

# Летняя молодежная школа-стажировка Intel-ННГУ

Июль-август 2009 г.

---

## Список задач производственной составляющей стажировки

### **Задача 1: Исследование возможностей и применения технологии Access Control Engine (ACE) в системе SAP CRM**

Выполнение данной работы дает возможность расширить свои знания о стандартных и нестандартных технологиях в системах SAP, конфигурации и бизнес-процессах системы SAP CRM, а также получить углубленный опыт программирования на языке ABAP.

Для решения задачи необходимо: исследовать технологию Access Control Engine (ACE), применяемую как альтернативный подход к управлению доступом и правами пользователей в системе SAP CRM.

Планируемые результаты:

*Основной:* Подготовить шаблоны кода и/или системных настроек для решения типовых задач по разграничению доступа пользователей к различным объектам (мастер-данным, транзакциям, иерархиям) в зависимости от позиции пользователей в организационной структуре, территориальной иерархии, адреса и т. д.

*Дополнительный:* Несколько примеров и пособий по применению данной технологии для демонстрации ее возможностей.

Источники информации: Документация по данной технологии.

Требуемый уровень подготовки: Базовые теоретические и практические знания о системах SAP.

\* Проект осуществляется совместно с [Центром компетенций в области современных ERP систем](#), созданным на базе Государственного Университета – Высшей Школы Экономики, Нижегородский филиал.

### **Задача 2: Разработка интранет-приложения с использованием Silverlight**

Решая задачу, Вы сможете

- развить навыки разработки надежных бизнес-приложений;
- углубить знания о паттернах объектно-ориентированного проектирования;
- научиться практикам экстремального программирования.

Предлагается исследовать технологию Silverlight на предмет применимости в бизнес-приложениях и разработать интранет-приложение с богатым пользовательским интерфейсом (деловая графика).

Варианты прикладной области:

- Управленческие отчеты по результатам сервисной деятельности.
- Сравнение показателей качества исходного кода интеловских приложений.

Работа над проектом предполагает

- изучение доступной документации по данной технологии;
- составление нескольких примеров применения Silverlight для демонстрации возможностей;
- сравнительный анализ Silverlight и альтернативных возможностей (например перенос бизнес-логики в VSTO);
- собственно разработка прототипа приложения;
- подготовка заключения и рекомендаций.

Планируемый результат:

Готовое приложение и типовые рекомендации по использованию технологии Silverlight для разработки интранет-сайтов (основные проблемы программиста, возможные шаблоны архитектуры, возможность разрабатывать через тестирование (практика TDD), скорость разработки и т. д.).

### **Задача 3: Сравнительный анализ программ для измерения покрытия кода (code coverage tools)**

Задача заключается в изучении различных средств измерения покрытия кода (code coverage tools), представленных на рынке, и сравнении их с разрабатываемым прототипом.

Планируемый результат:

- Выделение общих и специфических качеств (features) каждого из продуктов.
- Оценка производительности и качества результатов для распространенных коммерческих приложений.
- Оценка поведения программы при стрессовых нагрузках (стабильность).
- Оценка поведения и качества результатов при граничных условиях (corner cases), включает написание самостоятельных тестовых сценариев.

Требуемый уровень подготовки: Базовые знания структуры исполняемого файла в ОС Windows & Linux, знание C++, навыки работы в Windows & Linux, знания архитектур IA32/Intel64 и их различий.

### **Задача 4: Определение оптимальных конфигураций запуска анализатора корректности Intel® Parallel Inspector**

Задача заключается в анализе различных возможных режимов работы новейшего продукта Intel® Parallel Inspector с целью нахождения параметров запуска, оптимальных по качеству результатов и затраченному времени на анализ пользовательского приложения. Не секрет, что в случае глубокого анализа программ (особенно многопоточковых) на наличие ошибок работы с памятью или с потоками накладные расходы на анализ могут многократно превышать время работы самой программы.

Однако, если есть возможность контролировать коллекцию результатов (например за счет менее детального анализа каких-то параметров или отключения какого-то вида анализа вообще), появляется возможность контролировать накладные расходы с целью увеличения производительности анализа.

Планируемый результат:

Набор рекомендаций (базирующихся на прогоне популярных коммерческих программ с различными параметрами сбора данных) или предлагаемых «уровней анализа», сочетающих различные компромиссы между «глубиной» анализа и временем работы анализатора.

Требуемый уровень подготовки: Общие представления о параллельном программировании, средствах отладки и проверке параллельных программ, желательно знание Intel® Thread Checker, Intel® Thread Profiler, Numega\* Bounds Checker\*.

*\*\* Торговые марки трех фирм являются собственностью их владельцев.*

### **Задача 5: Параллельные технологии и инструменты в действии**

Проект, предлагаемый отделом инструментов анализа, профилировки и параллелизации, предполагает, что Вы

- имеете компьютерную программу–приложение, которое Вам интересно как специалисту и которую Вы либо разработали/сопровождаете сами, либо взяли из Open Source;
- приложение написано на одном из языков: C, C++, Fortran;
- приложение потребляет нетривиальное количество компьютерных ресурсов;
- Вы заинтересованы в ускорении данного приложения, например с целью быстрее просчитывать модель или чтобы реализовать дополнительные возможности;
- у Вас есть идеи по распараллеливанию кода приложения (а может быть, и начальный вариант параллельного кода).

Участвуя в нашем проекте, Вы сможете значительно ускорить свое приложение посредством параллелизации!

Наши инженеры помогут Вам освоить новейшие технологии параллельного программирования, методы профилирования и анализа программ, принципы оптимизации.

Для участия в данном проекте требуется совсем немного:

- желание;
- наличие приложения, над ускорением которого Вы бы хотели поработать.

*Напишите нам о себе, о Вашем опыте программирования и о приложении.*

## **Задача 6: Разработка физической модели разбиения твердого тела для игрового движка**

Для продвинутых геймеров, любящих смаковать детали игры (и особенно для гейм-дизайнеров, любящих ввернуть в игру что-нибудь этакое), далеко не последним является вопрос всевозможных украшательств игры, не являющихся частями самого геймплея (т. е. основной нити игры), например: красивые клубы дыма, идущие от самоходной установки, в которую попал снаряд, либо отражение луча прожектора от мокрых стен, либо «живая» вода, которая дает волны от ног залезшего в нее героя. Разумеется, большинство этих эффектных украшательств является ничем иным, как искусным инженерным обманом – гейм-дизайнер на самом деле комбинирует несколько текстур, возможно меняющихся в реальном времени, а глазу игрока кажется, что это на самом деле дым.

Предметом особых спекуляций в игровой и околоигровой средах являются более-менее «честные» эффекты, построенные на базе «физического» моделирования. В данном случае для изготовления очередного игрового украшательства используется физическая модель определенной степени грубости, которая может быть применена, исходя из текущих ограничений и условий игры. Украшательств таких немало, например разваливающиеся домики, дробящиеся кирпичи, рвущаяся одежда.

### Основная часть.

Остановимся на одном из игровых моментов – хулиганском желании разбить окошко в домике. И не только окошко – дверь и т. д. Еще лет 5 назад никакой надежды на физическую подоплеку такого явления не было – максимум, чем могли порадовать производители, была возможность получить заранее заданное разбиение объекта. В час «X» объект просто распадался по заранее проведенным линиям.

Задачу в данном случае можно сформулировать следующим образом: Существует двумерная поверхность некоторой формы – для начала можно остановиться на прямоугольнике. К некоторой точке поверхности прикладывается импульс. Необходимо промоделировать процесс физического разбиения поверхности, учитывая следующие требования:

1. Разбиение должно быть визуальным корректным: игрок, видящий эти осколки, не должен сомневаться, что из них нельзя составить первоначальный объект. Хорошим примером может быть падение чего-либо тяжелого на стекло, лежащее на полу, – в этом случае визуальная корректность разбиения должна проследиваться очень четко.
2. В зависимости от изменения физических свойств объекта должна меняться форма осколков и схема распространения трещин в материале. Пример: стекло и оргстекло. В первом случае удар порождает множество осколков (несколько круговых и множество радиальных трещин), во втором случае – только несколько трещин.
3. Необходимо учитывать физические теории, объясняющие форму и распространение трещин в материале.
4. Желательно сразу продумывать аспекты технической реализации, от которых зависит возможность интеграции алгоритма в игру:
  - а) сколько надо памяти алгоритму для работы;
  - б) сколько необходимо времени для работы алгоритма и на каком количестве процессоров;
  - в) можно ли распараллелить алгоритм;
  - г) можно ли применить команды векторной обработки данных (SIMD) для увеличения производительности алгоритма.

## **Задача 7: Разработка алгоритмов и блоков обработки сигналов для современных беспроводных систем связи**

В рамках данной работы предполагается знакомство студентов с основными алгоритмами обработки сигналов, используемыми на физическом уровне современных беспроводных OFDM и OFDMA-систем связи, таких как Wi-Fi, WiMAX или DVB-T.

В процессе работы над проектом будет выполнено построение программной модели изучаемой системы связи (симулятора).

После разработки общего программного симулятора каждый студент получит индивидуальное задание по созданию одного из блоков обработки сигналов, используемого в OFDM-системах связи, на языке описания аппаратуры Verilog.

## **Задача 8: Разработка примера использования библиотеки Intel® Threading Building Blocks**

Предлагается разработать пример кросс-платформенной программы для многоядерных систем, демонстрирующей совместное использование Intel® Threading Building Blocks и операций ввода-вывода для файлов и/или сети для достижения масштабируемой производительности. Язык разработки: C++.

Для примера Вы можете предложить свою задачу или развить следующую: программа может функционировать как веб-сервер, предоставляя по запросу клиентов определенные данные, требующие вычислений (и хранения) на стороне сервера.

Для участия в проекте необходимо:

- изучить возможности TBB, используя документацию, примеры и исходные тексты, доступные на сайте [www.threadingbuildingblocks.org](http://www.threadingbuildingblocks.org);
- изучить возможности кросс-платформенной библиотеки организации ввода-вывода, например входящей в состав Boost ([www.boost.org](http://www.boost.org)), или другой по Вашему выбору;
- предложить концепцию организации программы в виде текста описания, диаграмм, псевдокода или исходного кода.

Участвуя в нашем проекте, Вы сможете получить навыки проектирования современных приложений для многоядерных систем, разобраться в особенностях организации вычислений для максимального использования аппаратных ресурсов, попрактиковаться в разработке масштабируемых многопоточных программ.

## **Задача 9: Алгоритмы для оценки лучшего качества и выбора разбиения/типа макроблока/типа компенсации движения**

Необходимо провести исследовательскую работу и придумать алгоритмы для оценки лучшего качества и выбора разбиения/типа макроблока/типа компенсации движения.

Исследование включает в себя работу с программой для сжатия H.264 видео. Результаты оцениваются с помощью тестового сжатия эталонных видеопоследовательностей и сравнения конечного качества сжатой последовательности до и после «улучшений».

Требуемый уровень подготовки: Знание языка C/C++, принципов отладки и построения программного обеспечения, основ видеосжатия.

## **Задача 10: Проблемы растеризации сцен**

Художники любят оперировать полигонами, в общем случае невыпуклыми. Современное графическое железо работает исключительно с треугольниками. Следовательно, при растеризации некоторых сцен нам потребуется

### **Задача 10.1: Триангуляция невыпуклых многоугольников.**

В 3д-сценах присутствуют как видимые геометрические объекты, так и невидимые вспомогательные объекты, связанные с видимыми. При многократном перекраивании сцены могут появляться объекты-сироты, ни с чем не связанные и невидимые. В конечном продукте наличие таких моделей недопустимо, следовательно, возникает

### **Задача 10.2: Выбрасывание «мертвых» объектов и ссылок.**

Часто художник использует множество файлов текстур, которые разбросаны по файловой системе его рабочего компьютера. Наличие в конечной сцене как абсолютных путей, так и длинных относительных путей недопустимо. Следовательно, нам необходимо

### **Задача 10.3: Перекладывание внешних ресурсов (картинок).**

Для анимации 3д-моделей часто используется анимация по ключевым кадрам с заданной интерполяцией между ними. Часто при сохранении сцены такие анимации сэмпляются и значения анимируемого параметра сохраняются для каждого кадра. Такие сцены занимают много памяти и требуют больших вычислительных ресурсов во время выполнения в реальном времени. Для уменьшения размеров сцен, увеличения скорости работы и скорости загрузки нам нужно

### **Задача 10.4 (повышенной сложности): Ресэмплинг анимационных кривых.**

*Все задачи предлагается выполнить в виде плагинов к open source продукту COLLADA Refinery.*

## **Задача 11: Технологии беспроводной связи\***

Участники будут решать индивидуальные задачи и участвовать в интеграции компонент в систему, выполнять экспериментально-тестовые работы, получают навыки работы в команде в перспективном проекте, опыт применения различных современных технологий: беспроводные коммуникации, сенсорика, обработка сигналов, обработка видеоинформации и машинное зрение, создание специфических баз данных и знаний, искусственный интеллект и машинное обучение, опыт использования современного инструментария и технологий разработки программ.

Цель проекта – создание сервиса, позволяющего с помощью мобильного беспроводного устройства оперативно ориентироваться на территории университетского кампуса, получать информацию об объектах и предметах, находящихся на этой территории, а также иметь мобильный доступ к справочным данным по университету и университетским объявлениям.

### Описание системы:

Этот прототип универсальной системы локального информирования предназначен для повышения эффективности взаимодействия с университетской инфраструктурой студентов, сотрудников университета и посетителей. Пользуясь системой, любой владелец сотового телефона, находящийся на территории университетского кампуса, может быстро сориентироваться на открытой территории или внутри зданий с отображением его положения на плане университетского кампуса. Ему предоставляется возможность идентифицировать находящийся вблизи него объект или предмет, а также найти требуемый объект (здание, аудиторию, площадку и т. д.), предмет или службу на территории кампуса.

Система построена по принципу клиент-серверного приложения. Программа-клиент (Java приложение) устанавливается на сотовом телефоне и позволяет пользователю считывать идентификационную информацию об окружающих объектах, специальным образом помеченных с помощью камеры и Bluetooth (Wi-Fi)-устройства телефона. По GPRS каналу идентификаторы объектов передаются на сервер, который возвращает в программу-клиент контекстную информацию об объекте или предмете, извлеченную из базы данных.

### Предлагаемые задачи:

1. Создание системы ориентирования на территории университетского кампуса, как на открытом пространстве, так и внутри зданий.
2. Создание системы оперативного получения информации об объектах кампуса и предметах, расположенных на его территории.
3. Создание системы инвентаризации материальных ценностей в университете и предоставления оперативной информации о них с распределением прав доступа.
4. Создание системы распространения локальных объявлений.
5. Поисковые и справочные средства по университету, включая пространственно-временное ориентирование (где, что, когда было, есть, будет).
6. Создание интерфейса для добавления и структурирования информации об объектах и предметах и самопополнения баз данных в стиле Wikipedia.

### Используемые ключевые технологии:

1. Распределенная система серверов и баз данных, позволяющая агрегировать и обрабатывать информацию.
2. Считывание/распознавание QR и им подобных кодов с помощью мобильных телефонов.
3. Локальное позиционирование на основе Bluetooth (Wi-Fi)-устройств.
4. Передача данных через Bluetooth, Wi-Fi, GPRS, 3G.
5. Лексикографическая обработка информации.

\*\*\* Проект осуществляется совместно с [Лабораторией физических основ беспроводной связи \(WLab\) ННГУ](#), созданной при поддержке компании Intel.