известных PaaS-сервисов:

- Google App Engine (для разработки программного обеспечения на языках Java, Python);
- Windows Azure (для ASP.NET, PHP);
- Cloud Foundry (языки программирования Java, Ruby, Scala).

Программное обеспечение как услуга (SaaS, *software as* a *service*). На этом уровне поставщик предоставляет пользователям облака готовое *программное обеспечение*. Все данные хранятся в облаке, и для доступа к ним пользователю требуется только наличие веб-браузера. Это наиболее интересный для образовательных учреждений тип облачных вычислений, поскольку он не требует дополнительных затрат на установку и настройку программного обеспечения, как это требуется при использовании IaaS и PaaS. Следует также иметь в виду, что в большинстве случае *плата* за использование программного обеспечения в рамках SaaS рассчитывается с учётом количества пользователей и не предполагает так называемых Enterprise-лицензий, позволяющих использовать некоторый сервис для любого количества пользователей без ограничений. Примеры бесплатных SaaS-решений для образовательных учреждений – это Google Apps for *Education* и Microsoft Office 365 for *education*. Они содержат в себе функции офисного пакета (работа с документами, таблицами и презентациями), средств коммуникации (*электронная почта*, календари, мгновенные сообщения) и средств эффектной подачи информации (в виде статических презентаций, видеороликов или интерактивных приложений).

1.2. Отличие облачных вычислений от Web2.0

Некоторые сотрудники сферы образования часто путают облачные вычисления с технологиями Веб 2.0, ошибочно полагая, что облачные вычисления – это любые сервисы, предоставляемые с помощью *Интернет*.

Перечислим типичные приложения Веб 2.0:

- онлайн-энциклопедии (например, http://www.wikipedia.org);
- блоги (например, http://www.livejournal.com);
- каналы RSS для рассылки дайджеста новостей с возможностью отслеживать обратную связь с читателями по количеству переходов на полную версию той или иной новости;
- сервисы mash-up, использующие в качестве источников информации другие сервисы (например, сервис поиска магазинов, который использует сторонний сервис для отображения найденных магазинов на карте);
- метки tags, позволяющие выявить наиболее популярные материалы среди пользователей на данный момент;
- медиа-библиотеки, формируемые участниками в режиме онлайн;
- социальные сети (например, http://www.vk.com);

Ключевой особенностью всех этих технологий является возможность онлайн-редактирования содержимого веб-страниц их посетителями. При этом все приложения Веб 2.0 могут быть размещены как в облаке, так и в локальной IT-инфраструктуре использующего их учреждения. Таким образом, главное отличие облачных вычислений от Веб 2.0 заключается в том, что приложения Веб 2.0 – это только определенный вид программного обеспечения, тогда как облачные вычисления – это метод хранения данных и предоставления их конечному пользователю.

1.3. Применение облачных вычислений в образовании

Одной из первых облачных услуг, которую стали использовать европейские образовательные учреждения, стала *электронная почта*. Обеспечение работоспособности (*аутсорсинг*) сервиса электронной почты – несложная задача, которая определенно не играет ключевой роли в работе образовательного учреждения. Корпорации Google и Microsoft предоставляют сотрудникам и учащимся образовательных учреждений *доступ* к электронной почте бесплатно.

Помимо услуг электронной почты эти корпорации обеспечивают возможность использовать в облаке функции стандартного офисного пакета для совместной работы с электронными документами, таблицами и для создания презентаций. Облачные сервисы для образовательных организаций Google Apps for *Education* и Microsoft Office 365 for *education* позволяют использовать встроенные системы для обмена мгновенными сообщениями, календари для совместного планирования и общие адресные книги. Каждый *пользователь* облачных систем получает значительное дисковое *пространство* для хранения любой информации, которая была получена в результате работы с облаком.

Может показаться странным, что эти услуги предоставляются образовательным учреждениям бесплатно, в то время как для коммерческих организаций цены на программное обеспечения как были, так и остаются традиционно высокими. Такая ценовая политика объясняется следующим образом. На современном рынке облачных технологий сохраняется высокая конкуренция между поставщиками программного обеспечения, поэтому они стараются предоставлять свои сервисы образовательным учреждениям бесплатно. Расчет идет на будущих выпускников, которые после получения образования устроятся на работу и смогут убедить будущих работодателей приобрести *программный продукт*, о преимуществах которого они уже знают. Также это обеспечит привязанность и лояльность пользователей к продуктам определенной марки и её узнаваемость и популярность.

Если для образовательного учреждения *безопасность* доступа к данным не является приоритетным направлением, тогда может оказаться выгодным использование низкоуровневых IaaS-сервисов в качестве систем хранения данных, например для видео- и аудиоматериалов.

Для некоторых образовательных учреждений может оказаться выгодным перемещение в "облако" внутренних систем управления обучением (*LMS, Learning Management* Systems). Это хорошая возможность для таких учреждений, которые не могут позволить себе покупку и поддержку дорогостоящего оборудования и программного обеспечения, что позволяет оптимизировать *расходы* на IT-инфраструктуру в современных посткризисных условиях.

2. Программирование в "облаке"

В этом разделе показано, как можно использовать облачные вычисления при обучении основам программирования. Приводится подробный пример работы с *Web*-сервисом, позволяющим создавать и отлаживать учебные программы на любом языке программирования с помощью облачного сервиса сайта http://ideone.com.

Современная практика программирования предполагает активное использование специализированных интегрированных средств разработки (*IDE* – Integrated *Development Environment*). Их использование связано со следующими двумя сложностями:

- Настройка и установка IDE требует высокой квалификации системного администратора.
- Современные IDE достаточно требовательны к ресурсам вычислительной машины, на которой они используются.

Поясним каждый *пункт* подробно. Для обеспечения полнофункциональной работы *IDE* требуется, чтобы квалификация системного *администратор*, осуществляющего установку, настройку и поддержку *IDE* была достаточно высока. Это приводит к необходимости нанимать в учебные заведения на должность системного администратора высококвалифицированных сотрудников, заработная *плата* которых может оказаться существенной статьёй расходов в бюджете образовательного учреждения.

Более того, *затраты* образовательного учреждения могут возрасти вследствие того, что современные *IDE* требуют наличия высокопроизводительных вычислительных машин. Например, одна из самых распространенных *IDE* Microsoft *Visual* Studio 2012 требует для нормальной работы *процессор* мощностью 1,6 ГГц или выше, 1 ГБ *ОЗУ* (или 1,5 ГБ для виртуальной машины), 10 ГБ свободного дискового пространства [7]. Для большинства задач образовательных учреждений не требуется компьютеров с такой высокой производительностью, поэтому их покупка может оказаться недопустимой роскошью.

Обе указанные проблемы позволяет решить применение облачных технологий при обучении программированию. В настоящее время существуют большое количество так называемых *онлайн-IDE*, которые не требуют установки на *компьютер* пользователя и которые требуют для запуска лишь наличие *Интернет*-браузера. *Системные требования* браузеров к оборудованию вычислительной машины традиционно являются скромными. Например, популярный Веб-*браузер* Mozilla Firefox 17 требует для установки *процессор* от 1300 МГц, 512 МБ ОЗЦ и 200 МБ свободного дискового пространства [6], что существенно меньше приведённых ранее цифр для *IDE* Microsoft *Visual* Studio 2012.

Рассмотрим ниже, как можно использовать *онлайн-IDE* в учебных заведениях для обучения основам программирования на примере http://ideone.com. Этот сервис позволяет в режиме *онлайн* создавать тексты программ на разных языках программирования и запускать эти программы на *исполнение* с возможностью анализа полученных результатов. Основные рабочие элементы Ideone показаны на рис. 2.1.

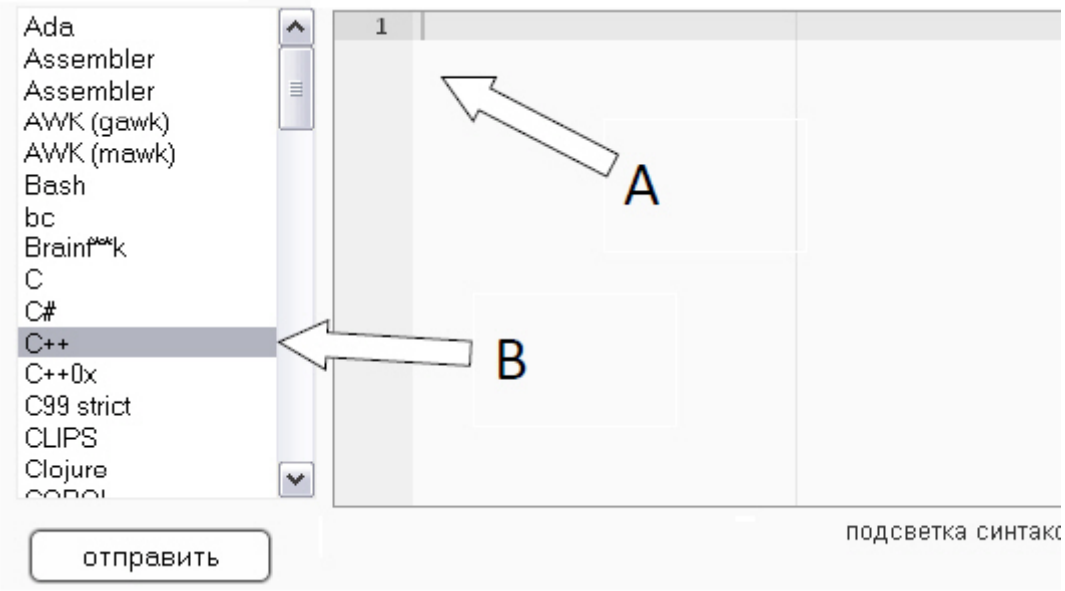
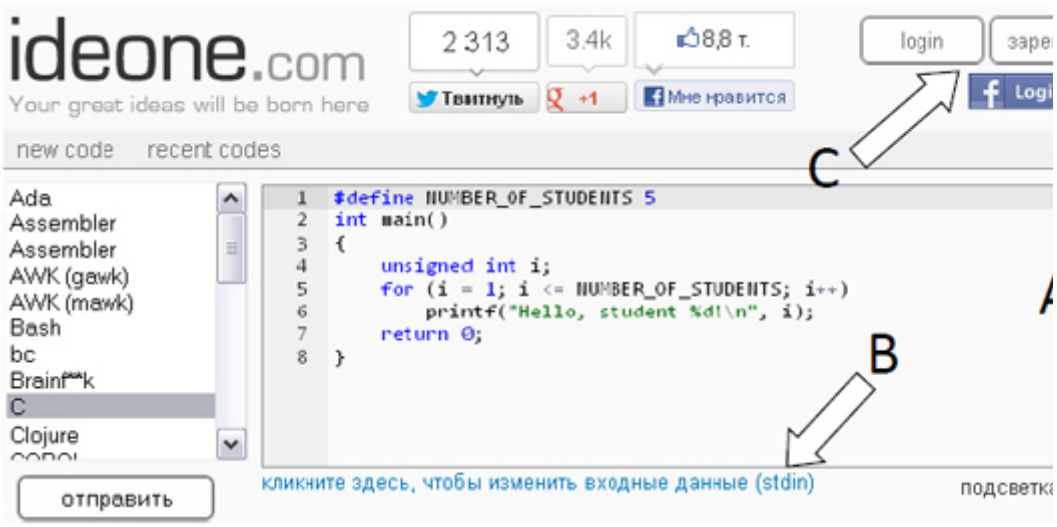


Рис. 2.1. Основные рабочие элементы Ideone

В *поле* "**A**" необходимо ввести текст программы, а в *поле* "**B**" нужно выбрать используемый *язык программирования*, затем нужно нажать кнопку "**Отправить**". В [2] указано, что Ideone поддерживает работу со следующими 55 популярными языками программирования: *Ada*, *Assembler*, *AWK*, *Bash*, bc, Brainf**k, C, C#, C++, *C++0x*, C99 *strict*, *CLIPS*, Clojure, *COBOL*, Common *Lisp* (clisp), D (dmd), Erlang, F#, Factor, Falcon, *Forth*, Fortran, *Go*, Groovy, Haskell, *Icon*, Intercal, *Java*, JavaScript, Lua, Nemerle, Nice, Nimrod, *Node*.js, Objective-C, Ocaml, Oz, PARI/GP, *Pascal*, Perl, *PHP*, Pike, Prolog, *Python*, R, *Ruby*, Scala, *Scheme* (guile), Smalltalk, *SQL*, Tcl, *Text*, Unlambda, VB.*NET*, Whitespace. Очевидно, что этого перечня достаточно при обучении основам программирования практически в любом учебном заведении мира. Более того, при использовании сервиса ideone.com у преподавателя появляется возможность использовать при обучении сразу несколько языков программирования без необходимости поддерживать работу нескольких *IDE*.

Покажем на примере, как может быть организована работа в группе при обучении основам программирования. Рис 2.2 иллюстрирует способ запуска простой программы на языке *Си*. Как можно видеть, в тексте программы используется *подсветка синтаксиса*, аналогичная той, что пользователи привыкли использовать в обычных офлайн-*IDE*. Однако при желании подсветка может быть отключена с помощью элемента управления "**A**". С помощью элемента управления "**B**" можно указать перечень входных данных для программы, что позволяет реализовать более сложную логику работы программы, чем в приведённом примере.

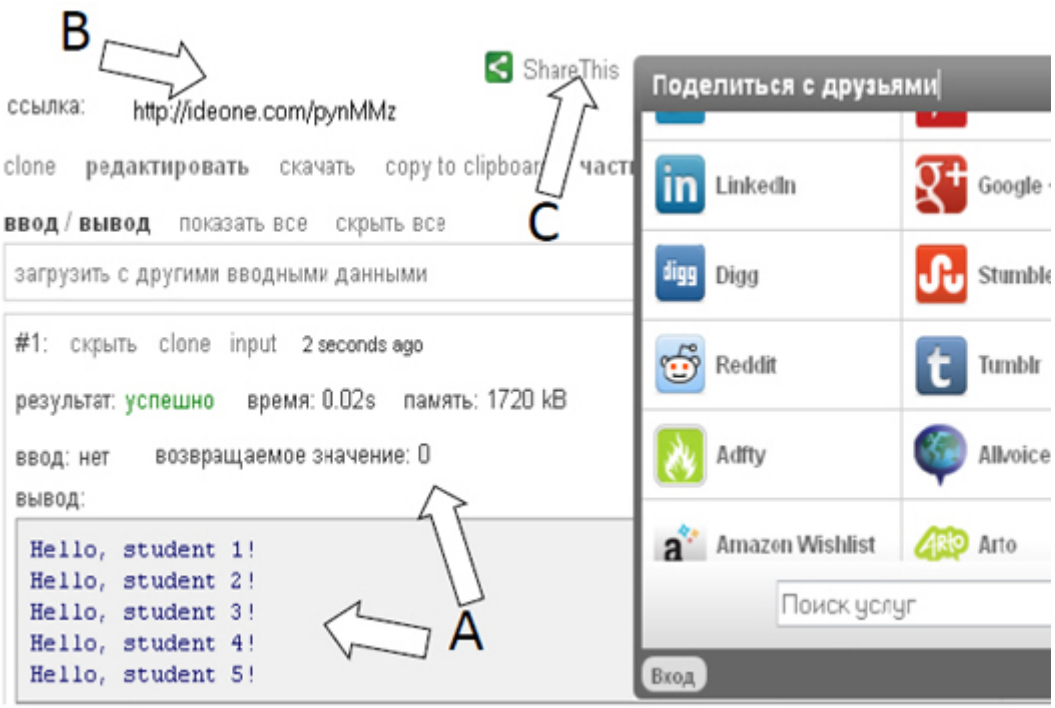


увеличить изображение

Рис. 2.2. Запуск программы в Ideone

Очень важным является элемент управления "**C**", который позволяет персонифицировать работу с программой. Данная возможность крайне ценна при организации учебного процесса. Если преподаватель попросит всех студентов зарегистрироваться в Ideone (или использовать для входа свою учётную *запись* Facebook), то появляется возможность сделать процесс работы с программой коллективным, а процесс совместной работы с программой будет проходить с использованием современных технологий *Web* 2.0. Подробнее об этом расскажем, используя рис. 2.3.

Рис 2.3 представляет собой результаты запуска программы, приведённой на рис. 2.2, самые важные из которых помечены знаком "**A**". Это консольный *вывод* программы и возвращаемое *значение*. Знаком "**B**" отмечена *Интернет-ссылка*, которую преподаватель может переслать студентам для ознакомления с результатам работы демонстрационной программы, либо сами студенты могут выслать подобную ссылку преподавателю в качестве отчёта о проделанной работе.



увеличить изображение

Рис. 2.3. Результаты работы программы в Ideone

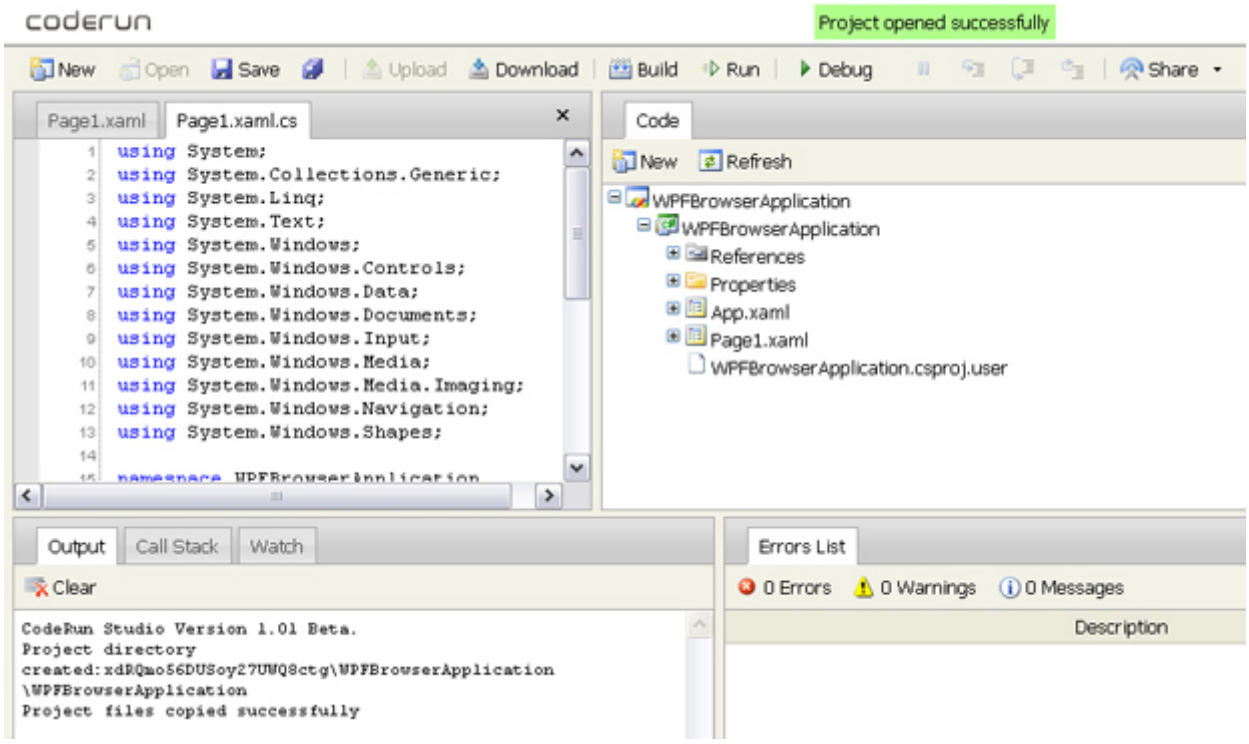
Следующим этапом совместной работы может стать *онлайн*-обсуждение результатов работы программы с помощью средств *Web* 2.0 одного из популярных сервисов социальных сетей. Это становится возможным благодаря использованию элемента управления "**C**" на рис. 2.3.

Этот элемент управления позволяет *выбрать из списка* в правой части экрана один из виджетов популярных сайтов социальных сетей.

К сожалению, *онлайн-IDE* Ideone позволяет реализовать не все из функций традиционных офлайн-*IDE*. Например, отсутствует возможность использовать функции работы с сетью, обращения к файлам и некоторые другие. Также невозможно запустить программу, *время выполнения* которой займёт более 15 секунд или потребности в оперативной памяти превысят 256 МБ, или *объём* программы превысит 64 КБ [2]. Все эти ограничения являются достаточно серьёзными, если планируется использовать Ideone для разработки профессионального программного обеспечения. Однако для образовательных целей эти ограничения более чем приемлемы. Кроме того, для более требовательных преподавателей существуют платные и бесплатные сервисы, аналогичные Ideone, которые при этом в большей степени реализуют функционал традиционных офлайн-*IDE*. Таким примером являются сервисы Cloud9 *IDE* (www.c9.io), CodeRun [3]. На рис. 2.4 показан вид интерфейса сервиса CodeRun: пользователи, работавшие с традиционными офлайн-*IDE*, сразу увидят много знакомых элементов управления. Присутствует панель со списком используемых классов, со списком задействованных в проекте файлов, а также окна с отладочной информацией о состоянии стека вызовов подпрограмм и с ошибками времени комплиляции или времени выполнения.

Созданный проект можно сохранить в офлайн, но предпочтительным является режим работы только в облаке. Все *операции*, включая отладочную сборку, *анализ* результатов выполнения в консольком режиме, компиляцию под разные платформы и операционные системы можно выполнить в режиме *онлайн*. В идеале окончанием работы программиста

будет скачивание готовых бинарных файлов с работающей программой. Такой подход позволяет сэкономить используемой офлайн дисковое *пространство*, а также позволяет компилировать проект существенно более быстро, чем на рабочем месте пользователя, если это рабочее *место* оборудовано устаревшим аппаратным обеспечением. Итогом этого является возможная финансовая экономия для образовательного учреждения.



увеличить изображение

Рис. 2.4. Интерфейс системы Coderun

Однако помимо экономического эффекта, можно получить и существенные преимущества при организации образовательного процесса. Студенты получают возможность совместно редактировать программные проекты, находясь у себя дома. Это позволяет реализовывать сложные курсовые проекты и лабораторные работы с существенной экономией на осуществление организационных мероприятий со стороны преподавателя.

2.1. "Облачные" сервисы хранения данных

В этом разделе показано, как можно использовать облачные технологии для хранения различных типов данных в "облаке". В качестве примера рассматривается популярный сервис "Dropbox", официальный *сайт* https://www.dropbox.com.

На сегодняшний день (конец 2012) в сети *Интернет* существует более 30-ти бесплатных(!) сервисов облачного хранения данных [4]. Каждый из них предлагает возможности *по* хранению данных любых типов, начиная от офисных документов и заканчивая мультимедийной информацией.

Почти все (около 90%) из поставщиков этих сервисов предлагают следующие услуги бесплатно [4]:

- Объём бесплатного хранилища – 2 и более ГБ (до 18 ГБ предлагает сервис Dropbox [5], до 20 ГБ – Яндекс.Диск, 50 ГБ – Adrive).
- Автоматическая синхронизация хранимых данных между всеми устройствами, которые подключены к облачному сервису. Не нужно использовать внешнее запоминающее устройство (Flash-диск, CD/DVD-накопители) для того, чтобы перенести данные на другое устройство (ПК, ноутбук, планшет, смартфон и т.д.). При подключении устройства к сети Интернет актуальная версия данных будет автоматически загружена на устройство. Эта функция экономит массу времени – можно быстро продолжить работу над текущей учебной задачей, вернувшись домой из учебного заведения.
- Безопасность хранения данных в "облаке". Весь трафик между клиентом и "облаком" шифруется (используется, как минимум, протокол SSL, а в некоторых RSA+AES), что очень сильно затрудняет просмотр передаваемой информации посторонними лицами. Поэтому уровень безопасности работы с данными выше, чем, например, при отправке обычным письмом по электронной почте. Некоторые сервисы облачного хранения (SpiderOak, Wuala) предлагают шифрование данных не только при передаче, но и при хранении в "облаке" [8, 9].
- Возможность публичного доступа через Интернет к файлам, хранящимся в облаке, для любого человека. Достаточно отправить коллеге ссылку на нужный файл, чтобы он смог ознакомиться, например, с результатами вчерашнего теста или последней версией учебного материала.
- Надежность хранения данных. Поставщики облачных решений хранят данные на своих сервисах с использованием избыточности, что само по себе гарантирует надежность. Дополнительно к этому, на любом из устройств, подключенных к "облаку", хранится, ещё как минимум, еще одна актуальная копия данных.

Для образовательного учреждения использование *онлайн*-сервисов хранения данных является экономически выгодным. Для создания сетевого файлового хранилища своими силами необходимо:

- приобрести сетевое и серверное оборудование;
- разработать политики хранения и общего доступа к информации;
- произвести первоначальную установку и настройку программного обеспечения;
- обеспечить регулярное резервное копирование данных и возможность быстрого восстановления;
- выделить рабочий персонал для администрирования файлового хранилища.

Финансовые *затраты* на проведение вышеперечисленных мероприятий зависят от числа пользователей и могут быть очень большими. Например, для организации файлового хранилища для 1000 учащихся с выделением 5 ГБ дискового пространства на каждого потребуется закупить *минимум* 4 жестких диска емкостью 2,5 ТБ (10 ТБ для обеспечения дублирования хранимых данных *по* технологии *RAID* 1), высокопроизводительный *RAID*-контроллер и серверное оборудование. Помимо дополнительных расходов на заработную плату обслуживающему персоналу, необходимо помнить о регулярных расходах на оплату счетов за потребляемую оборудованием электроэнергию.

Регистрация. Начать использовать любой облачный сервис *по* хранению данных очень легко (см. пример для сервиса Dropbox на рис. 2.5). Как правило для регистрации требуется указать действующий *адрес* электронной почты и придумать новый *пароль* для доступа к "облаку".

Сразу после регистрации пользователю доступно 2 ГБ в файловом хранилище. Однако выполнив ряд несложных действий, можно увеличить доступный размер бесплатного хранилища в несколько раз. Для этого нужно порекомендовать сервис Dropbox друзьям или выполнить иные действия рекламного характера.

Существует два режима работы с облачным сервисом: через вэб-*интерфейс* и через программу-клиент. Рассмотрим подробно каждый из них ниже.

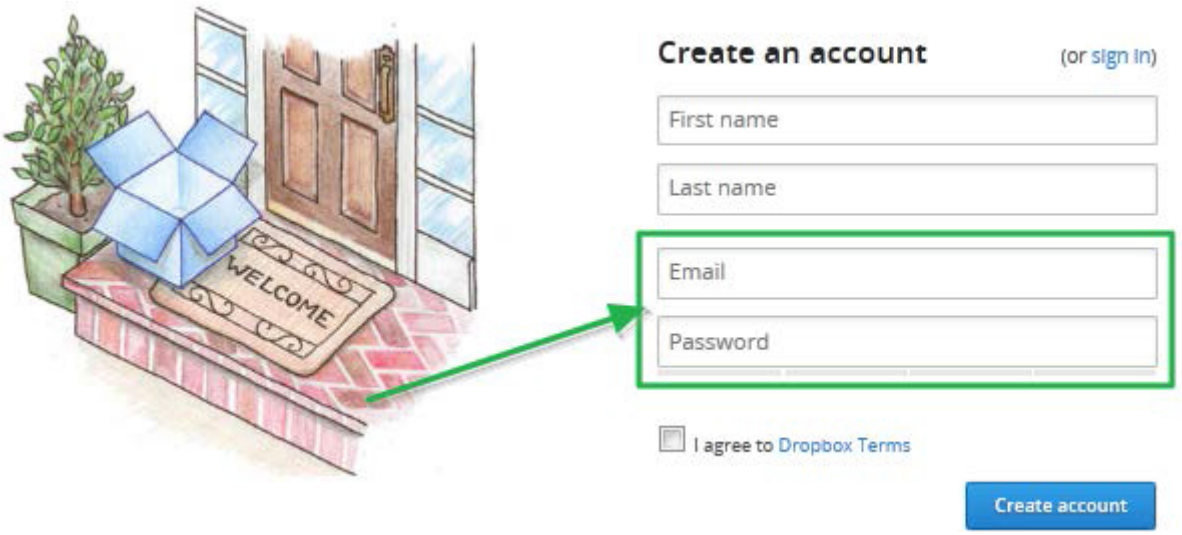
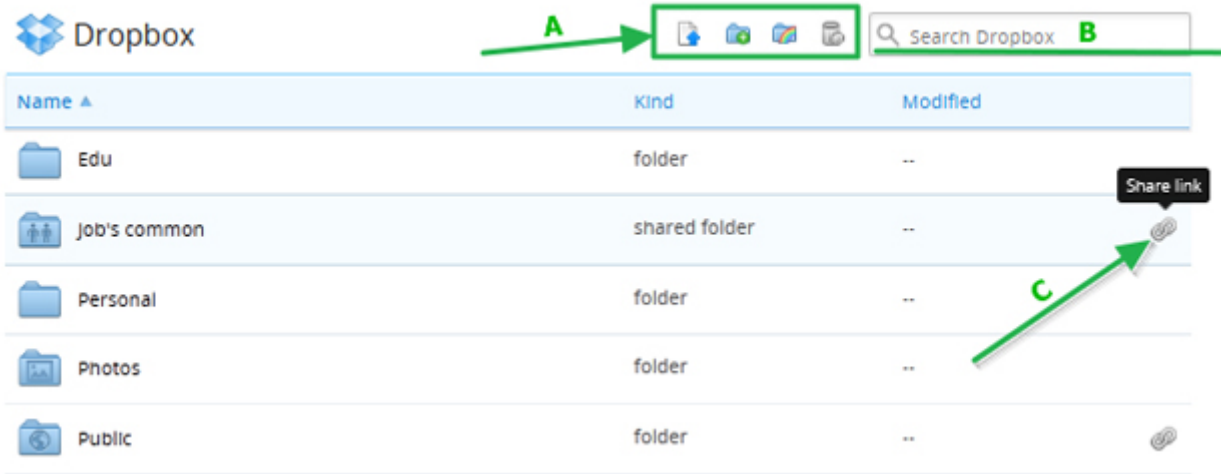


Рис. 2.5. Страница регистрации в сервисе "Dropbox"

Работа через веб-интерфейс. Веб-*интерфейс* работы с облачным хранилищем очень прост. Для работы с ним нужен только *доступ* к сети *Интернет* и вэб-*браузер*. На рис. 2.6 показаны возможности работа в популярном сервисе облачного хранения данных "Dropbox". Меткой "**A**" указаны общие *элементы управления* данными:

- загрузка новых файлов (пользователь указывает, какой файл на своем компьютере он желает поместить в "облако");
- добавление новых каталогов (каталоги позволяют удобно структурировать данные пользователя для ускорения доступа к ним);
- предоставление общего доступа (пользователь может указать, кому именно следует дать права на чтение или запись его файлов);
- удаление (если файл стал не нужен, то его можно удалить из "облака").



увеличить изображение

Рис. 2.6. Веб-интерфейс сервиса "Dropbox"

Рядом с элементом "**A**" располагается строка поиска "**B**". Общий *доступ на чтение* к файлам и каталогам можно предоставить, нажав на значок "**C**", после чего на указанные электронные адреса участников будет отправлена *ссылка* на выбранные объекты общего доступа

Работа через программу-клиент. Режим работы через программу-клиент предоставляет больше возможностей пользователю. Для работы с ней необходимо скачать установочный *файл* и установить программу на локальный *компьютер*. У самых популярных "облачных" сервисов имеется кросс-платформенные клиенты, которые можно запускать из разных операционных систем (*Windows*/Linux/*Mac*/Android и т.д.). Как видно ниже на рис. 2.7, после установки программы-клиента необходимо ввести регистрационные данные и указать имя устройства (*Computer name*), данные которого будут синхронизироваться с "облаком".

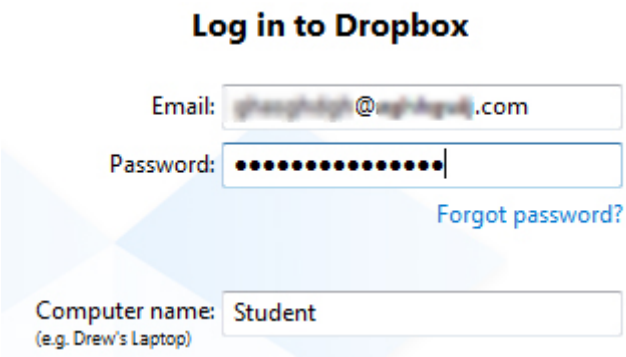


Рис. 2.7. Ввод регистрационных данных в Dropbox

После входа в клиентскую программу нужно настроить один (в сервисе Dropbox) или несколько каталогов (в сервисах SpiderOak, *Box*, Wuala, BitCasa и другие), содержимое которых будет синхронизироваться с "облаком". На рис. 2.8 показано, как может выглядеть содержимое каталога Dropbox при запуске программы-клиента на операционной системе Ubuntu 11.10.

Важно понимать относительность понятия "*Computer name*", которое используется в большей степени не для идентификации того или иного компьютера, а для идентификации конкретного набора файлов в облаке. Поэтому если установить клиент Dropbox на другие устройства, но указать в программе-клиенте те же учетные данные, то при изменении файлов в каталоге "Dropbox" на одном устройстве, файлы в каталоге "Dropbox" на всех остальных устройствах с одинаковым "*Computer name*" будут также обновлены. Это утверждение справедливо не только для сервиса Dropbox, но и для прочих клиентских программ облачных сервисов хранения данных.

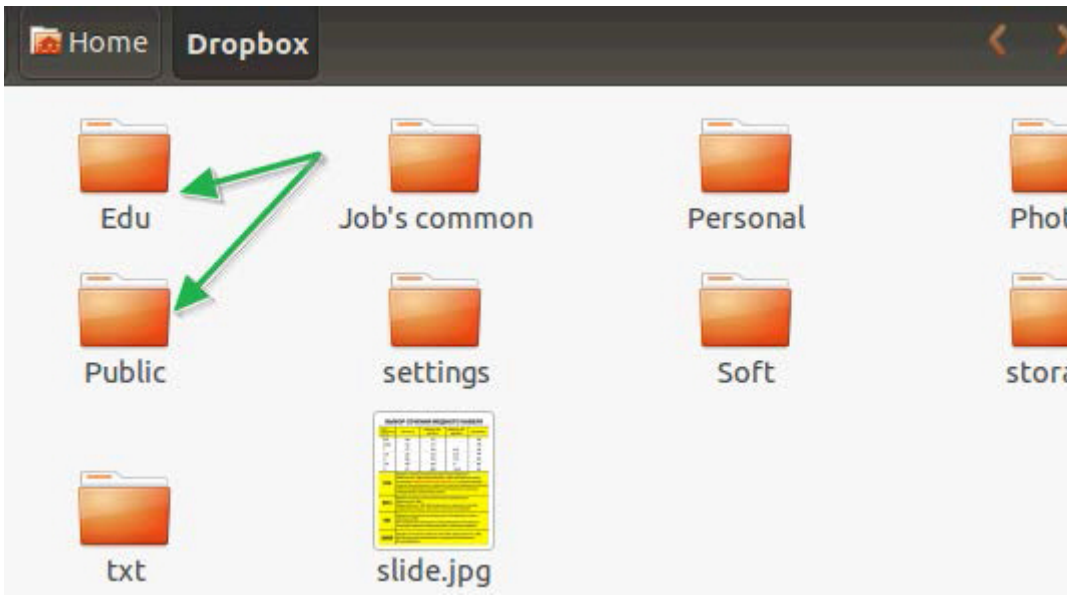
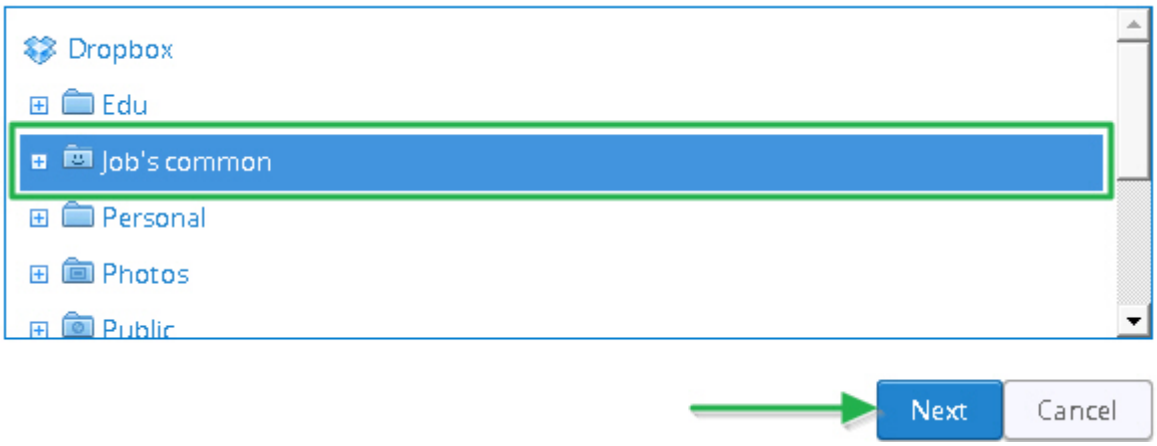


Рис. 2.8. Содержимое синхронизированного каталога "Dropbox" в Ubuntu 11.10

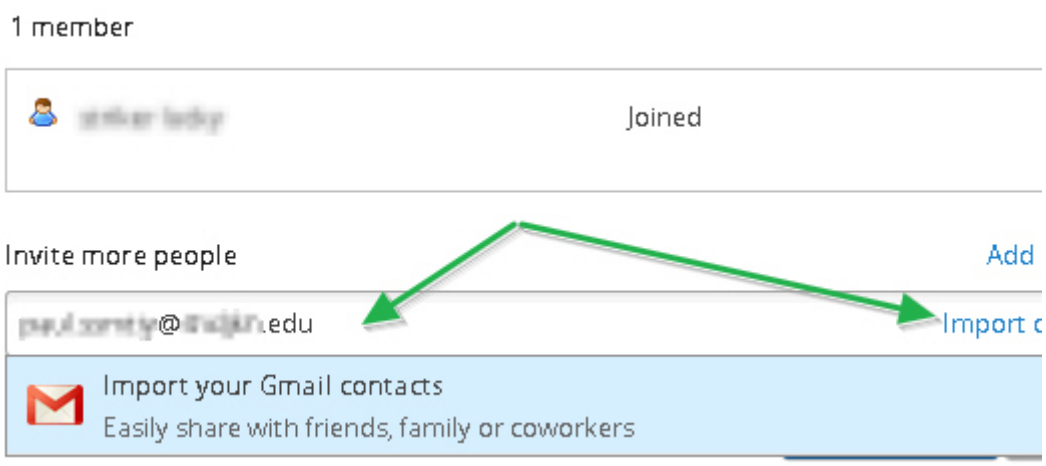
Предоставление общего доступа. Общий *доступ* к файлу или каталогу в "облаке" осуществляется в три шага (на примере "Dropbox"):

- В веб-клиенте в разделе "**Sharing**" (предоставление общего доступа) необходимо нажать кнопку "**New shared folder**":

- Далее следует выбрать существующий каталог либо создать новый:



- Затем добавить адреса электронной почты участников проекта, которым необходимо предоставить общий доступ к указанному каталогу:



После этих действий каждый из участников получит *по* электронной почте ссылку, *по* которой он сможет открыть общий *ресурс* и начать участие в проекте.

Общие рекомендации *по* использованию *онлайн*-сервисов хранения данных в образовательных учреждениях можно сформулировать следующим образом. В учебных заведениях можно предложить следующий *вариант использования* хранения данных в "облаке".

Прежде всего, необходимо зарегистрировать участников образовательного процесса на сервисе "облачного" хранения. Затем создать общие каталоги и настроить *права* доступа пользователей к ним. Например, учебные материалы *по* курсу можно поместить в каталог, к которому сделать *доступ* только на чтение. Для каждого слушателя можно сделать индивидуальный каталог с полным доступом для сдачи лабораторных *работ* или отчетов. Это один из вариантов использования облачных сервисов хранения данных. Можно придумать любую другую удобную схему обмена данными для решения задач образования в выбранном учебном заведении.

Сравнение облачных сервисов для хранения данных. Как было отмечено в начале раздела, в настоящее время существует достаточное количество *онлайн*-сервисов хранения данных. Для удобства выбора характеристики самых популярных их этих сервисов сведены в табл. 2.1. Критерием выбора тех сервисов, которые попали в сравнительную таблицу, была в том числе *минимизация* стоимости лицензии, т.к. это является важным фактором при выборе облачных услуг в образовательных учреждениям. Поэтому в табл.1 включены данные только о бесплатных сервисов тех или иных поставщиков услуг.

Таблица 2.1. Сравнение бесплатных облачных услуг для хранения данных

Название	**Бесплатный объем, ГБ**	**Шифро-вание данных**	**Поддерживаемые операционные системы**	**Общий доступ**	**Коллективная работа**	**Кол-во клиентских компьютеров**
Drop box	2	SSL, AES 256	Windows, Mac OS, Linux, Android, iOS	Да	Нет	∞
Spider Oak	2	RSA 2048, AES 256	Windows, Mac OS, Linux, Android, iOS	Да	Нет	∞
MS Sky Drive	7	SSL, AES 128	Android, iOS, Windows, Mac OS	Да	Да	∞

Box. com	5	SSL, AES 256	Android, Windows Mobile,Ipad, Iphone	Да	Да	∞
Wuala	5	AES 256, RSA 2048, SHA-256	Windows, Mac OS, Linux, Android, iOS	Да	Нет	∞
Adrive	50	SSL	Android, iOS	Да	Да	1
Яндекс. Диск	10	Нет	Windows, Mac OS, Linux, Android, iOS	Да	Нет	∞

Подводя итоги краткого сравнения "облачных" сервисов хранения данных, можно сделать следующие выводы:

- любые современные онлайн-сервисы предлагают достаточное количество дискового пространства для хранения документов и учебных материалов;
- почти все сервисы поддерживают современные алгоритмы шифрования при передаче данных;
- в качестве хранилища резервных копий учебных материалов можно использовать сервис "Adrive" благодаря бесплатному объему в 50 ГБ;
- если необходима конфиденциальность хранения данных, то самый высокий уровень защиты при передаче и хранении информации в облаке обеспечивает SpiderOak (шифрование данных происходит на клиентском устройстве);
- для совместной работы над документами и электронными таблицами прекрасно подойдет сервис Box.com.

Онлайн-сервисы хранения данных обладают большими преимуществами *по* сравнению с локальными сетевыми хранилищами. Использование в процессе обучения одного или нескольких облачных сервисов хранения данных значительно повысит его эффективность, а также позволит образовательному учреждению идти в ногу со временем.

2.3. Защита информации при использовании сервисов облачного хранения

Рассмотрим вопросы обеспечения защиты информации при использовании сервисов облачного хранения на примере Dropbox. Поскольку условия обслуживания Dropbox могут измениться без предварительного уведомления пользователя (т.е. учебного заведения), а данные могут храниться на сервисе Dropbox достаточно долгое время (для использования в учебном процессе в течение последующих лет), необходимо уделить особое внимание защите информации при использовании сервисов облачного хранения. Для этого необходимо решить следующие задачи:

- Защита персональных данных.
- Защита метаданных в открытых файлах.
- Защита доступа к закрытым данным.

Защита персональных данных. Для защиты персональных данных учащихся, студентов, преподавателей и других сотрудников учебного заведения необходимо применять следующие меры: 1) не предоставлять избыточную информацию и 2) отключить элемент **cookies** (куки) в используемом браузере. Рассмотрим подробно каждую из двух предложенных мер.

Для обеспечения защиты персональных данных не стоит предоставлять избыточную информацию при регистрации на сервисе в личном профиле, т.к. потенциальный *злоумышленник* может использовать избыточную информацию в преступных целях. Например, не имеет смысла указывать следующие *персональные данные*:

- номер телефона;
- сведения о кредитной карте;
- почтовый адрес;
- сведения о регистрации в социальных сетях.

Для корректной работы облачных сервисов, как правило, требуются лишь минимальный набор персональных данных: *имя пользователя*, действующий *адрес* электронной почты и *пароль*.

Второй рекомендацией является совет отключить **cookies** (куки) в браузере для того, чтобы при посещении сайта не сохранялась *информация* о персональных предпочтениях и настройках пользователя, не записывались данные о логине и пароле пользователя, не велась *статистика* различных персональных особенностей работы пользователя. Например, для того чтобы отключить куки в браузере Mozilla Firefox, необходимо всего лишь отключить галочку "**Принимать куки с сайтов**" в настройках браузера на странице "**Приватность**" (рис. 2.9).

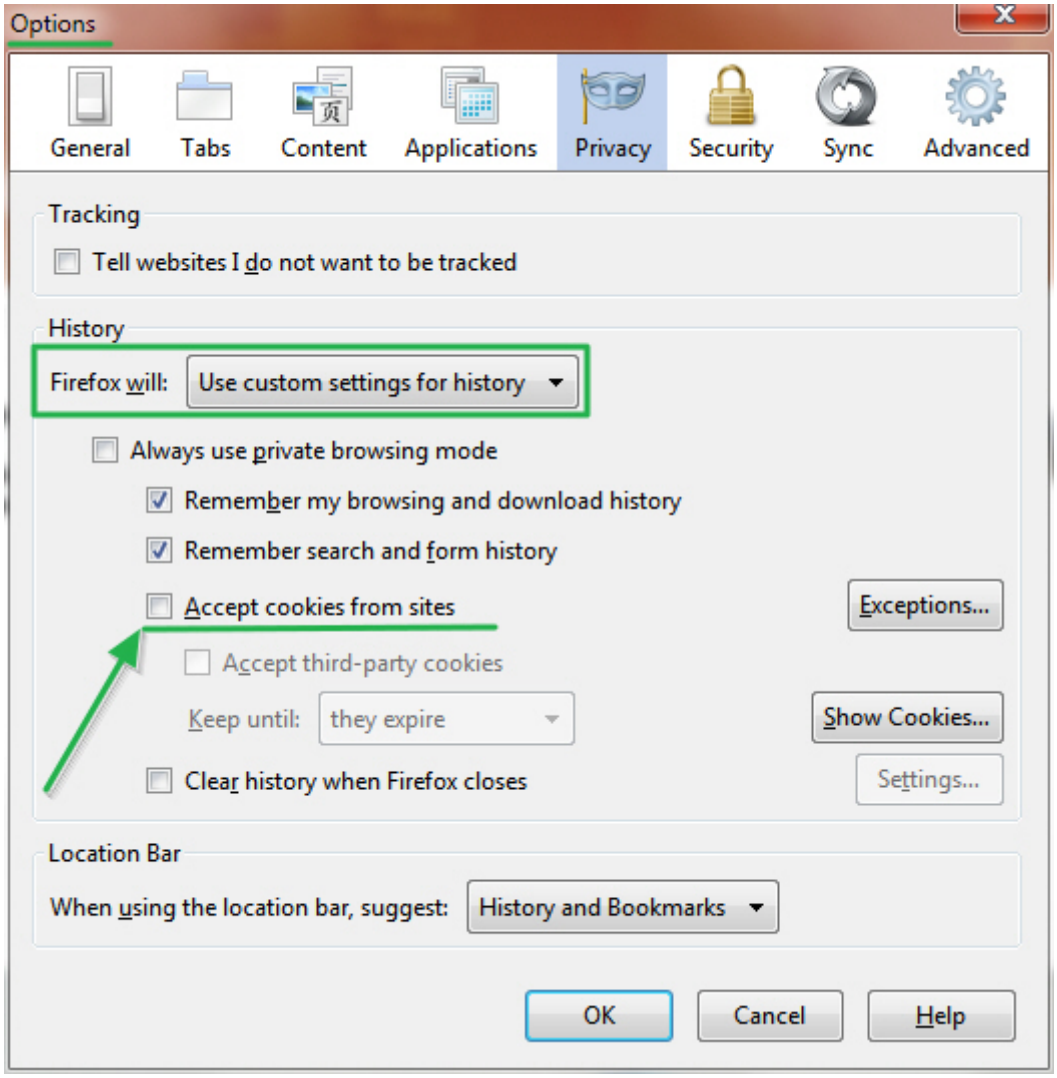


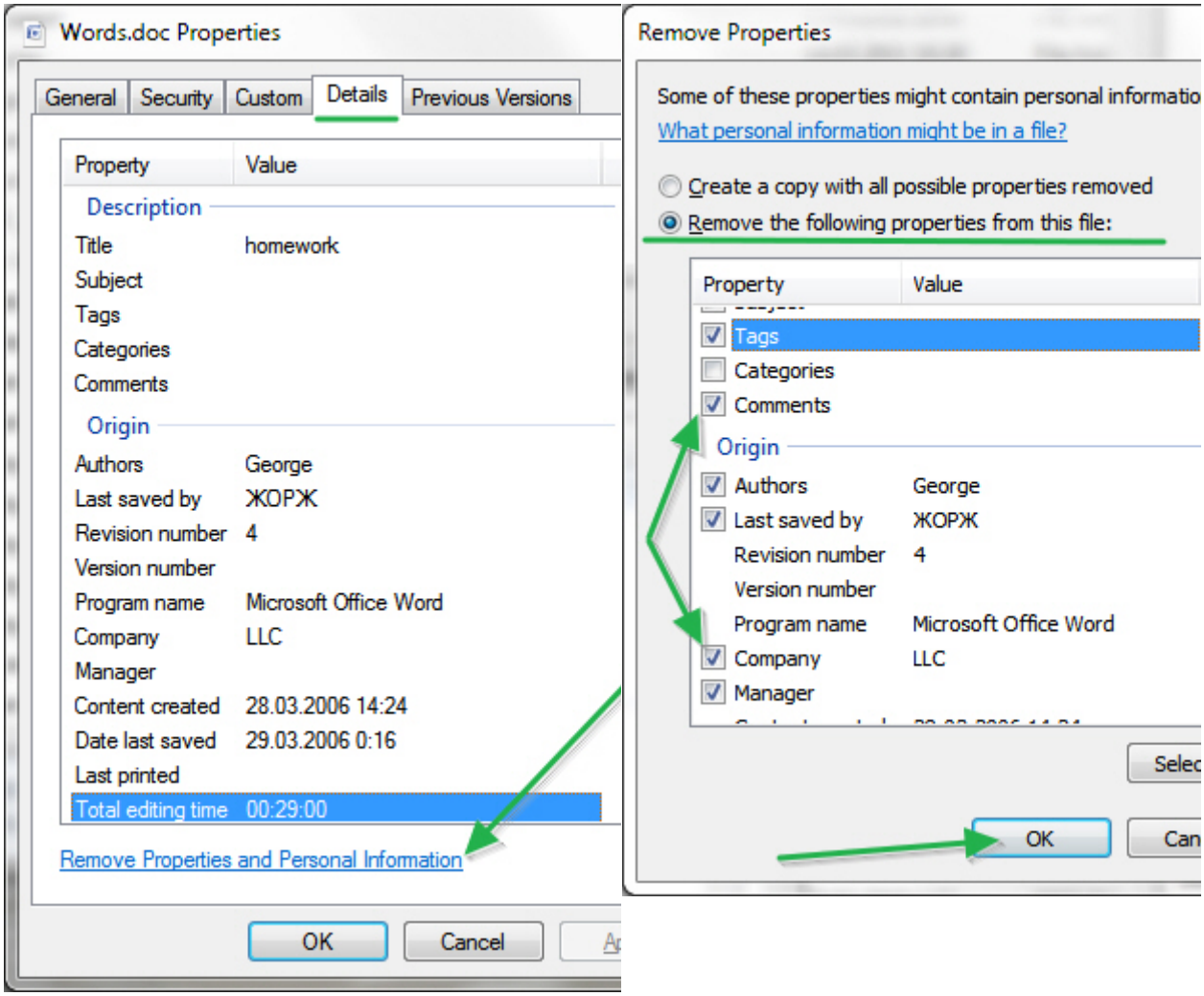
Рис. 2.9. Отключение куки в популярном браузере Mozilla Firefox (версия 18.0)

Стоить помнить, что после отключении куки, *браузер* не будет запоминать логин и *пароль* для входа на сервис облачного хранения, поэтому придется вводить эти данные вручную при каждом посещении сервиса через *web-интерфейс*.

Защита метаданных. Под метаданными понимаются некоторые виды автоматически создаваемых в этих файлах служебных записей, которые характеризуют каким-либо образом эти файлы. К метаданным относятся, например, *EXIF*-данные, автоматически добавляемые современной фото- и видеоаппаратурой в создаваемые файлы с медиаданными (например, *JPEG*).

Защитить *метаданные* при предоставлении общего доступа можно только путем их удаления из файлов перед загрузкой на сервис облачного хранения. Это можно сделать как средствами операционной системы, так и при помощи специального программного обеспечения.

Удаление метаданных средствами операционной системы. Для просмотра метаданных файла в *Windows* 7 необходимо зайти в контекстное *меню* файла (нажать на нём правой кнопкой мыши), перейти на вкладку "**Подробно**" и выбрать "**Удаление свойств и личной информации**" (указано стрелкой слева на рис. 2.10).



Для того, чтобы удалить персональную информацию нужно в появившемся окне (справа на рис. 2.10) выбрать режим "**Удалить следующие свойства для этого файла**", отметить галочками пункты свойств, которые необходимо удалить и нажать кнопку "**ОК**".

Удаление данных EXIF-данных. Существует специальные программы для удаления мета-данных из изображений и других медиафайлов. Одной из таких программ является бесплатная *программа* MetaStripper (http://www.photothumb.com/metastripper/), которая удаляет *EXIF*-информацию из изображений, сохраненных в формате *JPEG*.

Программа очень проста в использовании: после запуска необходимо указать исходный каталог с изображениями и каталог, куда будут сохранены изображения без метаданных *EXIF* (рис. 2.11):

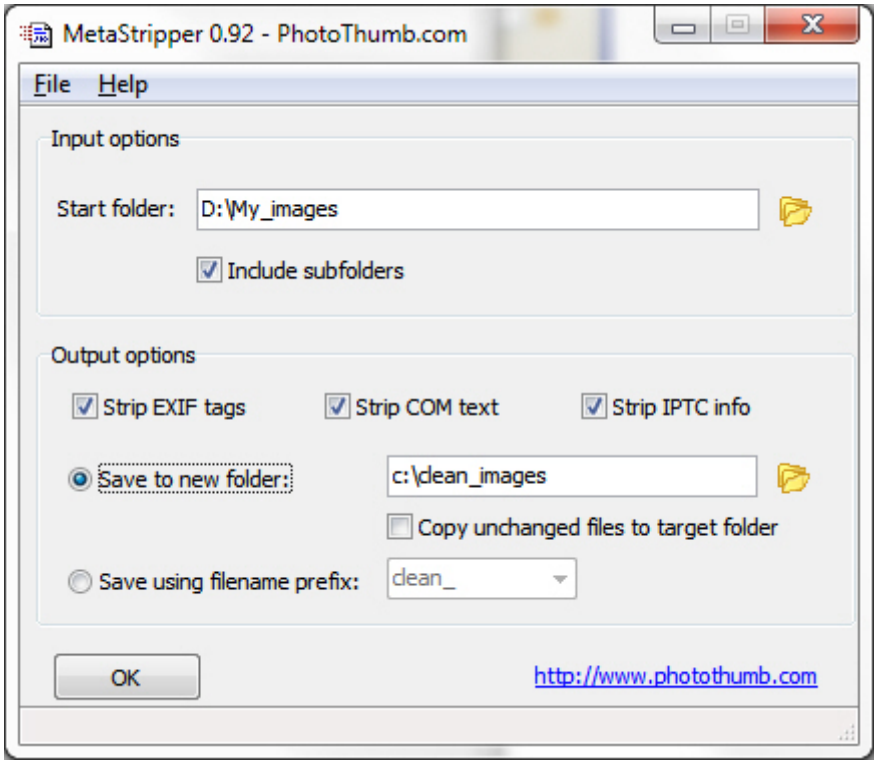


Рис. 2.11. Окно настроек программы MetaStripper

Единственный недостаток программы в том, что она работает только под управлением операционной системы *Windows*. В операционной системе Ubuntu следует пользоваться программным
средством jhead (http://manpages.ubuntu.com/manpages/intrepid/man1/jhead.1.html). Для установки этого средства необходимо открыть окно терминала и в командной строке написать (если используется ОС Ubuntu):

```
sudo apt-get install jhead
```

Чтобы полностью удалить информацию **EXIF** из определенной картинки необходимо выполнить в командной строке команду:

```
jhead -purejpg /path_to_image.jpg
```

В этой строке "path_to_image.jpg" обозначает *путь* к файлу, из которого необходимо удалить *метаданные*. В качестве пути к файлам можно указать *шаблон*, например "/usr/home/Pictures /*.jpg", тогда *exif-информация* будет удалена из всех файлов с расширением .jpg в папке "**/usr/home/Pictures/*.jpg**".

Защита доступа к закрытым данным. Надежным способом защиты информации при размещении в облачном хранилище является применение средств шифрования. При шифровании данных исходная *информация*, которую необходимо защитить от посторонних, преобразуется специальным образом с использованием одного или нескольких криптографических алгоритмов.

Будучи зашифрованной, эта *информация* может быть прочитана или изменена, только если *пользователь*, которому предназначается *информация*, знает правильный *пароль*, при помощи которого он сможет расшифровать эту информацию. Для целей шифрования информации рекомендуется использовать бесплатное открытое *программное обеспечение* TrueCrypt (http://www.truecrypt.org/).

TrueCrypt позволяет создать виртуальный зашифрованный *логический диск*, хранящийся в виде обычного файла с любым расширением (*файл-контейнер*). Этот *файл* можно опубликовать на сервисе облачного хранения и предоставить к нему *доступ* тем пользователям, с которыми необходимо поделиться конфиденциальной информацией.

Достоинства TrueCrypt. TrueCrypt является кросс-платформенной программой и поддерживается такими операционными системами как *Windows* 7/Vista/XP/2000, Linux, *Mac* OS X. TrueCrypt является свободным программным обеспечением с открытым исходным кодом, поэтому его можно рекомендовать к применению в образовательных учреждениях. Также TrueCrypt поддерживает несколько криптографических алгоритмов, позволяет осуществлять каскадное *шифрование* файлов, а также предлагает возможность правдоподобного отрицания. Рассмотрим эти достоинства подробно.

TrueCrypt поддерживает следующие криптографические алгоритмы блочного шифрования: *AES*, Twofish, *Serpent* (рис. 2.12). Эти алгоритмы являются победителями открытого конкурса *AES*, организованного *NIST* (https://en.wikipedia.org/wiki/Advanced_Encryption_Standard_process), и гарантируют хороший уровень стойкости к взлому при использовании надежных паролей. *Алгоритм AES* (Rijndael), который является победителем конкурса *AES*, принят в качестве стандарта Агентства национальной безопасности США для защиты сведений, составляющих государственную тайну (https://en.wikipedia.org/wiki/Rijndael). Алгоритмы Twofish и *Serpent* также являются надежными алгоритмами шифрования – в настоящее время не было обнаружено каких-либо реально реализуемых атак на них http://www.ixbt.com/soft/alg-encryption-aes-2.shtml)

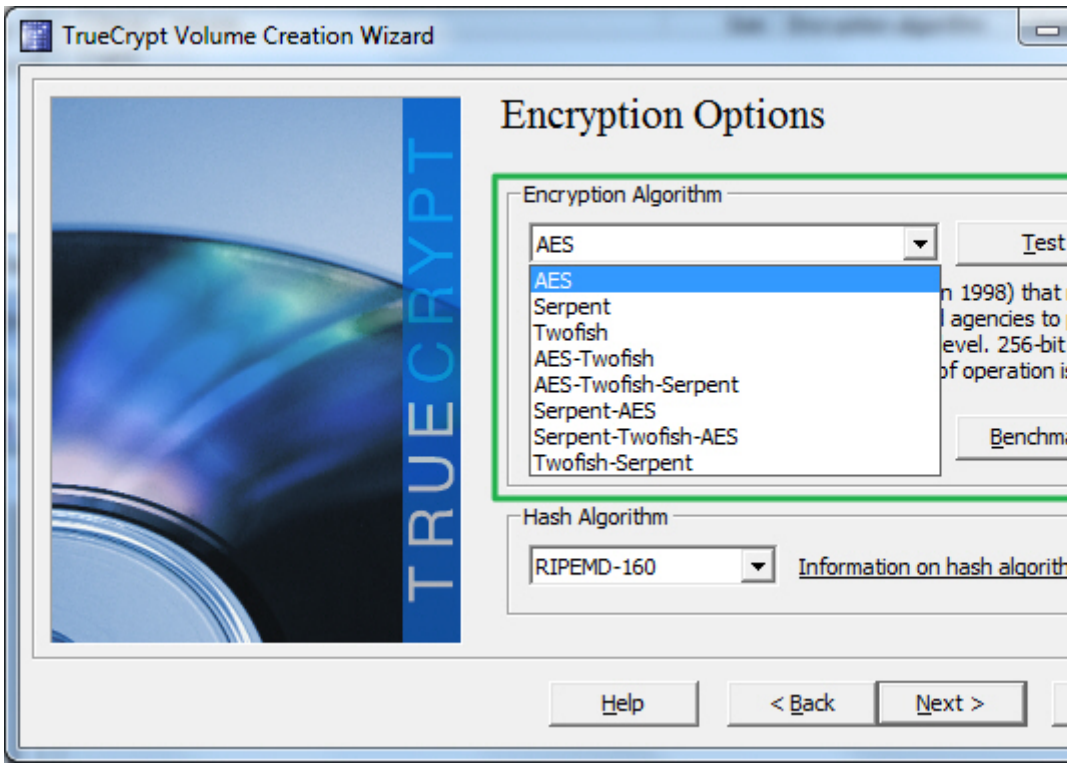


Рис. 2.12. Выбор алгоритмов шифрования в TrueCrypt

Для шифрования сведений уровня "**Совершенно секретно**" (*Top Secret*) рекомендуется использовать длину ключа шифрования в 256 *бит*. Для уровня "**Секретно**" (*Secret*) используются ключи длиной 128 и 192 бита. Стандарт *AES* поддерживает эти три длины ключа шифрования (128/192/256) и поддерживает размер блока в 128 *бит*. Это означает, что перед шифрованием *информация* разбивается на блоки размером 128 *бит*, каждый из которых шифруется с использованием ключа шифрования.

Каскадное шифрование. Помимо поддержки криптостойких алгоритмов шифрования, TrueCrypt поддерживает каскадное *шифрование* (http://www.truecrypt.org/docs/cascades) различными шифрами. Например, данные можно зашифровать каскадом из сразу трёх шифров **AES+Twofish+Serpent**. В этом случае каждый из 128-битных блоков исходной информации будет зашифрован сначала при помощи алгоритма *AES* (с длиной ключа 256 *бит*), затем при помощи Twofish (с длиной ключа 256 *бит*) и в конце – при помощи алгоритма *Serpent* (с такой же длиной ключа в 256 *бит*). Каждый из каскадных шифров использует свой собственный *ключ*. Все ключи шифрования абсолютно независимы друг от друга, даже не смотря на то, что формируются они на основе одного пароля.

Другие виды каскадов, которые поддерживает TrueCrypt: *AES*-Twofish, *Serpent*-*AES*, *Serpent*-Twofish-*AES*, Twofish-*Serpent*. Очевидно, что использование каскадного шифрования значительно увеличивает уровень защиты зашифрованной информации.

Правдоподобное отрицание. TrueCrypt позволяет использовать правдоподобное *отрицание* (Plausible Deniability), если кто-то принуждает пользователя сообщить *пароль* от файла-контейнера. Это становится возможным благодаря двум особенностям (http://www.truecrypt.org/docs/plausible-deniability):

- Возможность создания скрытого тома внутри файла-контейнера, для доступа к которому необходимо указать отдельный пароль. Все данные, которые необходимо уберечь от посторонних глаз, хранятся на скрытом томе, а данные, которые пользователь сможет

выдать в случае шантажа – на обычном томе. Таким образом, в случае шантажа пользователь может сообщить злоумышленнику пароль от обычного тома и ценная информация при этом не будет раскрыта.

- Невозможность идентификации томов TrueCrypt. Файл-контейнер TrueCrypt невозможно отличить от набора случайных данных (не содержит никаких сигнатур), то есть файл нельзя ассоциировать с TrueCrypt как с программой, его создавшей.

Использование TrueCrypt с сервисами облачного хранения данных. Покажем на примере, как можно использовать программу TrueCrypt для защиты данных при работе с системами облачного хранения данных. Для создания зашифрованного файла-контейнера нужно в программе TrueCrypt открыть *меню* "**Тома**", выбрать *пункт* "**Создать новый том**", в появившемся окне отметить *пункт* "**Создать зашифрованный файл-контейнер**" (рис. 2.13).

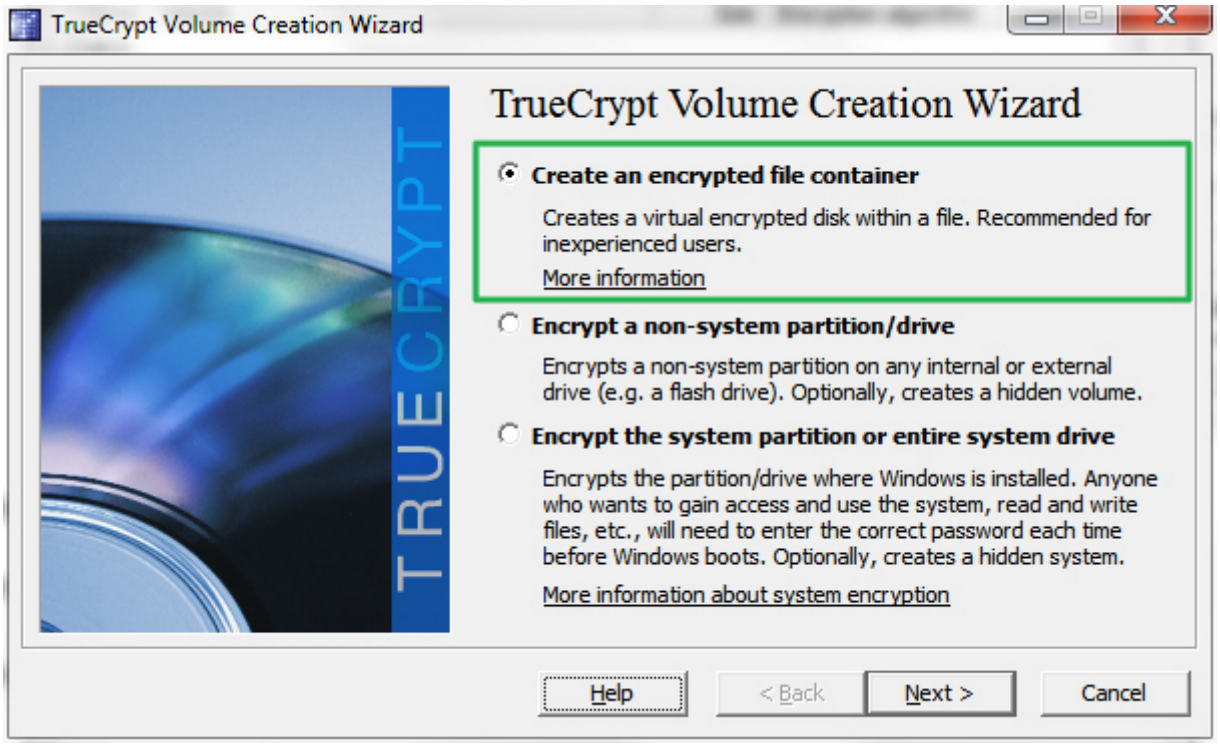


Рис. 2.13. Мастер создания томов TrueCrypt

На следующем шаге необходимо выбрать вид тома, который необходимо создать: стандартный или скрытый (рис. 2.14). Далее необходимо указать *путь* к новому файлу, в котором будет храниться *контейнер* с зашифрованной информацией. Для использования в облачных сервисах хранения необходимо указать *путь* к файлу в стандартном каталоге облачного сервиса, например для сервиса Dropbox это стандартная *папка* "**Dropbox**".

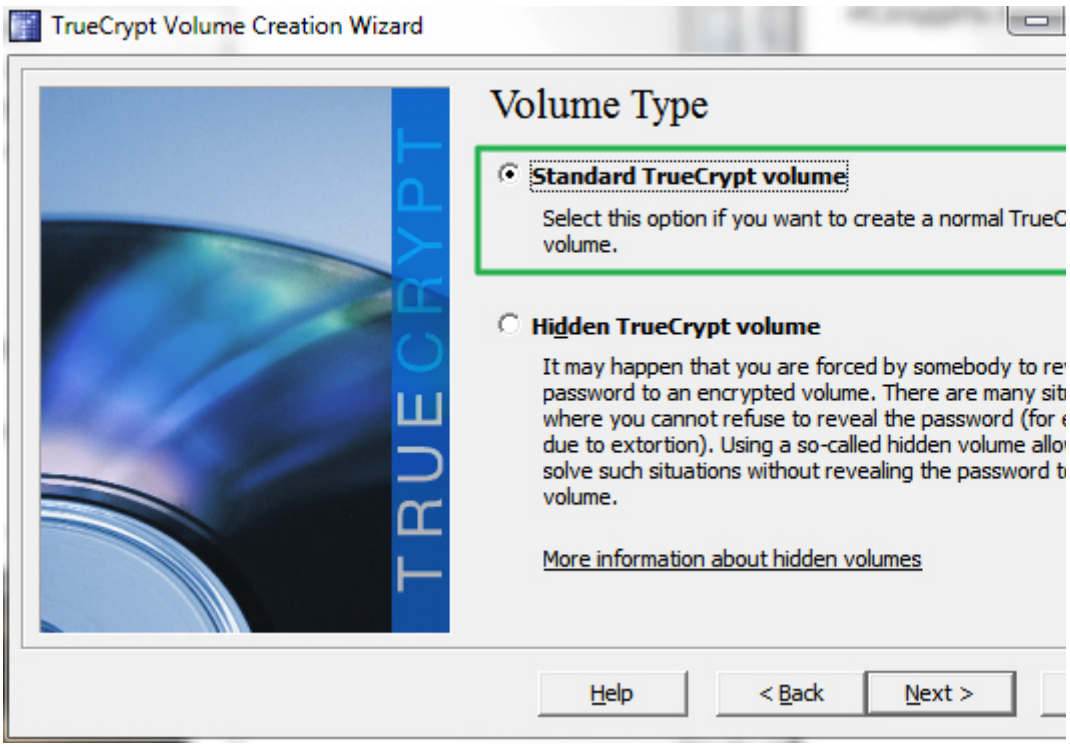


Рис. 2.14. Выбор типа тома TrueCrypt

Имя файла и его расширение могут быть произвольными. В целях повышения конфиденциальности лучше придумать нейтральное *имя файла* (например, "students_manual.pdf", "*course*.iso" и т.д.), чтобы оно не вызывало большого интереса у потенциальных злоумышленников. Для дополнительного увеличения уровня безопасности необходимо поставить галочку "**Никогда не сохранять историю**", см. рис. 2.15.

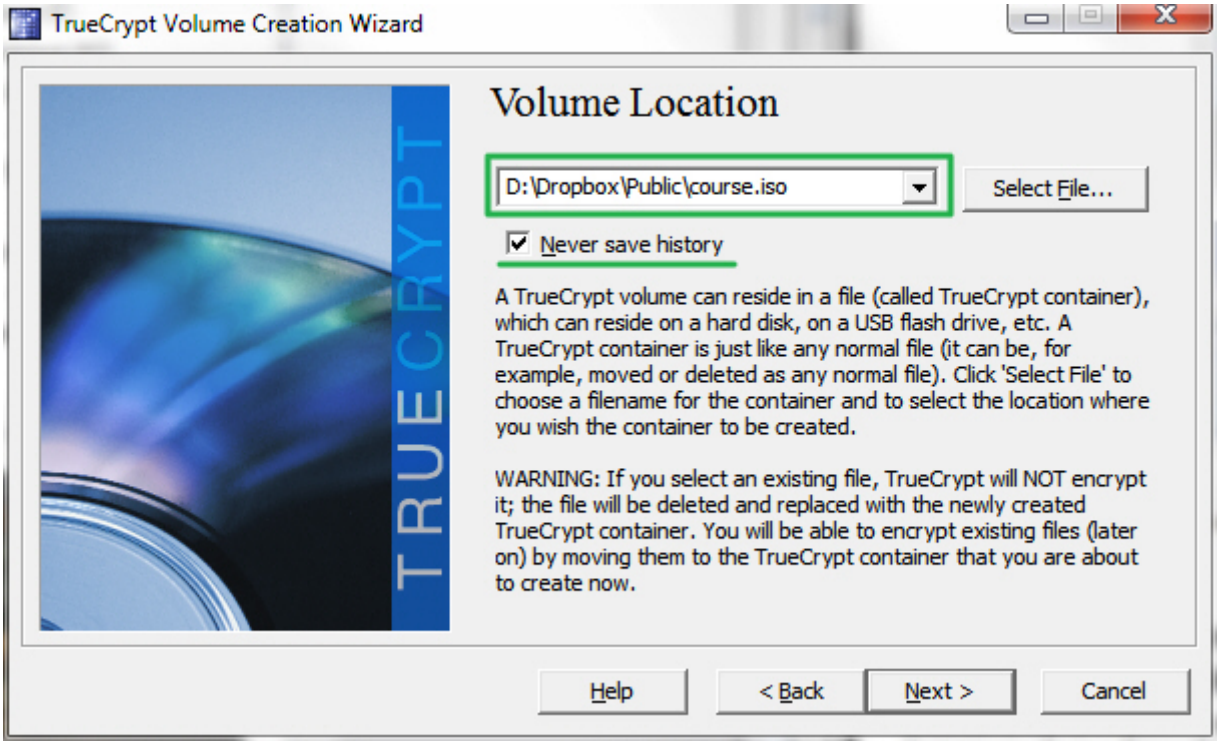
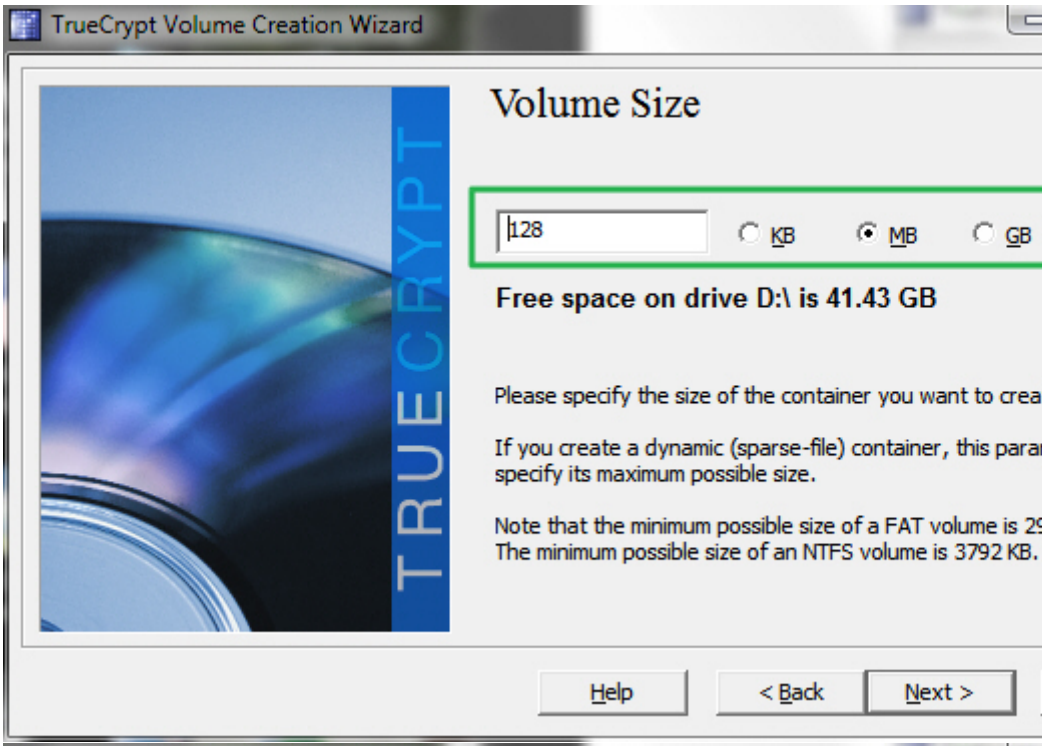


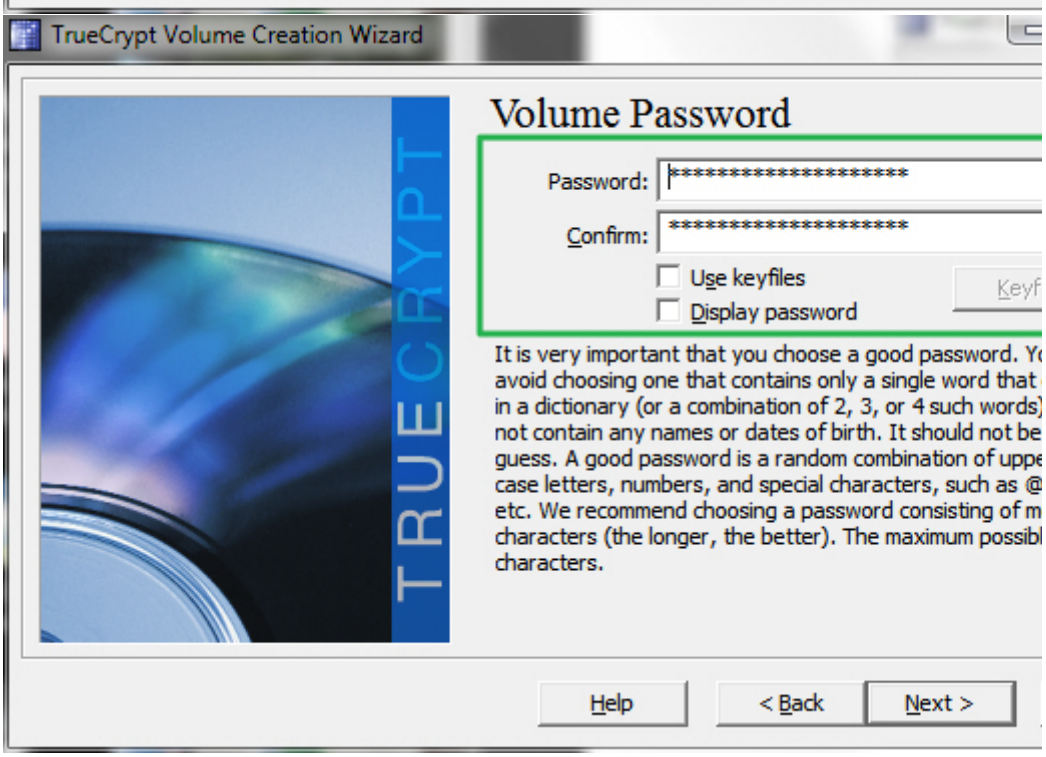
Рис. 2.15. Выбор файла-контейнера

На следующем этапе следует выбрать *алгоритм* шифрования и *алгоритм* хеш-функции, которые будут использоваться при создании тома. Стоит отметить, что от выбора алгоритмов шифрования зависит скорость работы с файлом-контейнером. Если используется каскад алгоритмов, скорость работы будет ниже, чем при использовании одного алгоритма. Оценить *производительность* системы можно при нажатии кнопки "**Оценить производительность**" ("**Benchmark**"), см. рис. 2.16.

Рис. 2.16. Выбор опций шифрования в TrueCrypt

Затем необходимо указать размер тома и придумать надежный *пароль* для доступа к содержимому файла-контейнера. Рекомендации *по* созданию надежного пароля следующие: он не должен содержать слов из словаря, общеизвестных данных пользователя и представлять собой случайную комбинацию из букв, цифр и специальных символов в верхнем и нижнем регистрах. *Длина* пароля не должна быть меньше 20-ти символов, см. рис. 2.17.





Сразу после ввода пароля мастер предложит выбрать тип файловой системы и отформатирует том. Заметим, что хотя на приведённом рис. 2.18 TrueCrypt позволяет выбрать лишь файловые системы *FAT* и *NTFS*, однако это не означает, что нельзя использовать в качестве типа файловой системы один из популярных типов ОС Linux. Указанное ограничение связано с тем, что *программа* TrueCript в данном случае запускалась автором на ОС *Windows*.

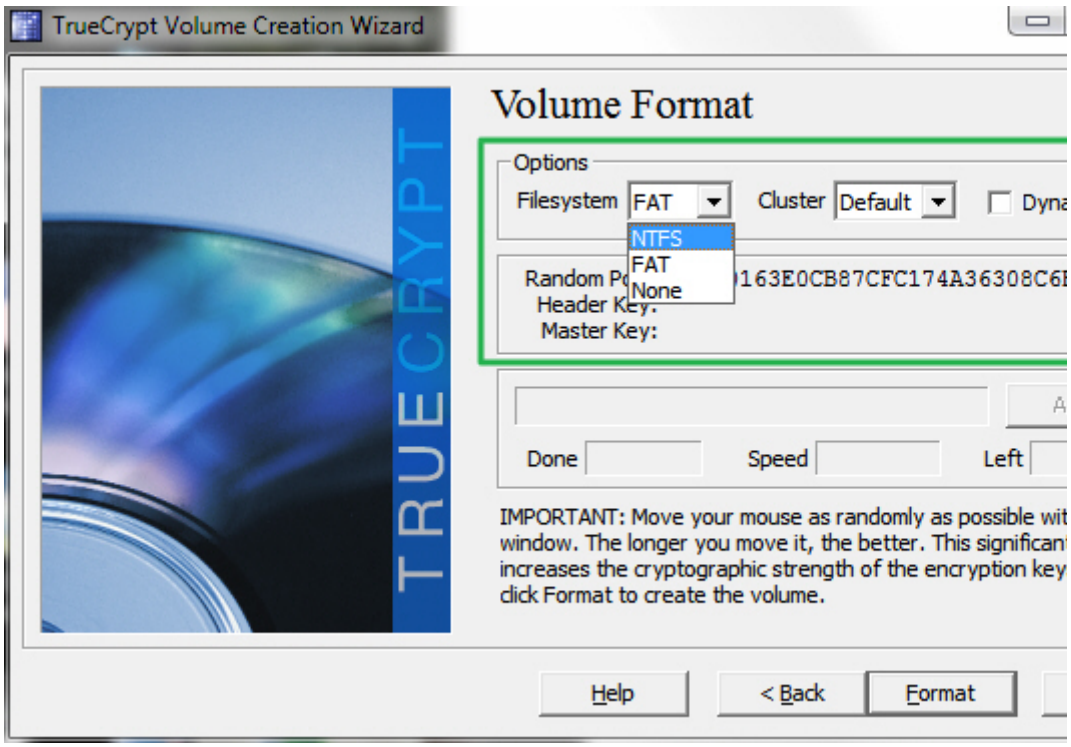


Рис. 2.18. Параметры форматирования тома TrueCrypt

После окончания процесса форматирования зашифрованный *контейнер* готов к использованию.

Работа с зашифрованным файлом-контейнером. Схема работы с зашифрованным файлом-контейнером очень проста: *файл-контейнер* монтируется как виртуальный *диск* через TrueCrypt, после чего с ним можно работать так же, как с обычным диском (или *USB* флеш-накопителем). Для монтирования нужно выполнить действия, указанные на рис. 2.19).

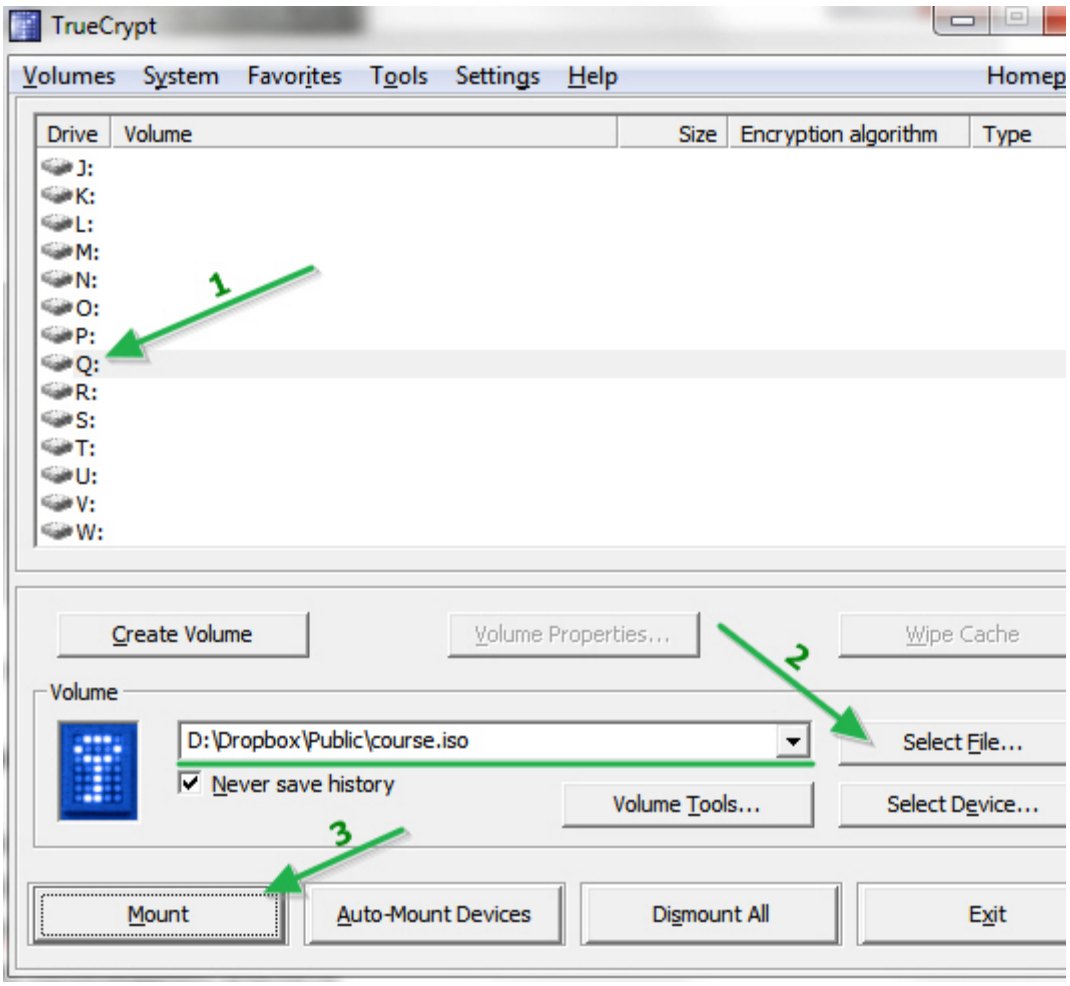
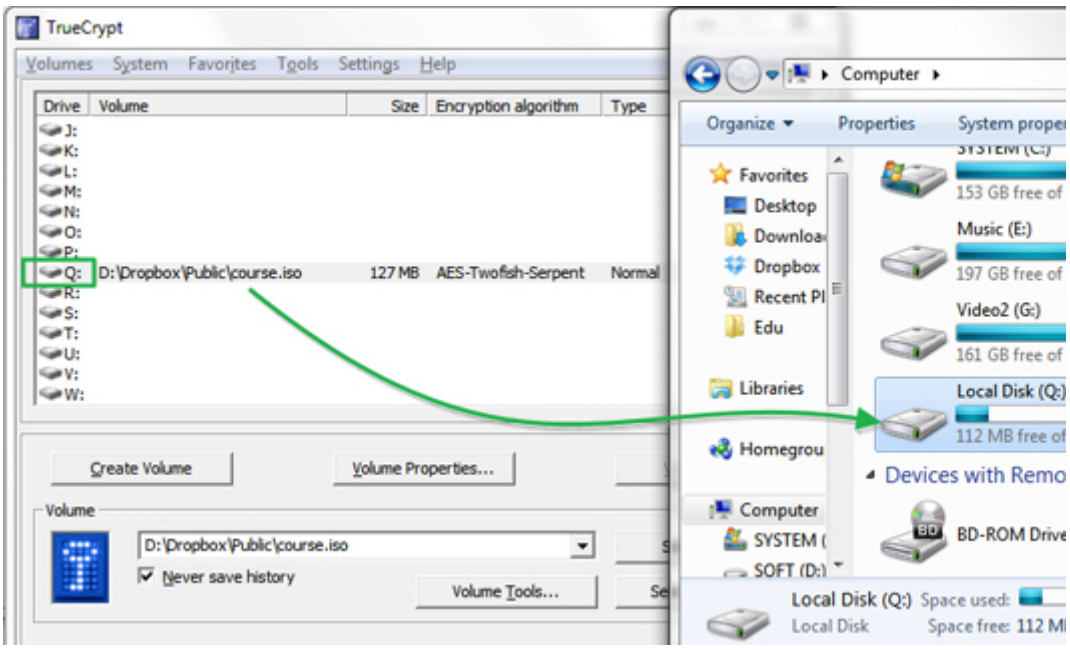


Рис. 2.19. Монтирование зашифрованного файла-контейнера

Перечислим эти действия с подробным описанием:

- выбрать букву диска, к которому необходимо подключить виртуальный зашифрованный диск;
- указать путь к зашифрованному файлу-контейнеру;
- нажать кнопку "**Смонтировать**" для монтирования контейнера и ввести правильный пароль для доступа к зашифрованной информации.

После ввода пароля *файл* будет примонтирован в качестве виртуального диска и будет доступен для дальнейшей работы (рис. 2.20).



увеличить изображение

Рис. 2.20. Зашифрованный файл-контейнер доступен для работы

Шифрование данных при работе с виртуальным диском осуществляется "на лету". Таким образом, не нужно совершать никаких дополнительных действий для шифрования/дешифрования информации. Единственное, что необходимо сделать *по* окончании работы с защищаемыми данными, находящимися на виртуальном диске – демонтировать виртуальный *диск*. Для этого нужно (рис. 2.21):

- указать в главном окне программы букву диска, который необходимо демонтировать;
- нажать на кнопку "**Демонтировать**"

Стоит особо отметить, что перед процедурой демонтирования виртуального диска необходимо закрыть все программы, которые использовали данные на этом диске. В противном случае возможно повреждение данных.

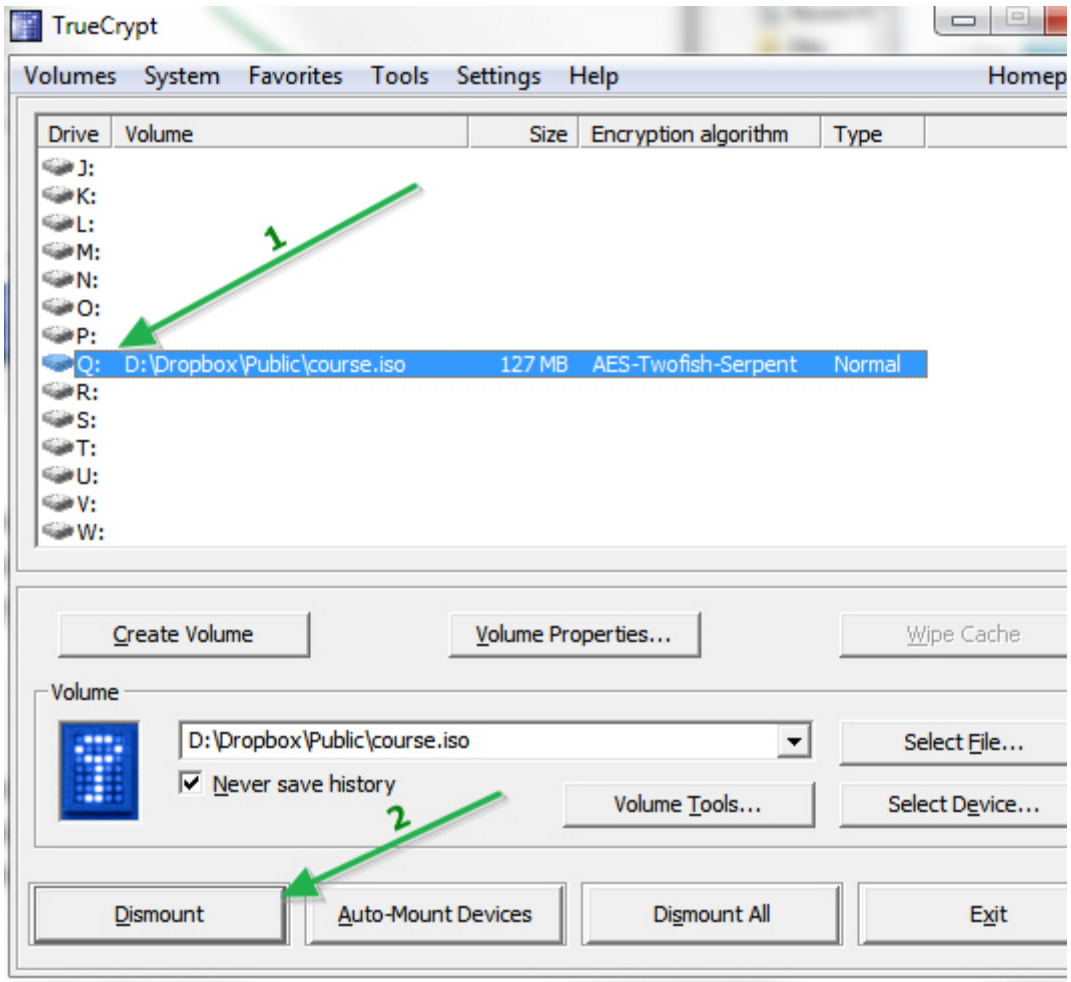


Рис. 2.21. Демонтирование виртуального диска с зашифрованной информацией

Хранение зашифрованного файла-контейнера в "облаке". Если *файл-контейнер* с зашифрованной информацией находится в каталоге, который синхронизируется с сервисом облачного хранения, то сразу после демонтирования этот *файл* будет синхронизирован с облаком. До демонтирования *файл-контейнер* блокируется TrueCrypt-ом, что делает синхронизацию невозможной.

Поскольку TrueCrypt шифрует *файл* блоками *по* 128 *бит*, то синхронизироваться будут только те блоки, которые изменились в результате шифрования информации. Сервисы облачного хранения синхронизируют не файлы целиком, а только изменившиеся блоки, что дает значительный выигрыш *по* скорости передачи данных *по* сравнению с передачей целых файлов.

Необходимо принять во внимание, что *совместная работа* с файлом-контейнером возможна только в последовательном режиме. Это подразумевает обязательное *завершение работы* с контейнером одного пользователя (демонтирование виртуального диска в TrueCrypt) и синхронизацию с облаком перед началом работы с контейнером другого пользователя или с другого устройства. При одновременной работе двух пользователей Dropbox с одним файлом-контейнером создаст две версии этого файла и добавит к имени одного из них пометку "**device conflicted copy date**", где "**device**" - имя устройства, на котором обнаружена конфликтующая копия файла, а "**date**" - дата обнаружения конфликта (см. рис. 2.22).

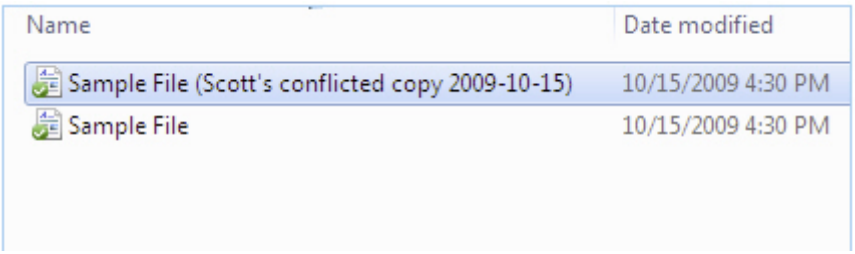


Рис. 2.22. Конфликтующие версии файлов в сервисе Dropbox

В этом случае придется провести дополнительную работу *по* поиску актуальной версии файла-контейнера и синхронизации данных между двумя версиями файла-контейнера.

Вывод – использование *сочетания* мер *по* защите персональных данных, метаданных и современных средств шифрования позволяет надежно защитить важную информацию от посторонних глаз при использовании сервисов облачного хранения.

2.4. Технология "Google Apps for Education"

Google Apps for *Education* (GAfE – это набор облачных приложений, которые предоставляются компанией Google бесплатно для образовательных учреждений в рамках выбранного образовательным учреждением домена [16]. *По* данным на январь 2013 Google Apps для учебных заведений используют более 14 миллионов [17] студентов и преподавателей.

История Google Apps берет начало в феврале 2006 года, когда был впервые представлен сервис электронной почты для использования с собственным доменным именем организации. Приложения Google для учебных заведений стали доступны в октябре 2006 и объединяли в себе, помимо почтовой службы, *приложение* для чата Google Talk, сервис *Calendar* и сервис создания веб-страниц Google Page Creator. *По* состоянию на январь 2013 года приложения Google Apps включают в себя инструменты для коммуникации и планирования, инструменты для коллективной работы, инструменты для повышения эффективности работы [18]. Рассмотрим подробнее каждый из этих инструментов.

Инструменты для коммуникации и планирования:

- Электронная почта "**Gmail**" с размером почтового ящика в 25 ГБ на одного сотрудника/студента образовательного учреждения.
- Служба "**Google Talk**" с возможностью обмена текстовыми, аудио- и видео сообщениями, которые удобно использовать для, например, ведения удалённых лекционных занятий.
- Инструмент "**Календарь**" для управления расписаниями занятий, встреч, рабочим временем с возможностью ведения общего календаря между сотрудниками.

Инструменты для коллективной работы:

- Сервис "**Документы**", позволяющий создавать текстовые документы, таблицы, презентации и специальные опросные формы.
- Сервис "**Диск**" для безопасного хранения любых документов в "облаке" с возможностью общего доступа с бесплатным объемом 5 ГБ дискового пространства на одного пользователя.

- Сервис "**Сайты**" позволяет создавать веб-сайты для учащихся и преподавателей на основе шаблонов.
- Группы Google для создания списков рассылок и предоставления коллегам общего доступа к документам, сайтам и календарям.
- "**Видео для учебных заведений**" – сервис, позволяющий использовать учебные видео материалы в процессе обучения.

Инструменты для повышения эффективности работы:

- Служба поиска по письмам, сообщениям, документам.
- Единая централизованная консоль администрирования сервисов Google образовательного учреждения.

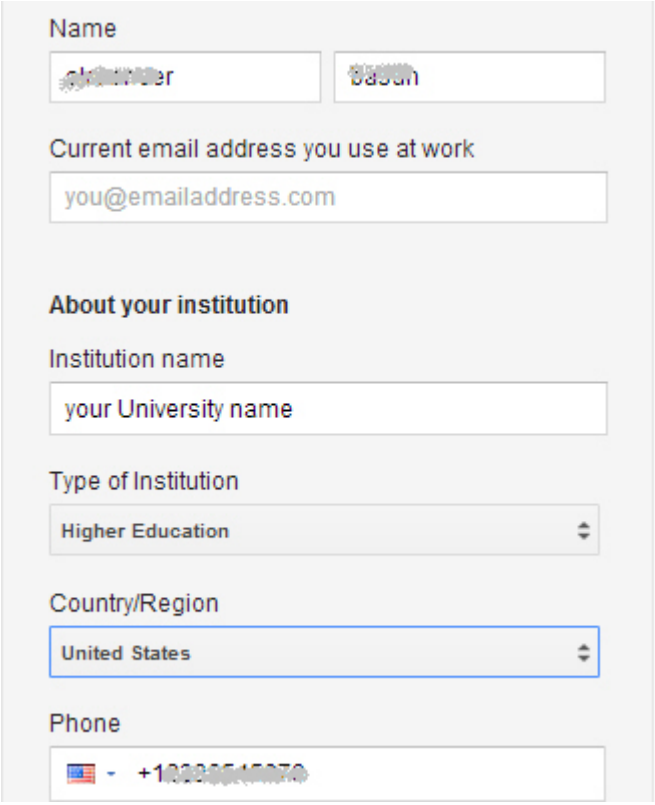
Все вышеперечисленные инструменты являются бесплатными при использовании в образовательных учреждениях. При этом Google Apps for *education* обладают следующими характерными особенностями [19]:

- Осуществляется полное резервное копирование всех данных. Например, если сломается компьютер преподавателя или студента, работу можно будет продолжить через несколько секунд с другого устройства, на котором есть доступ к сети Интернет.
- Предоставляется надежное шифрование и безопасная аутентификация. Все данные при передаче между серверами Google и клиентскими устройствами шифруются с помощью протокола SSL.
- Обеспечивается высокий коэффициент готовности сервисов. Компания Google гарантирует работоспособность служб в течение 99,9% времени.
- Гарантируется соблюдение авторских прав и защита конфиденциальности данных ОУ. Политика конфиденциальности гарантирует, что компания Google не может публиковать или использовать не по назначению личную информацию, размещенную в облачных приложениях Google. Образовательное учреждение является единственным владельцем своих данных и полностью управляет ими.
- Возможность использовать ОУ собственного доменного имени при работе с Google Apps. Для каждого сотрудника учебного заведения регистрируется учетная запись электронной почты в виде **name@youruniversity.com**.

Чтобы начать использовать Google Apps для образовательных учреждений, необходимо проделать несколько шагов:

- Зарегистрировать образовательное учреждение.
- Подать заявку на подключение сервисов GAfE.
- Настроить GAfE в соответствии с особенностями учебного заведения.

Регистрация ОУ в GAfE. Для регистрации необходимо перейти *по* ссылке https://www.google.com/a/signup/?enterprise_product=GOOGLE.EDU и следовать инструкциям мастера регистрации, который предложит заполнить ряд стандартных форм (рис. 2.23).

В процессе регистрации Google Apps for *Education* потребуется подтвердить *право собственности* на доменное имя учебного заведения или зарегистрировать новое доменное имя. В связи с этим рекомендуется воспользоваться услугами соответствующего ИТ-специалиста, ответственного за регистрацию доменного имени вашей организации.

Если доменное имя у учебного заведения отсутствует, то его можно приобрести у любого поставщика *интернет*-услуг. В среднем абонентская *плата* за использование доменного имени составляет 8 долларов США за год использования. При этом следует учитывать, что процесс регистрации нового доменного имени может потребовать вплоть до 24 часов в связи с обновлением информации на корневых серверах *DNS*.