Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Институт информационных технологий

Специальность ИПОИТ

**КОНТРОЛЬНАЯ РАБОТА**

По предмету «Операционные системы»

Студент-заочник 2 курса

Группы: №680971

ФИО: Барковская Ольга Вячеславовна

Тел.: +375(29) 141-14-74

Проверил: Бакунов Александр Михайлович

Минск, 2017

**CОДЕРЖАНИЕ**

[1. Облачные технологии и операционные системы. 3](#_Toc503372417)

[2. Привести исходный код программы, запускающей команду ping google.com и выводящий результаты работы этой команды в Message Box. Использовать анонимные каналы. 9](#_Toc503372418)

[Список литературы 12](#_Toc503372419)

### Облачные технологии и операционные системы.

Облачные вычисления (cloud computing) - это технология распределённой обработки данных в которой компьютерные ресурсы и мощности предоставляются пользователю как интернет-сервис.

Принцип облачных вычислений прост, хотя и непривычен. Компьютер пользователя рассматривается лишь как часть вычислительной системы, непосредственно взаимодействующая с пользователем, в то время как большая часть ресурсов предоставляется серверами в сети (они-то и называются облаком). Самый простой и распространенный пример — хранение файлов на сервисах вроде Dropbox. Выгода от этого двойная: во-первых, компьютер не нуждается в большом объеме локальных дисков, во-вторых, пользователь получает доступ к своим файлам с любого подключенного к Интернету компьютера. Ясно, что хранение данных — далеко не единственный пример облачных вычислений. В облаке могут храниться не только данные, но и приложения. Такой подход применяется, на пример, в «Документах Google».

Облачная модель обещает разработчикам немало преимуществ.

* Доступность. Доступ к информации, хранящейся на облаке, может получить каждый, кто имеет компьютер, планшет, любое мобильное устройство, подключенное к сети интернет. Из этого вытекает следующее преимущество.
* Мобильность. У пользователя нет постоянной привязанности к одному рабочему месту. Из любой точки мира менеджеры могут получать отчетность, а руководители — следить за производством.
* Экономичность. Одним из важных преимуществ называют уменьшенную затратность. Пользователю не надо покупать дорогостоящие, большие по вычислительной мощности компьютеры и ПО, а также он освобождается от необходимости нанимать специалиста по обслуживанию локальных IT-технологий.
* Арендность. Пользователь получает необходимый пакет услуг только в тот момент, когда он ему нужен, и платит, собственно, только за количество приобретенных функций.
* Гибкость. Все необходимые ресурсы предоставляются провайдером автоматически.
* Высокая технологичность. Большие вычислительные мощности, которые предоставляются в распоряжение пользователя, которые можно использовать для хранения, анализа и обработки данных.
* Надежность. Некоторые эксперты утверждают, что надежность, которую обеспечивают современные облачные вычисления, гораздо выше, чем надежность локальных ресурсов, аргументируя это тем, что мало предприятий

Есть, конечно, и минусы:

* Полная зависимость от поставщика облачных сервисов. Для работы облачных приложений требуются большие вычислительные мощности, которые должны быть обеспечены поставщиками ПО. Если разработчик хочет добиться признания у пользователей, ему придется гарантировать бесперебойную работу своих сервисов, а также должный уровень конфиденциальности и устойчивости к взлому;
* Необходимость непременного подключения к Интернету;
* Отсутствие у пользователя контроля над собственными данными. Зато правоохранительные органы могут на законных основаниях получить к ним доступ без ведома пользователя. Если же сервис не отличается особой надежностью — опасностей еще больше. Взломав его, злоумышленник обеспечит себе контроль над данными множества пользователей, а возможно, и над их компьютерами;

Любой облачный сервис подразумевает хранение данных на сервере в Интернете. Различие кроется лишь в том, какие это данные и для чего они применяются. Сегодня все более широкое развитие получает концепция облачного программного обеспечения, предоставляемого пользователям в качестве сервиса.

В облачных технологиях принято выделять три основные модели:

* 1. Инфраструктура как услуга» (IaaS, infrastructure as a service). Пользователи получают базовые вычислительные ресурсы: процессоры и устройства для хранения информации – и используют их для создания собственных операционных систем и приложений. Потребитель не управляет базовой инфраструктурой облака, но имеет контроль над операционными системами, системами хранения, развернутыми приложениями. Возможен ограниченный контроль выбора сетевых компонентов (например, хост с сетевыми экранами).

1. «Платформа как услуга» (PaaS, platform as a service). Пользователи имеют возможность использования облачной инфраструктуры для размещения базового программного обеспечения для последующего размещения на нём новых или существующих приложений (собственных, разработанных на заказ или приобретённых тиражируемых приложений). В состав таких платформ входят инструментальные средства создания, тестирования и выполнения прикладного программного обеспечения — системы управления базами данных, связующее программное обеспечение, среды исполнения языков программирования — предоставляемые облачным провайдером.

Контроль и управление основной физической и виртуальной инфраструктурой облака, в том числе сети, серверов, операционных систем, хранения осуществляется облачным провайдером, за исключением разработанных или установленных приложений, а также, по возможности, параметров конфигурации среды (платформы).

1. «Программное обеспечение как услуга» (SaaS, software as a service). В «облаке» хранятся не только данные, но и связанные с ними приложения, а пользователю для работы требуется только веб-браузер. Потребитель пользуется приложениями провайдера, работающего в облачной инфраструктуре. При этом пользователь не управляет базовой инфраструктурой облака – сетями, серверами, операционными системами, системами хранения, даже индивидуальными настройками приложений за исключением некоторых пользовательских настроек конфигурации приложения. Примерами такого ПО являются Google Docs, Google Calendar и др.

По форме собственности, облачные технологии разделяют на три категории:

1. Публичное облако — это ИТ-инфраструктура, используемая одновременно множеством компаний и сервисов. Пользователи не имеют возможности управлять и обслуживать данное "облако", а вся ответственность по этим вопросам возложена на владельца ресурса. Абонентом, предлагаемых сервисов может стать любая компания и индивидуальный пользователь. Примерами могут служить онлайн-сервисы: Amazon EC2, Google Apps/Docs, Microsoft Office Web.
2. Частное облако — это безопасная ИТ-инфраструктура контролируемая и эксплуатируемая в интересах одной-единственной организации. Организация может управлять частным "облаком" самостоятельно или поручить эту задачу внешнему подрядчику. Инфраструктура может размещаться либо в помещениях заказчика, либо у внешнего оператора (либо частично у заказчика и частично у оператора).
3. Гибридное облако — это ИТ-инфраструктура использующая лучшие качества публичного и приватного облака при решении поставленной задачи. Часто такой тип применяется, когда организация имеет сезонные периоды активности, другими словами, как только внутренняя ИТ-инфраструктура не справляется с текущими задачами, часть мощностей перебрасывается на публичное "облако" (например, большие объемы статистической информации), а также для предоставления доступа пользователям к ресурсам предприятия через публичное "облако".

Получив довольно широкое распространение облачные хранилища стали эволюционировать до облачных сервисов по удаленной обработке информации и доросли до полноценных облачных операционных систем.

Облачные ОС мало чем отличаются в плане пользования ими. Все те же привычные функции, те же привычные окна, папки и файлы. Разница лишь в том, что все ресурсоемкие операции производятся не вычислительными средствами Вашего компьютера, а серверами компании, предоставляющей доступ к облачному сервису. Даже программы могут располагаться за пределами Вашего жесткого диска.

Другими словами, компьютер пользователя должен быть производительным ровно настолько, насколько это необходимо для полноценной работы в сети интернет.

Все же остальные задачи будут выполняться на существенно более мощном сервере. То есть пользователь имеет дело лишь с веб-интерфейсом, с помощью которого дает команды облачному серверу, а затем получает готовый результат.

Самое главное для полноценной работы в среде облачной операционной системы – это наличие доступа к глобальной сети Интернет по стабильному высокоскоростному каналу с широкой пропускной способностью.

Кроме того, помимо размещения всех вычислительных возможностей за пределами Вашего компьютера, все результаты их работы можно хранить также на сервере, не захламляя свой винчестер лишней информацией.

Внедрение облачных операционных систем сгладит различия в производительности между мощными стационарными рабочими станциями, ноутбуками, планшетами и даже карманными гаджетами. Скорость обработки информации будет зависеть исключительно от пропускной способности канала, по которому будет происходить обмен данными между виртуальной операционной системой и сервером.

На сегодняшний день существует множество виртуальных ОС:

* Старейшая из облачных систем – eyeOS. Это open-source виртуальная надстройка над ОС. На сегодняшний день ОС насчитывает шестьдесят семь приложений, список которых легко расширить, путем создания новых с помощью инструмента eyeOS Toolkit. ОС замечательно интегрирована с всевозможными мобильными гаджетами.

Внедрена возможность работы над совместными проектами. Кроме того, технически эту операционную систему eyeOS можно разместить и на персональных серверах, благодаря чему проблема обеспечения приватности и сохранности данных автоматически исчезает.

* CloudMe и CloudTop. CloudMe – новое название сервиса iCloud, который размещался на домене iCloud до весны 2011 года, пока компания Apple не приобрела права на этот домен у шведской студии Xcerion для собственного облачного сервиса и синхронизации устройств.

Сейчас CloudMe и CloudTop – мощная облачная операционная система со стильным дизайном, работающая на все популярных устройствах – Windows, Mac OS, Linux, iPhone, iPad, Android, BlackBerry.

Сервис также построен на языке HTML5 и включает в себя рабочий стол, ярлыки, виджеты, панели управления и кнопку «Пуск». В системе предусмотрены стандартные приложения (почта, календарь, фотоальбом) и дополнительные (около 50), которые устанавливаются через менеджер приложений. Единственным минусом является отсутствие встроенного текстового редактора.

CloudMe предоставляет на выбор несколько тарифных планов:

1) Бесплатный (3 Гб свободного пространства);

2) 25 Гб (4,90 € в месяц);

3) 100 Гб (14,90 € в месяц);

4) 500 Гб (49,90 € в месяц).

Интерфейс полностью локализован на русский язык.

* Google Chrome OS – эта операционная система представляет себя довольно аскетичным графическим интерфейсом, являющимся, по сути, окном браузера Google Chrome. Другими словами, с эстетической точки зрения, мягко говоря, ничего особо выдающегося. ОС получилась максимально легкой, удивительно шустрой, стабильной и функциональной.

Требования к вычислительным ресурсам компьютера, на котором она может быть установлена, ничтожно низки. Единственное требование – широкий интернет-канал.

Система базируется на ядре Linux, однако в своем рвении максимально облегчить и оптимизировать систему, Google провела генеральную уборку, после которой в системе почти не осталось ничего напоминающего об её происхождении

Одним из неоспоримых преимуществ системы является её молниеносная скорость загрузки. К тому же множество сервисов Google, таких как хранилище Google Docs, например, обеспечивают систему всеми необходимыми компонентами для полноценной работы. Ну а количество приложений позволит справиться с задачей любой сложности практически в любой сфере деятельности. А поскольку облачная операционная система Google Chrome OS основана на проекте Chromium OS, а её исходный код открыт – начинают появляться пользовательские сборки, созданные энтузиастами, которые при желании можно попробовать лично.

* Zero PC. Одна из самых функциональных облачных операционных систем в сети Интернет. Сервис работает как в браузере, так имеет готовые приложения для iPhone, iPad, Android и Amazon Appstore. Кроме того, сервис поддерживает авторизацию через социальные сети Facebook, Google +, Twitter.

После регистрации в Zero PC пользователь получает готовый рабочий стол с окнами, ярлыками, папками и кнопкой «Apps» по аналогии с «Пуск» у Windows.

Интерфейс разработан на HTML5 и Java без использования Adobe Flash. При нажатии на правую клавишу срабатывает традиционное контекстное меню, которое выполняет все основные функции операционной системы – копирование, вставка, перемещение, удаление, архивация. Имеется встроенный текстовый редактор Thinkfree Office 4, интернет-браузер, аудио- и видеопроигрыватель.

Для хранения файлов можно использовать облачное хранилище Google Drive или встроенное собственное. В бесплатной версии предоставляется 1 Гбайт свободного места, при переходе на тариф Basic (2,99 $ в месяц) – 5 Гб, а на тариф Pro (9,99 $ в месяц) – целых 50 Гб.

Из минусов можно отметить отсутствие полной локализации на русский язык.

* Silve OS. се поклонникам ОС Microsoft Windows придется по душе облачная Silve OS. Сервис разработан на технологии Silverlight и требует установки дополнительных плагинов в браузерах.

Система включает в себя самые необходимые приложения: браузер, проводник, медиапроигрыватель, блокнот, графический редактор, калькуляторы, онлайн-карты Bing Maps и небольшое количество игр.

Сервис содержит множество ограничений – не предусмотрено контекстное меню, нельзя перетаскивать файлы кликом мыши, нет обратного сохранения файлов с персонального компьютера в облако.

Стоит отметить, что у сервиса существует мобильное приложение для Windows Phone, позволяющее владельцам устройств на базе этой ОС использовать облачную «десктопную» Silve OS на смартфоне.

### Привести исходный код программы, запускающей команду ping google.com и выводящий результаты работы этой команды в Message Box. Использовать анонимные каналы.

PING - PROGRAM.CS

|  |
| --- |
| using System;  using System.Diagnostics;  using System.IO;  using System.IO.Pipes;  using System.Net.NetworkInformation;  namespace ping  {  class Program  {  static void Main()  {  Process pipeClient = new Process();  pipeClient.StartInfo.FileName = "D:\\Session 2017\\OS\\Barkovskaya\\ping\\ping2\\bin\\Debug\\ping2.exe";  using (AnonymousPipeServerStream pipeServer =  new AnonymousPipeServerStream(PipeDirection.Out,  HandleInheritability.Inheritable))  {  // Show that anonymous pipes do not support Message mode.  try  {  pipeServer.ReadMode = PipeTransmissionMode.Message;  }  catch (NotSupportedException e)  {  Console.WriteLine("[SERVER] Exception:\n {0}", e.Message);  }  // Pass the client process a handle to the server.  pipeClient.StartInfo.Arguments = pipeServer.GetClientHandleAsString();  pipeClient.StartInfo.UseShellExecute = false;  pipeClient.Start();  pipeServer.DisposeLocalCopyOfClientHandle();  try  {  // Read user input and send that to the client process.  using (StreamWriter sw = new StreamWriter(pipeServer))  {  sw.AutoFlush = true;  // Send a 'sync message' and wait for client to receive it.  sw.WriteLine("SYNC");  pipeServer.WaitForPipeDrain();  // Send the console input to the client process.  // Console.Write("[SERVER] Enter text: ");  Ping p = new Ping();  string s = "google.com";  PingReply r = p.Send(s);  var results = "";  if (r.Status == IPStatus.Success)  {  results = "Ping to " + s.ToString() + "[" + r.Address.ToString() + "]" + " Successful"  + " Response delay = " + r.RoundtripTime.ToString() + " ms" + "\n";  }  else  {  results = "ERROR PING " + s;  }  sw.WriteLine(results);  }  }  // Catch the IOException that is raised if the pipe is broken  // or disconnected.  catch (IOException e)  {  Console.WriteLine("[SERVER] Error: {0}", e.Message);  }  }  pipeClient.WaitForExit();  pipeClient.Close();  Console.ReadKey();  }  }  } |

PING 2 – PROGRAM.CS

|  |
| --- |
| using System;  using System.IO;  using System.IO.Pipes;  using System.Windows.Forms;  namespace ping2  {  class Program  {  static void Main(string[] args)  {  if (args.Length > 0)  {  using (PipeStream pipeClient =  new AnonymousPipeClientStream(PipeDirection.In, args[0]))  {  // Show that anonymous Pipes do not support Message mode.  try  {  pipeClient.ReadMode = PipeTransmissionMode.Message;  }  catch (NotSupportedException e)  {  Console.WriteLine("[CLIENT] Execption:\n {0}", e.Message);  }  using (StreamReader sr = new StreamReader(pipeClient))  {  // Display the read text to the console  string temp;  // Wait for 'sync message' from the server.  do  {  temp = sr.ReadLine();  }  while (!temp.StartsWith("SYNC"));  // Read the server data and echo to the console.  while ((temp = sr.ReadLine()) != null && !String.IsNullOrEmpty(temp))  {  MessageBox.Show(temp);  }  }  }  }  Console.Write("[CLIENT] Press Enter to continue...");  Console.ReadLine();  }  }  } |

### Список литературы

1. Практическое руководство. Использование анонимных каналов для локального взаимодействия между процессами [Электронный ресурс]. - Режим доступа: https://msdn.microsoft.com/ru-ru/library/bb546102(v=vs.110).aspx
2. Андрей Маркелов, OpenStack. Практическое знакомство с облачной операционной системой, ДМК Пресс, 2016. – 160 с.
3. Что такое облачные технологии и зачем они нужны [Электронный ресурс]. - Режим доступа: https://sonikelf.ru/oblachnye-texnologii-dlya-zemnyx-polzovatelej/