

Ахмедьянова Я.Е.

*Научный руководитель: канд. техн. наук, Е.П. Догадина  
Муромский институт (филиал) федерального государственного образовательного  
учреждения высшего образования «Владимирский государственный университет имени  
Александра Григорьевича и Николая Григорьевича Столетовых»  
602264, г. Муром, Владимирская обл., ул. Орловская, 23  
e-mail: kaf-eivt@yandex.ru*

### **Разработка автоматизированной системы подготовки документов для административно-воспитательной комиссии**

В каждом учебном заведении существует свой устав, устанавливающий, в том числе, и правила внутреннего распорядка, регулирующие поведение учащихся и преподавателей. Нарушения устава могут быть как незначительные, так и крупные. За несоблюдение данных правил полагаются определенные меры наказания. Например, за неподобающий внешний вид положено замечание или выговор, в случае, если подобное нарушение повторяется многократно, а за драку в стенах ВУЗа студента могут и отчислить. Нарушение фиксирует работник организации и передаёт докладную записку декану. При подготовке к заседанию стараются выяснить причину нарушения, узнать допускал ли ранее студент нарушения устава.

Разрабатываемая автоматизированная информационная система «Административно – воспитательная комиссия» поможет вести учет различного рода нарушений, подготовит набор документов к заседанию административно–воспитательной комиссии и решения по результатам её работы.

Основные функции разрабатываемой системы:

- Учет нарушений устава на территории учебного заведения (учебные корпуса, общежития, база отдыха).
- Подготовка документов к заседаниям административно – воспитательной комиссии.
- Формирование отчетов по результатам заседаний: приказы, протоколы, статистические отчёты.

В докладе представлены результаты моделирования предметной области, приведена модель данных, рассматриваются алгоритмы реализации некоторых функций и основные отчеты.

Володин А.Д.

*Научный руководитель: канд. техн. наук, доцент А.А. Белов  
Муромский институт (филиал) федерального государственного образовательного  
учреждения высшего образования «Владимирский государственный университет имени  
Александра Григорьевича и Николая Григорьевича Столетовых»  
602264, г. Муром, Владимирская обл., ул. Орловская, 23  
e-mail: kaf-eivt@yandex.ru*

### **Обучающий интерактивный WEB портал по фотосъемке и обработке изображений.**

В настоящее время очень развито дистанционное обучение в сети интернет с помощью специальных обучающих сайтов. Обучающий портал – это web портал с советами и инструкциями, необходимыми в процессе обучения фотосъемке и обработке изображений, набором тестовых заданий, вопросов и ответов на них, собранных в одном месте, к которому можно обратиться из любой точки, подключенной к сети интернет.

В обучающем портале будет реализована публикация статей с текстовыми и видеоуроками по фотосъемке и обработке изображений, разделенными по различной тематике в рассматриваемой предметной области. Портал обеспечит реализацию дополнительных обучающих функций за счет интерактивных тестов и заданий, которые необходимо пройти обучающемуся, для повышения своей квалификации. Планируется интеграция с социальными медиа сетями для расширения аудитории обучающего web-портала.

На сегодняшний день существует множество способов создания собственного сайта. К ним можно отнести: использование конструкторов сайтов, подобные wix.com, umi.com, ucoz.com, применение готовых систем управления содержимым сайта CMS, таких как Joomla, Wordpress, Drupal и другие, а также же создание собственной системы управления содержимым сайта с нуля.

Конструкторы позволяют создать сайт с помощью компонентов, который просто помещаются на главную форму. Хостом для сайта в данном случае является сервер конструктора сайтов. Также в адресе страницы имеется указатель конструктора сайтов. Данный способ подходит для людей, которые не сильно знакомы с программированием и созданием сайтов.

CMS – расшифровывается как «Content Management System» («система управления содержимым»). CMS это программа, которая устанавливается на хостинге и выполняет несколько основных функции: первая - показывает пользователям страницы портала, формируя их в момент запроса; вторая – помощь администратору в управлении сайтом. Одной из наиболее подходящих распространенных систем управления содержимым сайта можно считать WordPress. CMS WordPress имеет набор готовых шаблонов для сайтов, которые можно уникализировать под свои требования, используя знания PHP, основы компьютерной верстки и обработки изображений.

Создание сайтов с нуля более сложный и кропотливый процесс. Необходимо создать уникальный дизайн сайта, если страницы отличаются, то создать дизайн для отдельных страниц, а затем перенести его в HTML код. Также необходимо создать каскадные таблицы стилей CSS. Они описывают стили, которые используются на страницах сайта.

Для работы с большим количества уроков, для хранения информации о пользователях в рамках работы над порталом будет удобно использовать БД MySQL. Данная БД интегрируется с web-сервером Apache. К нему обращаются клиенты, как удаленные, так и локальные. Дизайн и контент разрабатываемого обучающего портала будут уникальными. Портал будет выгружен на сервер хостинг-провайдера.

Результатом проделанной работы должен быть обучающий интерактивный интернет ресурс по фотосъемке и обработке изображений, включающий в себя текстовые и видео уроки, задания для учеников, галерею фотографий пользователей.

Воробьева А.В.

*Научный руководитель: доцент Н.Е. Холкина*

*Муромский институт (филиал) федерального государственного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»  
602264, г. Муром, Владимирская обл., ул. Орловская, 23  
e-mail: kaf-eivt@yandex.ru*

### **Разработка автоматизированной системы для медицинских работников дошкольных образовательных учреждений**

При поступлении ребенка в детское дошкольное учреждение родители представляют медицинские документы, отражающие состояние здоровья, сделанные прививки, особенности здоровья, на которые надо обращать внимание (стоит ли на учете по какому-либо заболеванию, часто болеет, имеет острые аллергические реакции в анамнезе...). За годы посещения детского учреждения ведется медицинская карта ребенка, в которой отражаются результаты ежегодных профилактических обследований, история болезней (диагноз, длительность, плановые прививки и скрининги).

Разрабатываемая автоматизированная система предназначена облегчения работы медицинской сестры детского сада и направлена на решение следующих задач:

- Мониторинг выполнения календаря прививок.
- Контроль постановки и реакции на ежегодные туберкулиновые пробы (с учетом графика).
- Контроль за предоставлением справок о возможности посещать детский сад после перенесенной болезни или длительного пропуска.
- Направление на ежегодные плановые осмотры и контроль за их выполнением.
- Ведение учетно-отчетной документации.

Программа позволяет вводить и корректировать данные, формировать стандартные отчеты и бланки (план медицинских осмотров на месяц, год; результаты выполнения плана; выполнение календаря прививок воспитанниками; бланки согласия законного представителя (опекуна) ребенка на медицинское вмешательство и др.)

В докладе представлены результаты моделирования предметной области и данных, приведены алгоритмы реализации некоторых функций и основные отчеты.

Козлов А.В.

*Научный руководитель: канд. техн. наук, доцент А. А. Белов  
Муромский институт (филиал) федерального государственного образовательного  
учреждения высшего образования «Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
602264, г. Муром, Владимирская обл., ул. Орловская, 23  
e-mail: kaf-eivt@yandex.ru*

### **Разработка файловой системы SlimFS**

Файловая система — это порядок, который определяет способ организации, хранения и именования данных на носителях информации. Файловая система определяет как формат содержимого, так и способ физического хранения информации, которую принято объединять в виде файлов.

Файловая система связывает носитель информации и API для доступа к файлам. Когда программа обращается к файлу, то она не имеет представления о том, каким образом расположена информация в файле на носителе, так же как и на каком физическом типе носителя записан файл. Всё, что знает программа, — это имя файла, его атрибуты и размер. Она получает эти данные от драйвера файловой системы. Именно файловая система определяет, как и где будет записан файл на физическом носителе (например, на флеш-памяти или жестком диске).

Большая часть современных файловых систем разработана для использования в настольных и серверных ОС. Эти ФС представляют собой сложную структуру хранения информации, из-за чего доступ к ней занимает продолжительное время. Часто, в микропроцессорных системах, использующих низко производительные микроконтроллеры, разработчику необходимо хранить структурированные данные и иметь возможность быстрого доступа к ним. Почти единственным решением является использование FAT32. Однако эта файловая система имеет существенные ограничения и большую фрагментированность файлов, из-за чего вычислительная сложность алгоритмов чтения/записи файлов может быть высокой.

SlimFS спроектирована специально для использования в процессорах с малой вычислительной мощностью. Она позволяет максимально эффективно использовать пространство накопителя и экономично расходовать ресурс перезаписи. Это достигается благодаря простой структуре хранения файлов и механизму транзакций, описанному ниже.

Каждое изменение в файле можно представить в виде транзакции. Транзакция (англ. transaction) - группа последовательных операций, которая представляет собой логическую единицу работы с данными. В ходе транзакции сохраняются все необходимые изменения файла. Если в FAT32 при обращении к файлу все изменения записываются сразу на носитель, то в SlimFS они записываются в специальный буфер, находящийся в ОЗУ - буфер транзакций. И далее, при обращении к файлу, работа происходит с этим буфером. В конце работы с файлом вызывается функция, которая записывает новое содержимое из буфера транзакций на накопитель. Концепция транзакций заимствована из SQL. Структура хранения файлов в SlimFS приведена на рис. 1.

На рис. 1 представлены следующие структурные блоки:

1) Storage header. Размер = 512 байт. В первом секторе носителя (1 сектор = 512 байт) располагается информация о самом носителе: название носителя, его размер, количество файлов, записанных на нем, размер таблицы размещения файлов.

2) Allocation table. В таблице размещения файлов хранится информация о каждом файле (File header). Ее размер фиксирован и определен в Storage header.

File header. Размер = 512 байт. В каждом заголовке файла хранится информация о названии файла, адреса расположения на физическом носителе, о реальном размере файла, количестве зарезервированных под него секторов, различные атрибуты.

3) Data area. Зона расположения данных файлов. Здесь неструктурированно располагается содержимое файлов.

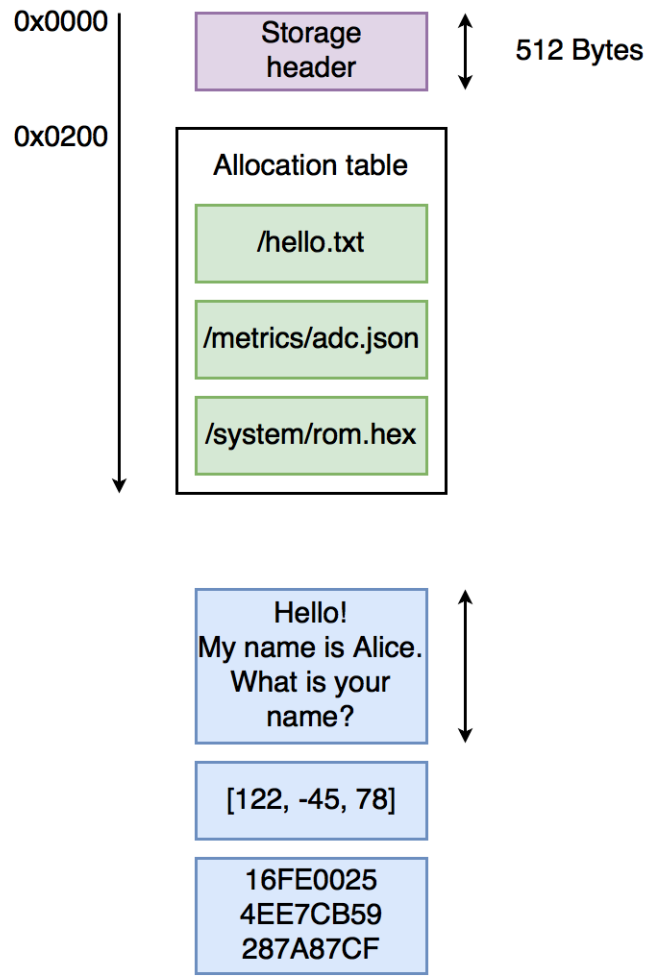


Рис. 1 - Структура хранения файлов в SlimFS

Разработанная файловая система (рис. 1) имеет ряд преимуществ: быстрая работа с файлами, экономичное использование ресурсов перезаписи накопителя, малый размер исходного кода, простое устройство. SlimFS может быть использована в низкопроизводительных процессорных системах с задачами хранения и обработки структурированных данных.

Крайнов В.С.

*Научный руководитель: канд. техн. наук, доцент А.А. Колпаков  
Муромский институт (филиал) федерального государственного образовательного  
учреждения высшего образования «Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
602264, г. Муром, Владимирская обл., ул. Орловская, 23  
e-mail: kaf-eivt@yandex.ru*

### **Разработка многопользовательской клиент-серверной автоматизированной системы обеспечения математических вычислений**

Как известно, современная эра информационных технологий связана с изобретением компьютера. Однако информация, накапливаемая на одном компьютере доступна только человеку, работающему на нем. Конечно, есть возможность передачи информации с помощью магнитных или оптических дисков, но это не самый удобный и быстрый способ. Поэтому со временем возникла идея обмена информацией между компьютерами через линии связи. Введем определение компьютерной сети:

Сеть представляет собой совокупность компьютеров, объединенных средствами передачи данных. Средства передачи данных в общем случае могут состоять из следующих элементов: связанных компьютеров, каналов связи (спутниковых, телефонных, цифровых, волоконно-оптических, радио - и других), коммутирующей аппаратуры, ретрансляторов, различного рода преобразователей сигналов и других элементов и устройств.

Архитектура сети ЭВМ определяет принципы построения и функционирования аппаратного и программного обеспечения элементов сети.

Современные информационные системы продолжают возникшую в конце 70-х гг. тенденцию распределенной обработки данных. Начальным этапом развития таких систем явились многомашинные ассоциации – совокупность вычислительных машин различной производительности, объединенных в систему с помощью каналов связи. Высшей стадией систем распределенной обработки данных являются компьютерные (вычислительные) сети различных уровней – от локальных до глобальных.

Цель работы – разработка многопользовательского клиент-серверного приложения для организации математических вычислений над матрицами.

Существует несколько аналогов, предоставляющих данную функцию.

1. Mathcad — система компьютерной алгебры из класса систем автоматизированного проектирования, ориентированная на подготовку интерактивных документов с вычислениями и визуальным сопровождением, отличается легкостью использования и применения для коллективной работы.

Mathcad имеет интуитивный и простой для использования интерфейс пользователя. Для ввода формул и данных можно использовать как клавиатуру, так и специальные панели инструментов. Работа осуществляется в пределах рабочего листа, на котором уравнения и выражения отображаются графически, в противовес текстовой записи в языках программирования.

2. MATLAB (сокращение от англ. «Matrix Laboratory», в русском языке произносится как Матлаб) — пакет прикладных программ для решения задач технических вычислений и одноименный язык программирования, используемый в этом пакете. MATLAB используют более 1 000 000 инженерных и научных работников, он работает на большинстве современных операционных систем, включая Linux, Mac OS, Solaris (начиная с версии R2010b поддержка Solaris прекращена) и Microsoft Windows.

Mathcad и MATLAB являются очень мощными приложениями, позволяющие выполнять очень сложные математические вычисления, но из-за этого они требуют для работоспособности мощные компьютеры, так как операции выполняются на этом же компьютере, что является их недостатком. Так же не каждому пользователю требуется такой большой набор функций. MATLAB в свою очередь требует знания одноименного языка

программирования, что затруднит работу пользователей, вообще не знакомых с основами программирования.

3. Интернет ресурсы на примере сайта matrixcalc.org. Данные ресурсы удобны тем, что не загромождают память компьютера и все операции выполняются на сервере. В связи с этим нет необходимости в мощном компьютере. Но интернет ресурсы требуют обязательного выхода в интернет, что не позволит воспользоваться им без выхода в глобальную сеть, или же заставит долго ждать ответа с интернетом с маленькой скоростью.

Разрабатываемое приложение будет иметь клиент-серверную архитектуру, что позволит выполнять операции на сервере, а все необходимые данные будут отсылаться с клиента. Это позволит устанавливать приложение на слабые компьютеры. Так же интерфейс приложения не будет требовать знания основ программирования. Пользователю потребуется только заполнить исходные данные (матрица) и выбрать нужную операцию. Но клиент-серверное приложение имеет недостаток в том, что при отсутствии подключения приложение не будет работоспособно.

Задачи работы:

1. Разработать архитектуру системы, определить основные потоки данных.
2. Выбрать базовые протоколы передачи данных.
3. Разработать протоколы взаимодействия компонентов системы.
4. Разработать серверную часть системы.
5. Разработать клиентскую часть системы.
6. Провести исследование производительности и надежности передачи данных в разрабатываемой системе.

Клиентская часть будет отправлять серверу данные, необходимые для выполнения операций, серверная – принимает данные от клиента, выполняет операции с принятыми данными и отправляет результат клиенту.

Ниже приведена схема взаимодействия подсистем.

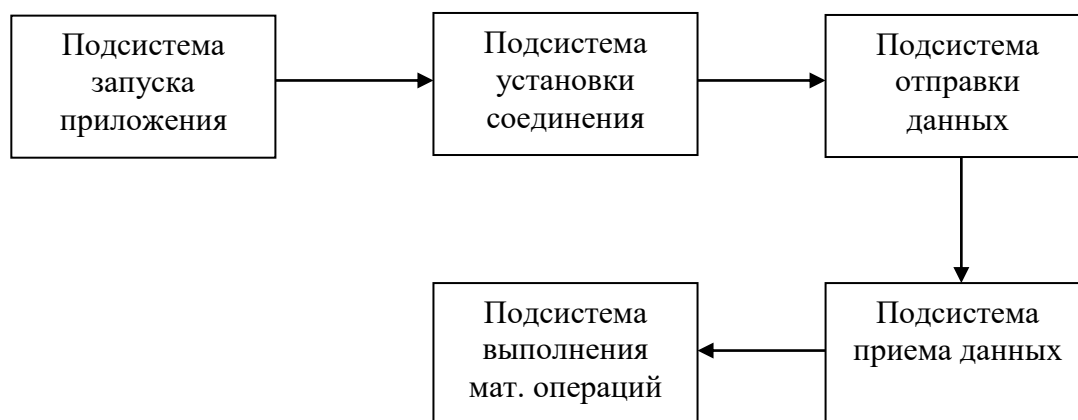


Рис. 1. Схема взаимодействия подсистем

### Литература

1. Олифер В.Г., Олифер Н.А. «Компьютерные сети. Принципы, технологии, протоколы» - Санкт-Петербург: Питер, 2010. – 943 с.
2. Стивенс У.Ричард «Протоколы TCP/IP. Практическое руководство» - BHV, Санкт-Петербург, 2003 г. – 672 с.
3. Шкрыль А.А. «Разработка клиент-серверных приложений в Delphi.» – Санкт-Петербург: БХВ-Петербург, 2006. – 480 с.
4. Страуструп Б. «Язык программирования C++» - М.: ЗАО Издательство БИНОМ, 2011 г. – 1136 с.

Лебедев М.С.

*Научный руководитель: канд. техн. наук, доцент А.А. Колпаков  
Муромский институт (филиал) федерального государственного образовательного  
учреждения высшего образования «Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
602264, г. Муром, Владимирская обл., ул. Орловская, 23  
e-mail: kaf-eivt@yandex.ru*

### **Разработка многопользовательской клиент-серверной автоматизированной системы чтения, хранения и каталогизации электронных книг**

Компьютерные сети появились сравнительно недавно, в конце 60-х годов прошлого столетия. Естественно, что компьютерные сети унаследовали много полезных свойств от других, более старых и распространенных телекоммуникационных сетей, а именно телефонных. В то же время компьютерные сети привнесли в телекоммуникационный мир нечто совершенно новое — они сделали общедоступными неисчерпаемые объемы информации, созданные цивилизацией за несколько тысячелетий своего существования и продолжающие пополняться с растущей скоростью в наши дни [1].

В настоящее время очень распространенным явлением стали электронные библиотеки. Электронная библиотека — упорядоченная коллекция разнородных электронных документов (в том числе книг), снабженных средствами навигации и поиска. Может быть веб-сайтом, где постепенно накапливаются различные тексты (чаще литературные, но также и любые другие, вплоть до компьютерных программ) и медиафайлы, каждый из которых самодостаточен и в любой момент может быть востребован читателем. Электронные библиотеки могут быть как универсальными, стремящимися к наиболее широкому выбору материала, так и специализированными [2].

По объему и возможностям доступа посетителей к раритетам электронная библиотека сопоставима с крупными государственными библиотеками. Например, Российская государственная библиотека (РГБ) имеет 40 млн. единиц хранения, а электронная библиотека Мошкова — 30 тысяч файлов. Первую посещают 3-4 тыс. читателей в день, а вторую — около 20 тыс. Художественную и научную литературу в Интернет ежедневно читают 150-200 тыс. человек. Себестоимость поддержания электронной библиотеки не сравнима с затратами на обычную библиотеку. Нет необходимости платить за аренду помещения, содержать большой штат специалистов и охранников, обслуживающий персонал. Нужны только компьютер и человек, который с его помощью размещает книги [2].

В отличие от бумажных, электронные книги могут быть копированы необходимое количество раз практически мгновенно. В зависимости от целей и принципов библиотеки, а также действия авторского права в отношении некоторых из представленных книг, они могут быть доступны к прочтению на сайте или для свободного скачивания.

Все это позволяет работать с одной и той же книгой любому количеству читателей, решая вопрос поиска специальной литературы.

Поиск книги по традиционной библиотечной картотеке, особенно если неизвестны ее полные данные, может занять немало времени. Конечно, скоротечное и развитие памяти поможет ускорить процесс, но несомненно — электронный каталог гораздо эффективнее.

Главные преимущества электронного каталога:

- самостоятельный поиск книги пользователем;
- легкий поиск при неполных известных данных;
- облегчение учета книг.

Немаловажен последний пункт. Заказ через электронный каталог может и должен использоваться для определения нахождения книги — была ли она принесена в читальный зал, выдана на руки или осталась в хранилище. В сочетании с системой электронного читательского билета, это позволяет контролировать каждую книгу, предотвращая хищения ценных экспонатов.



Данный проект посвящен разработке многопользовательского клиент-серверного приложения обеспечения доступа к электронным книгам.

Разрабатываемое приложение должно обеспечивать пользователям возможность чтения книг, хранящихся на сервере. При этом пользователь не должен иметь возможности скачать книгу. Запрет на скачивание книг обусловлен соображениями защиты авторских прав. Защита авторских прав, как и защита смежных, патентных и других прав интеллектуальной собственности обеспечивается предусмотренными законодательством способами с учетом существа нарушенного права и последствий нарушения этого права (пункт 1 статьи 150 ГК РФ).

Немаловажным требованием к разрабатываемой системе является возможность доступа к электронным книгам только для зарегистрированных пользователей.

Разрабатываемое приложение должно быть реализовано на основе архитектуры клиент-сервер. Выбор данной архитектуры обусловлен тем, что она позволяет избежать создания на клиентских компьютерах копии информационных ресурсов, хранящихся на сервере. При этом все вычисления выполняются на сервере, что позволяет снизить нагрузку на клиентские компьютеры. Кроме того, выбор клиент-серверной архитектуры позволит исключить возможность скачивания книг пользователями.

В функции серверной части входит:

- хранение списка зарегистрированных пользователей;
- авторизация пользователей;
- предоставление информации о доступных книгах;
- отправка пользователю запрашиваемых фрагментов книг.

В функции клиентской части входит:

- проведение процедуры авторизации;
- отображение списка доступных книг;
- запрос выбранного фрагмента книги;
- хранение информации о прочитанных книгах с возможностью ее просмотра.

В процессе выполнения работы необходимо решить следующие задачи:

- выбрать и обосновать архитектуру системы;
- выбрать базовые протоколы передачи данных;
- разработать алгоритмы работы серверной и клиентской частей системы;
- разработать протоколы взаимодействия клиентской и серверной частей системы;
- разработать приложения, реализующие основные функции клиентской и серверной частей системы;
- провести тестирование разработанной системы;
- исследовать производительность и надежность данных в разработанной системе.

### Литература

1. Олифер В.Г., Олифер Н.А. «Компьютерные сети. Принципы, технологии, протоколы» - Санкт-Петербург: Питер, 2010. – 943 с.
2. Стивенс У.Ричард «Протоколы TCP/IP. Практическое руководство» - BHV, Санкт-Петербург, 2003 г. – 672 с.
3. Шкрыль А.А. «Разработка клиент-серверных приложений в Delphi.» – Санкт-Петербург: БХВ-Петербург, 2006. – 480 с.
4. Страуструп Б. «Язык программирования C++» - М.: ЗАО Издательство БИНОМ, 2011 г. – 1136 с.

Меньшов И.С., Миловидов А.Е.

*Научный руководитель: канд. техн. наук, доцент А.Ю. Проскуряков  
Муромский институт (филиал) федерального государственного образовательного  
учреждения высшего образования «Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
602264, г. Муром, Владимирская обл., ул. Орловская, 23  
e-mail: menshov.ivan1998@gmail.com*

### **Автоматизированный торговый робот для операций на спотовом рынке криптовалют**

Торгуя на биржевом рынке можно получить прибыль на поочерёдных подъемах и падениях цены валютной пары. Но это сопровождается огромным риском, так как человек не может с необходимой точностью предугадать то, как будет изменяться цена. Также зачастую нужно за секунду среагировать на изменение курса, подсчитать и выставить ордера, что представляет сложность ввиду человеческого фактора. К тому же человек не может постоянно находиться у терминала и вести торги. Для этого возможно разработать программную модель робота для автоматизации торгов, который будет удовлетворять вышеизложенным условиям.

Программирование робота производится на языке JL Script. Это упрощённый язык программирования, в котором полностью поддерживаются функции и методы JavaScript. Он разработан для создания сложных алгоритмических стратегий и автоматизации торговли.

Для адаптации расчетов при выполнении торговых операций применен математический статический анализ [3] в частности расчет средней стоимости актива с учетом их объёмов. При запуске робот выставляет два ордера, основываясь на цене в данный момент времени. Первый ордер на покупку в размере 20% от текущего депозита по цене  $m\%$  меньше, чем текущая. Второй ордер выставляется на продажу 20% баланса на  $m\%$  больше, чем цена в данный момент.

Дальнейшие действия робота будут осуществляться после выполнения хотя бы одного из выставленных ордеров. Как только это происходит, второй ордер снимается, рассчитываются и выставляются новые ордера по формуле (1), в зависимости от того сколько ордеров подряд было исполнено в одну сторону.

$$x = \frac{c_1 * a_1 + c_2 * a_2 + \dots + c_n * a_n}{a_1 + a_2 + \dots + a_n} * (1 \pm m), \quad (1)$$

где  $x$  – цена следующей покупки/продажи,  $c_1, c_2, \dots, c_n$  – цены покупок/продаж выполненных в одну сторону,  $a_1, a_2, \dots, a_n$  – объем покупки/продажи выполненных в одну сторону,  $m$  – коэффициент изменения цены.

На рисунке 1 представлен график, на котором изображен резкий импульс вверх цены валютной пары, которую человек не мог предугадать и успеть отреагировать на него. И скорее всего, потерял бы большую часть прибыли, так как слишком поздно запустил терминал и движение цены показало глубокую коррекцию от роста.

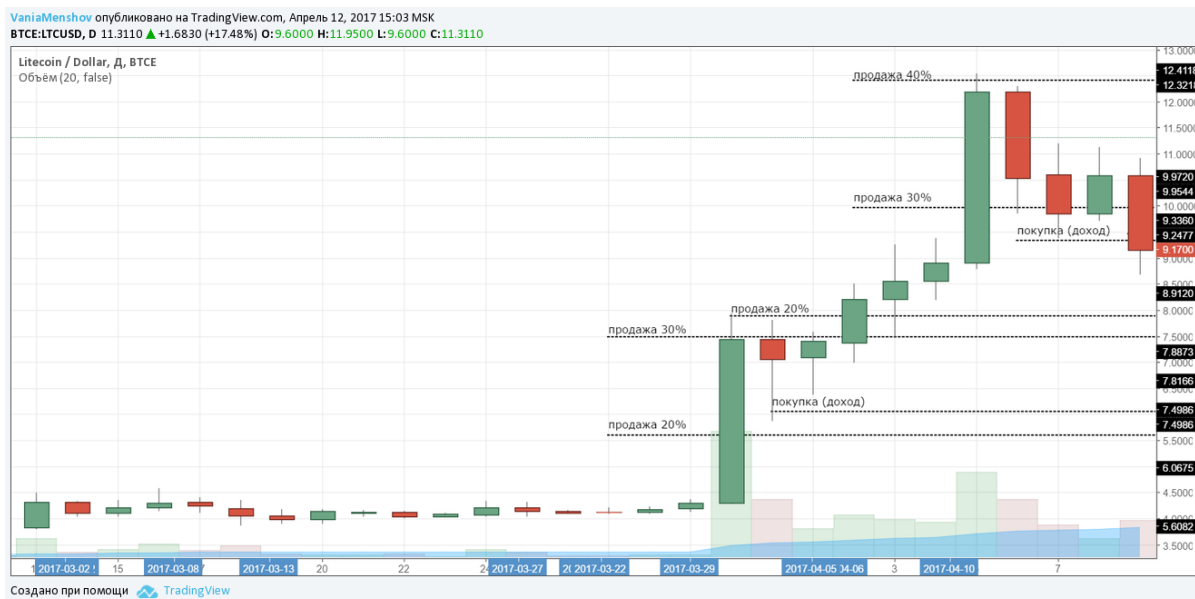


Рис.1. Пример работы робота на графике валютной пары (LTC/USD)

Робот, выполняет периодические торговые операции, постепенно повышая объём и высчитывая среднее значение выполненных операций, в момент падения цены выполнит покупку, при которой получит доход при любых изменениях ценовых показателей. Также на графике указаны уровни, на которых робот совершил торговые операции по заданному программному алгоритму. Наиболее высокая эффективность разработанной автоматизированной системы достигается при установке робота на облачную виртуальную машину с постоянным бесперебойным режимом работы.

### Литература

1. A. Proskuryakov. Intelligent System for Time Series Forecasting. XII International Symposium Intelligent Systems 2016, INTELS 2016, 5-7 October 2016, Moscow, Russia. Procedia Computer Science. [ <http://dx.doi.org/10.1016/j.procs.2017.01.122> ] Volume 103, 2017, Pages 363–369.
2. TradingView - бесплатные графики акций, котировки и биржевые графики онлайн [Электронный ресурс]. URL: <https://ru.tradingview.com/> (дата обращения 10.04.2016).
3. Кобзарь, А.И. Прикладная математическая статистика. Для инженеров и научных работников/ А.И. Кобзарь. ФИЗМАТЛИТ. – 2006. – 816 с.

Сорокина О.О.

*Научный руководитель: доцент Н.Е. Холкина*

*Муромский институт (филиал) федерального государственного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»  
602264, г. Муром, Владимирская обл., ул. Орловская, 23  
e-mail: kaf-eivt@yandex.ru*

### **Разработка автоматизированной системы «Методическая обеспеченность дисциплин»**

Для организации успешной подготовки специалистов в ВУЗе, преподаваемые дисциплины должны быть обеспечены актуальными учебниками, учебными пособиями, методическими указаниями для выполнения лабораторных, практических и курсовых работ. Контроль наличия методической обеспеченности дисциплин каф. и должна обеспечить разрабатываемая автоматизированная система.

Программа должна обеспечить загрузку данных из учебных планов, ввод информации об учебной литературе с указанием срока актуальности данного издания (последние 5 лет, 10 лет, бессрочно). На основании полученных данных надо сформировать план издания методической литературы на текущий учебный год (или на ближайшие 3 года). Назначить исполнителей на каждый запланированный пункт и сгенерировать план-отчет. Контролировать исполнение плана можно меняя статус запланированной методической разработки (запланировано, в разработке, сдано в издательство, издано). Итоговый (годовой) и промежуточные отчеты о ходе выполнения плана должна быть возможность сгенерировать и напечатать.

В докладе представлены результаты моделирования предметной области и данных, приведены алгоритмы реализации некоторых функций и основные отчеты.

Ярошенко А.И.

Научный руководитель: к.т.н. Проскуряков А. Ю.

Муромский институт (филиал) федерального государственного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»  
602264, г. Муром, Владимирская обл., ул. Орловская, 23  
e-mail: artawower@protonmail.com

### Микро сервис для автоматизации генерации валидных email писем из документов Microsoft Word

Электронная почта представляет собой технологию по пересылке и получению электронных сообщений между пользователями компьютерной сети. На сегодняшний день это один из популярнейших, и в тоже время, долговечных способов коммуникации в сети. Данный метод также широко используется различными компаниями для массовой коммуникации между сотрудниками организации. Однако постоянное создание электронных писем требует значительное время на стилизацию и оформление элементов.

Актуальность данной работы заключается в автоматизации процесса создания валидных электронных писем. Данная тема актуальна для крупных компаний занимающихся массовой рассылкой email писем. Однако на большинстве современных ресурсах (MailChimp, Sendulse, UniSender) предоставляющих такие услуги присутствует только визуальный редактор позволяющий генерировать html код с помощью JavaScript в браузере, заполняя его статическим контентом вручную. Данный микро сервис, в отличии от аналогов, позволяет обрабатывать большие объемы текста с помощью специального шаблонного языка [1, 2], переводя документы формата .docx, txt в корректный html код.

На рис. 1 представлена схема работы сервера при обработке полученных данных со стороны клиента.

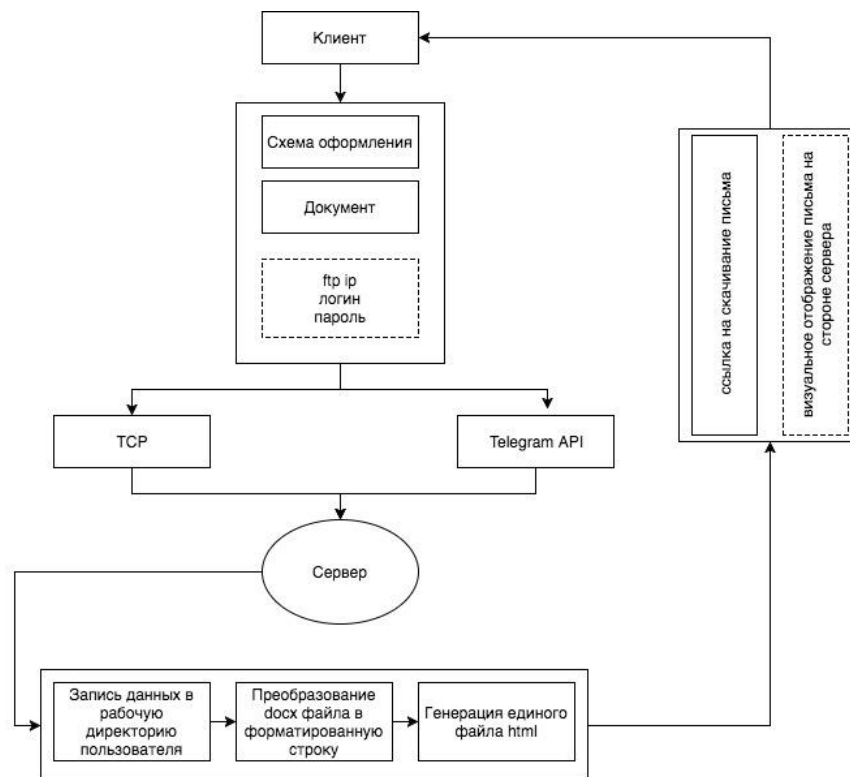


Рис. 1. Структура работы сервера по обработке полученных данных

Как видно на рис. 1, принцип взаимодействия пользователя с программой может осуществляться двумя способами, через http сокет [3], либо через Telegram API.

Так как управление осуществляется с помощью встроенных функций, возможно использовать протокол передачи данных с помощью специального API сервиса, напрямую отправляя запросы с других ресурсов, что делает систему мобильной и легко интегрируемой в уже существующие проекты.

Данные методы отправляют запрос на сервер с необходимым документом, а также с рядом необязательных аргументов [3], для последующего преобразования в html письмо с inline каскадной таблицей стилей. В ответ пользователь получит ссылку на загруженный файл на указанный в аргументах сервер, а в случае его отсутствия выведет в default режиме ссылку на загрузку файла.

Для хранения небольших данных пользователя, используется реляционная база данных SQLite. Данные пользователя собираются с использованием регистрационного ключа, который предоставляется при регистрации на его электронную почту. При входе в систему клиенту доступна возможность комбинировать доступные темы для стилизации с имеющимися на сервере документами, а также просматривать в режиме реального времени результат генерации.

К особенностям функционала данного приложения можно отнести ряд вспомогательных параметров и функций: автоматическая отправка данных на сервер в рабочую область, генерация html шаблона с использованием специального языка для форматирования на JavaScript, подгрузка изображений из файлов docx в отдельную папку.

По результатам проделанной работы представлено программное обеспечение в виде веб приложения для повышения производительности при рассылке электронных писем, а также для улучшения мобильности при смене frontend составляющей проекта. Кроме того, данная тема является актуальной для пользователей не знакомых с языком гипертекстовой разметки и правилами оформления таблиц каскадных стилей, позволяя клиенту предоставлять формальный набор правил и текстовое содержание письма.

На данный момент данная технология активно применяется при массовом оповещении клиентской базы в стартап проекте OldBriefcase [4]. При активной поддержке и разработке дополнительного функционала, данное программное обеспечение в значительной степени ускорило процесс кооперирования членов организации, при создании новых публикаций и редактировании уже имеющегося материала.

### Литература

1. Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
2. Хахаев И.А. Практикум по алгоритмизации и программированию на Python. – М.: Альт Линукс, 2010. — 126 с.
3. Мигель Гринберг Разработка веб-приложений с использованием Flask на языке Python. – МСК.:ДМК Пресс, 2014. – 272 с.
4. OldBriefcase. [Электронный ресурс] / инвестиционные идеи для сохранения сбережений с применением науки и современных компьютерных технологий, -М.: Справочно-информационный портал OldBriefcase, 2016. Режим доступа: <https://old-briefcase.ru>. - Загл. с экрана.