Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Институт информационных технологий

Факультет компьютерных технологий

Кафедра информационных систем и технологий

Контрольная работа по дисциплине

«Основы программной инженерии»

Выполнил студент группы 181073

Наркевич Михаил Викторович

Проверил

Матвеев Андрей Владимирович

Минск 2021

Задание 1.

23. Средства межпроцессного взаимодействия в операционных системах.

ОС Windows обеспечивает программистов механизмами, облегчающими взаимодействие приложений и совместный доступ к данным. Возможности, обеспечиваемые этими механизмами, принято называть межпроцессным взаимодействием (IPC –interprocess communications). Одни механизмы IPC позволяют нескольким специализированным процессам совместно выполнять обработку данных, а другие – распределенную обработку данных в сети.

Приложения, использующие IPC, принято разделять на клиентские и серверные. Клиент – приложение или процесс, запрашивающее сервис у другого приложения или процесса, а сервер – приложение или процесс, предоставляющее такой сервис. Многие приложения, в зависимости от ситуации, являются и клиентами, и серверами. Например, текстовый процессор может выступать в роли клиента при получении некоторых данных от приложения, поддерживающего электронные таблицы, которое в свою очередь, выступает в роли клиента по отношению к приложению, предоставляющему данные для электронной таблицы.

Когда Вы решите, что Ваше приложение должно получить от IPC, Вы должны выбрать соответствующее средство (метод) IPC. Вполне возможно, что приложение будет использовать не один механизм IPC. Определить, что именно Вам нужно использовать, помогут ответы на следующие вопросы: должно ли Ваше приложение взаимодействовать с приложениями, расположенными на одном компьютере или в сети; должно ли Ваше приложение взаимодействовать с приложениями, выполняемыми под управлением разных ОС (MS-DOS, 16-битная Windows или UNIX); должен ли пользователь Вашего приложения сам выбирать другие приложения или этот выбор (или поиск) должно сделать само приложение; должно ли приложение взаимодействовать с другими стандартным способом, таким как команды «вырезать» и «вставить» или же это взаимодействие ограничено специфическими свойствами других приложений; является ли производительность критичным показателем, так как использование всех механизмов IPC связано с определенными ресурсными затратами. является приложение консольным или это должно быть приложение с графическим интерфейсом (GUI). Для использования некоторых средств IPC требуется приложение GUI.

ОС Windows поддерживает такие механизмы IPC: буфер обмена Clipboard; COM модель; Data Copy – передача данных с помощью сообщения WM\_COPYDATA Windows; DDE – динамический обмен данными; File Mapping – файлы, отображаемые в память; Mailslots – объекты ядра Windows «почтовые ящики»; Pipes – каналы; RPC – удаленный вызов процедур; объекты Windows Sockets.

Использование буфера обмена Clipboard. Буфер обмена служит центральным хранилищем данных, разделяемых приложениями. Когда пользователь некоторого приложения выполняет операцию копирования или вырезания, приложение помещает выделенные данные в буфер обмена в одном или нескольких определенных приложением форматов. любое другое приложение может получить эти данные из буфера обмена, выбрав подходящий ему формат представления данных. Буфер обмена является очень малосвязанной областью обмена, в рамках которой приложения могут только согласиться с предложенным форматом данных. Приложения могут находиться на одном и том же компьютере или на разных компьютерах сети.

Задание 2.

23. Понятие веб-сервисов. Разработка с веб-сервисами. Интеграция сервисов.

Прежде всего, веб-сервисы (или веб-службы) — это технология. И, как и любая другая технология, они имеют довольно четко очерченную среду применения.

Если посмотреть на веб-сервисы в разрезе стека сетевых протоколов, мы увидим, что это, в классическом случае, не что иное, как еще одна надстройка поверх протокола HTTP.

С другой стороны, если гипотетически разделить Интернет на несколько слоев, мы сможем выделить, как минимум, два концептуальных типа приложений — вычислительные узлы, которые реализуют нетривиальные функции и прикладные веб-ресурсы. При этом вторые, зачастую заинтересованы в услугах первых.

Но и сам Интернет — разнороден, т. е. различные приложения на различных узлах сети функционируют на разных аппаратно-программных платформах, и используют различные технологии и языки.

Чтобы связать все это и предоставить возможность одним приложениям обмениваться данными с другими, и были придуманы веб-сервисы.

По сути, веб-сервисы — это реализация абсолютно четких интерфейсов обмена данными между различными приложениями, которые написаны не только на разных языках, но и распределены на разных узлах сети.

Именно с появлением веб-сервисов развилась идея SOA — сервис-ориентированной архитектуры веб-приложений (Service Oriented Architecture).

Протоколы веб-сервисов

На сегодняшний день наибольшее распространение получили следующие протоколы реализации веб-сервисов:

SOAP (Simple Object Access Protocol) — по сути это тройка стандартов SOAP/WSDL/UDDI

REST (Representational State Transfer)

XML-RPC (XML Remote Procedure Call)

На самом деле, SOAP произошел от XML-RPC и является следующей ступенью его развития. В то время как REST — это концепция, в основе которой лежит скорее архитектурный стиль, нежели новая технология, основанный на теории манипуляции объектами CRUD (Create Read Update Delete) в контексте концепций WWW.

Безусловно, существуют и иные протоколы, но, поскольку они не получили широкого распространения, мы остановимся в этом кратком обзоре на двух основных — SOAP и REST. XML-RPC ввиду того, что является несколько «устаревшим», мы рассматривать подробно не будем.

Нас в первую очередь интересуют вопросы создания новых веб-служб, а не реализация клиентов к существующим (как правило поставщики веб-сервисов поставляют пакеты с функциями API и документацией, посему вопрос построения клиентов к существующим веб-службам менее интересен с точки зрения автора).

Практическое применение веб-сервисов.

Поскольку речь идет о практическом применении, нам нужно выбрать платформу для построения веб-службы и поставить задачу. Так как автору ближе всего PHP 5, мы и выберем его в качестве технологии для построения службы, а в качестве задачи примем следующие требования.

Допустим, нам необходимо создать службу, предоставляющую доступ к информации о курсах валют, которая собирается нашим приложением, и накапливается в базе данных. Далее посредством веб-сервиса, данная информация передается сторонним приложениям для отображения в удобном для них виде.

Как видим задача довольно проста и, с точки зрения самой службы, ограничивается лишь чтением информации, но в практических целях нам этого будет достаточно.

Интеграция – это организация обмена информацией с сервисом-поставщиком данных. Поставщик данных определяет способ взаимодействия со своим сервисом и описывает ее в документации. Это запросы к интерфейсу API по протоколу http различного предназначения — получение информации, изменение, удаление, добавление сущности.