

ГРАФИК РАБОТЫ СЕКЦИИ COMPUTER SCIENCE

10⁰⁰ – 12⁰⁰

1. Гречкин Максим (9^Г). Средства удаленной загрузки (remote boot) операционных систем.
 2. Артюхин Стас, Янонис Павел (8^В). Программа для построения многоугольников.
 3. Татаринов Андрей (МГУ ВМК, 2 курс). Редактор трехмерных поверхностей, использующий технологию NURBS.
 4. Борисов Владимир (МГУ ВМК, 2 курс). Малобюджетный трехмерный лазерный сканер.
 5. Гречаник Сергей (10^Г). Концепция и платформа межмодульного взаимодействия для разработки моделирующих систем.
 6. Переславцев Алексей (9^В). Разработка компилирующего транслятора выражений.
-

12²⁰ – 14⁰⁰

7. Абрамов Дмитрий (7^В). Сетевые технологии.
 8. Бартунов Сергей, Зайцев Виктор (9^Г). Применение современных технологий программирования на примере разработки программы построения графиков функций.
 9. Левкович-Маслюк Федор (10^Г). Частотный анализ разноязычных текстов с использованием AVL-деревьев.
 10. Сальников Всеволод (9^А). Необходимое и достаточное количество моторов объекта, движущегося по заданной траектории в горизонтальной плоскости.
 11. Федоров Владимир (10^Г). Разработка кроссплатформенного сервера моделирования мультиагентных систем.
 12. Куренев Павел (МГТУ, 1 курс). Графическая библиотека для отображения трехмерных сцен на базе Microsoft DirectX.
-

14²⁰ – 17²⁰

13. Зотов Алексей (МГТУ, 1 курс). Разработка аппаратной и программной части цифрового датчика температуры.
14. Буданов В.М. (НИИ Механики МГУ). Задачи микропроцессорного управления в робототехнических системах.
15. Хохлова Марина (9^Г). Компьютерный скальпель.
16. Руэда Валентин, Чикунов Егор (9^Г). Программа – сетевой чат «VISQ».
17. Чашкин Виктор (9^Г). Моделирование игры «Жизнь».
18. Рахманов Михаил (10^Г). Разработка поисковой машины Search Server.
19. Кожанов А., Пахомов В.Б. (НИИ Механики МГУ). Одноколесный автономный мобильный робот.
20. Теплов Сергей (10^Г). Программа моделирования Солнечной системы.
21. Девятков Ростислав (10^А). Реализация индексаторов на языке Borland C++ 3.1.
22. Сугробов Виталий (10^В). Реализация звукового редактора.

ТЕЗИСЫ ДОКЛАДОВ СЕКЦИИ COMPUTER SCIENCE

10⁰⁰ – 12⁰⁰

1. Гречкин Максим (9^Г). Средства удаленной загрузки (remote boot) операционных систем.

Удаленная загрузка – процесс загрузки рабочей станции, используя информацию с сервера. Это позволяет не администрировать каждый компьютер в отдельности, а настраивать их параметры на сервере. Такая организация сети значительно снижает затраты на обслуживание сети и повышает стабильность ее работы. Если что-то не работает, то стоит совершить несколько простых действий и рабочая станция придет в начальное состояние (в идеальном варианте это просто перезагрузка системы). Для удаленной загрузки было разработано несколько протоколов – RPL (Novell), NetBoot и PXE (Intel). Стандарт PXE (Preboot Execution Environment) определяет набор сервисов, которые могут быть использованы в процессе разработки сетевого загрузчика. Набор этих сервисов предельно прост – доступ к ответу DHCP, передача файлов по протоколу TFTP, Доступ к протоколам IP и UDP. Впервые наиболее обще проблему удаленной загрузки решили в университете Женевы, создав утилиту bpbatch. Заслугой создателей bpbatch было осознание очень важного факта – удаленная загрузка дисковых станций более интересна, чем бездисковых. Но большинство фирм тратят годы усилий для создания системы бездисковой загрузки, считая, что это экономит деньги на «ненужных» жестких дисках. Такие методы имеют один огромный недостаток – они генерируют огромный сетевой трафик, для приемлемой работы сети требуется установка гигабитных сетевых карточек и хабов а их цена перекрывает экономию на жестких дисках, где клиенты могли бы кэшировать данные. Вопрос о кэширование данных в оперативной памяти не встает – ее просто не хватает.

В процессе разработки системы удаленной загрузки были изучены все описанные выше варианты (в том числе создан вариант, основанный на bpbatch, планируется создания варианта на основе Venturcom VXP – системы бездисковой загрузки).

2. Артюхин Стас, Янонис Павел (8^В). Программа для построения многоугольников.

Важный класс геометрических задач составляют задачи на построение. Одна из них рассмотрена в этой работе – задача на построение пятиугольника по серединам его сторон. Способ заключается в вычислении вспомогательной точки с помощью нахождения четвертой вершины параллелограмма. Далее треугольник достраивается по серединам его сторон, после чего образуются две стороны и три вершины пятиугольника. Далее достраиваются стороны пятиугольника по имеющимся точкам. Данный способ представлен в виде компьютерной программы. В ходе построения пользователь выбирает пять точек (середин сторон), по которым строится пятиугольник. Программа имеет приятный интерфейс.

3. Татаринев Андрей (МГУ ВМК, 2 курс). Редактор трехмерных поверхностей, использующий технологию NURBS.

В настоящее время существует большой спектр задач, в которых необходимо точное моделирование сложных гладких поверхностей, требующее от дизайнера минимальных трудозатрат. Это такие задачи, как дизайн автомобилей, самолетов, кораблей, создание персонажей и ландшафтов для компьютерных мультфильмов, и т. д.

NURBS (неравномерные рациональные B-сплайны) – математический аппарат, позволяющий путем задания небольшого количества контрольных точек построить сложную гладкую кривую или поверхность с минимальными трудозатратами – дизайнер может изменять разные участки поверхности, всего лишь меняя позиции контрольных точек.

Редактор N. E. X. позволяет создавать, редактировать и просматривать кривые и поверхности NURBS, а также задавать им свойства материала и визуализировать их методом трассировки лучей. В редакторе предусмотрены возможности настройки параметров отображения, различные инструменты просмотра и редактирования поверхностей, а также инструменты по созданию поверхностей по заданным кривым и операции по работе с несколькими поверхностями.

Трассировщик лучей может работать в двух режимах – в программном режиме и в режиме аппаратного ускорения. Для визуализации NURBS-поверхностей применяется алгоритм конвертирования поверхностей NURBS в поверхности Безье и метод рекурсивного разбиения поверхностей Безье. Для аппаратного ускорения трассировки лучей используются библиотека OpenGL и шейдеры второй версии.

4. Борисов Владимир (МГУ ВМК, 2 курс). Малобюджетный трехмерный лазерный сканер.

(...В течение длительного наблюдения за изменением формы тени, отбрасываемой прямоугольным объектом на рельеф окружающей местности, тонущий в звонком стрекотании мириад кузнечиков, шелесте сухой травы в бескрайнем золотистом поле, растворяющимся в желтоватой дымке, под лазурным небом, с высоко висящем в нем ослепительным, палящим летним солнцем...)

В настоящее время трехмерное сканирование объектов имеет большое значение для профессиональных дизайнеров и для любого человека просто интересующегося трехмерной графикой. Однако существующие сканеры доступные для приобретения, обладая рядом несомненных достоинств, таких как точность сканирования, имеют несколько принципиальных недостатков: существенно ограниченный размер объекта сканирования и самое важное, высокую стоимость порядка 10000 долларов.

Целью данной работы было создание трехмерного сканера, позволяющего проводить сканирование объектов неограниченного размера и главное – имеющего низкую стоимость используемых компонентов и написание необходимого программного обеспечения для обработки полученной от сканера информации.

В итоге работы, созданная рабочая модель системы, позволяет быстро и просто сканировать объекты практически неограниченных размеров (размер объекта сканирования определяется лишь расстоянием до него и мощностью используемого лазера), обладает малой себестоимостью (порядка 50-100 долларов – стоимость простой платы видеозахвата) и крайне простой конструкцией. Тем самым подтверждающая возможность реализации предложенной идеи. Разработанное программное обеспечение позволяет интерпретировать данные, полученные со сканирующей головки, визуализировать полученную модель с произвольной точки, и сохранять эту модель в формате внутреннего представления программы, для ее последующего использования в специальных программах, разработанных для этой системы.

Возможные направления продолжения работы: увеличение точности работы сканера путем использования автоматической системы перемещения сканирующей головки; использование системы зеркал для устранения артефактов возникающих при работе системы во время сканирования резких перемен глубин объекта; разработка более высококачественного программного обеспечения обработки данных, включающей в себя конверторы полученных моделей в распространенные форматы их представления.

5. Гречаник Сергей (10^Г). Концепция и платформа межмодульного взаимодействия для разработки моделирующих систем.

Цель проекта – создание системы с модульной архитектурой и модулей для этой системы, в том числе физического. Основная задача проекта – это обеспечение возможности создания модулей, а также переноса несовместимых с системой модулей под систему при минимальных затратах ресурсов программиста.

Система состоит из ядра, загрузчика и загружаемых модулей. Ядро системы предоставляет модулям основные функции системы: вывод в консоль, информацию о типах, возможность экспортировать и вызывать функции по имени. Ядро состоит из менеджера модулей, менеджера информации о типах, менеджера команд, основных потоков вывода.

Модуль обычно является динамически загружаемой библиотекой и предоставляет загрузчику специфический для загрузчика интерфейс. Загрузчик выполняет загрузку модулей в систему и выгрузку из неё.

Модули системы могут взаимодействовать через компоненты ядра и напрямую. В менеджере модулей регистрируются все загружаемые модули. Менеджер модулей позволяет получить указатель на другие части ядра и другие модули. В менеджер информации о типах каждый модуль может экспортировать свои типы данных для использования их другими модулями. Информация о типах содержит информацию об отношениях между типами, способе создания объекта и т. д. В менеджер команд каждый модуль может экспортировать любой функтор с его именем. Этот функтор может потом быть вызван при помощи ввода команды в консоли или каким-либо другим способом.

Функцией физического модуля является обработка взаимодействий между объектами среды. В данной версии программы объектами среды являются атомы, взаимодействующие друг с другом по закону потенциала Леннарда–Джонса. Расчёт поведения объектов выполняется численными методами. Был проведён анализ некоторых разностных схем на выполняемость закона сохранения энергии.

В процессе создания системы были применены следующие шаблоны проектирования: Команда, Функтор, Итератор, Стратегия, Посетитель, Фабричный метод, Адаптер.

6. Переславцев Алексей (9^В). Разработка компилирующего транслятора выражений.

Целью работы являлась программа построения произвольного графика функции. Используемый в ней алгоритм отличается тем, что в нем проходит предварительная компиляция введенного пользователем выражения в функцию в понимании языка Си (т. е. непосредственно в машинный код), а лишь затем идет вычисление Y-координаты точки с помощью этой функции. Это существенно увеличивает скорость работы программы, и поэтому без торможения можно работать сразу с несколькими графиками функций.

В ходе проекта пришлось выполнить следующие задачи: 1) Создание грамматики, способной описать арифметическое выражение. 2) Перевод обычной записи выражения в обратную польскую. 3) Компиляция обратной польской записи в машинный код. 4) Разработка интерфейса под операционную систему Windows.

Из перечисленных вариантов наиболее сложным оказался третий. Для его реализации было использован ДОС-компилятор ассемблера (Turbo Assembler) с опцией создания файла листинга, в котором каждой команде ассемблера соответствует код машинной команды в шестнадцатеричном виде. Для отладки самостоятельно скомпилированного кода использовался отладчик среды Microsoft Visual Studio.

12²⁰ – 14⁰⁰

7. Абрамов Дмитрий (7^Б). Сетевые технологии.

Сделан обзор сетевых технологий, их история и перспективы развития. Рассмотрены локальные и глобальные сети, особенности хранения данных.

8. Бартунов Сергей, Зайцев Виктор (9^Г). Применение современных технологий программирования на примере разработки программы построения графиков функций.

Цель работы – создание среды для исследования математических функций с возможностью разбора выражения, характеризующего функцию, и исследования ее на различных участках.

В проекте использованы современные технологии сохранения и восстановления произвольных объектов, которые могут применяться в любых других задачах и расширяемый формат данных, основанный на XML, что позволяет не терять информацию о ранее исследованных функциях и реализовать многоязыковой интерфейс (совместимый с Unicode). Данный формат данных хранит тип сохраненного объекта, его значение, преобразованное в строку, а также аналогичные сведения обо всех его свойствах, что и представляется в виде файла XML.

Проект реализован под платформу .NET, его компоненты могут быть использованы в других

разработках на практически любых языках программирования, совместимых с данной технологией. Программа способна работать на многих современных платформах (включая мобильные).

Среда способна отображать график в разном масштабе, передвигать центр координат и устанавливать точность получения значений функции при различных аргументах. Графическая часть проекта использует возможности GDI+ MS Windows. Также программа способна исследовать функцию на четность/нечетность и возрастание/убывание на различных промежутках.

9. Левкович-Маслюк Федор (10^Г). Частотный анализ разноязычных текстов с использованием AVL-деревьев.

Целью работы было определение наиболее употребительных слов в текстах на различных языках. Анализировались несколько крупных текстов на русском и английском языках. Слова, составляющие текст, записывались в узлы дерева, и вычислялось распределение слов по частотам их вхождения в текст. Эффективность программ обеспечивалась за счет использования сбалансированного двоичного дерева (AVL-дерева).

AVL деревья были введены Адельсоном-Вельским и Ландисом в 1962 году. Оно характеризуется тем, что высоты левого и правого поддеревьев любого узла отличаются не более, чем на 1. Это свойство достигается путем локальных (затрагивающих 2 или 3 узла) модификаций дерева при каждой вставке нового элемента. В результате обеспечивается примерно одинаковое (близкое к логарифму от числа узлов) время доступа к данным, записанным в такое дерево.

В рассматриваемой задаче использование сбалансированного AVL-дерева позволило быстро вычислять частоты вхождения слов в текстовые массивы. Полученные данные сравнивались для текстов на различных языках. В работе представлены результаты сравнения частот для нескольких классов текстов.

10. Сальников Всеволод (9^А). Необходимое и достаточное количество моторов объекта, движущегося по заданной траектории в горизонтальной плоскости.

Исследуется задача нахождения минимального необходимого количества моторов для обеспечения движения по заданной траектории и пути её решения, примененные на «Мобильном Роботе», сделанном в качестве опытного образца на базе программируемого блока RCX 1.0. Таким образом, для решения этой задачи необходимо собрать опытный образец из стандартных деталей фирмы Legotm и запрограммировать движение. Не составляет никакого труда организовать движение, используя два мотора, но можно ли обеспечить движение по траектории, заданной, например, стенками, которые необходимо объезжать, используя меньшее количество моторов? Если не устанавливать на объект ни одного мотора, то он становится неуправляемым, следовательно, поставленный вопрос сводится к проверке возможности движения объекта с одним мотором, и написанию программы для управления этим устройством. Если удастся запрограммировать движение, использующее лишь один мотор, то мы получим возможность использовать в качестве основного программируемого устройства плату лишь с одним выходом, тем самым заметно упростив ее. В то же время необходимо подавать электрический ток лишь на один мотор, что обеспечивает меньшее количество деталей, которые могут изменять характер своего движения, что дает возможность быстрее определять неисправности в работе механизма. При получении ответа на поставленный вопрос был построен опытный образец, который действительно использует лишь один мотор и три световых датчика для свободного передвижения в горизонтальной плоскости и объезда препятствий.

При создании опытного образца выявилась проблема отсутствия определенного типа деталей, необходимых для осуществления проекта, которых не существует в Legotm, поэтому пришлось сделать эти детали, используя подручные средства. Из-за отсутствия обратной связи мотора с программируемым блоком возникают трудности при написании программы, управляющей движением, так как не предоставляется возможности определить угол, на который повернута ось.

Среди программного обеспечения, которое программирует блок RCX, наиболее удобным на данный момент считается Not Quite C (NQC, D. Baum, J. Hansen). Именно эта среда и была использована при решении данной задачи.

11. Федоров Владимир (10^Г). Разработка кроссплатформенного сервера моделирования мультиагентных систем.

Цель работы – реализация сервера моделирования мультиагентных систем. Сервер моделирования мультиагентных систем – это программный комплекс, включающий в себя модули обработки сетевых подключений, программ поведения агентов, обработки взаимодействий объектов моделируемой системы. Он представляет собой совокупность средств для задания модели, самого процесса моделирования и средств администрирования.

Особый интерес представляют системы предполагающие наличие действующих программных единиц – агентов и объектов, управляемых пользователем – клиентами. Их использование позволяет решать задачи моделирования, предполагающие вариативность действий объекта.

В ходе разработки комплекса были поставлены следующие задачи о разработке: 1) Системы обработки клиентов и агентов, и их взаимодействий между собой и с объектами моделируемой системы; 2) Средств описания объектов (их реакций и возможностей их действий) и поведения агентов моделируемой системы; 3) Средств администрирования моделируемой системы, изменение различных условий протекания процесса; 4) Клиентского приложения, работающего с сервером посредством сетевого соединения; 5) Средств визуализации в серверном и клиентском приложениях.

В докладе рассматриваются различные варианты решения данных задач на примере игровых серверов.

В ходе разработки был создан программный комплекс сервера моделирования мультиагентных систем, использующий: 1) Многопоточность современных ОС; 2) Универсальное кроссплатформенный сетевой API BSD Sockets (WinSock1.1); 3) Язык надпрограммных настроек и OpenSource-библиотека его обработки Lua (v. 5); 4) OpenSource-библиотеку OGRE (Object-Oriented 3D Graphical Engine) для реализации визуализации моделируемой системы в клиентском приложении.

В результате разработки были достигнуты следующие результаты: 1) Создана программная часть, реализующая сетевое взаимодействие для серверного и клиентских приложений; 2) Создана консоль администрирования сервера; 3) Создан тестовый вариант клиентского приложения.

В дальнейшем на базе полученного программного комплекса будет вестись разработка многопользовательской ролевой игры, и решаться ряд задач моделирования различных процессов.

12. Куренев Павел (МГТУ, 1 курс). Графическая библиотека для отображения трехмерных сцен на базе Microsoft DirectX.

Основной целью проекта является создание библиотеки с набором классов для упрощения создания приложений, использующих Direct3D для отображения трёхмерной графики.

Сначала была создана программа 3DTest, в которой велась разработка и отладка графических функций, а затем весь код был вынесен в библиотеку.

Библиотека написана на Microsoft Visual C++ (поддерживаются версии 6.0 и 7.1). Проект состоит из самой библиотеки (GraphLib.dll) и заголовочного файла (GraphLib.h), в котором находятся описания функций, типов данных, структур и прототипы классов. Библиотека работает следующим образом: в программе создаётся указатель на объект класса, далее, с помощью соответствующей функции библиотеки, создаётся новый объект этого класса и запоминается указатель на него. По завершению работы с объектом он удаляется функцией, также экспортируемой библиотекой.

Библиотека имеет средства для работы Direct3D: класс с базовыми функциями, классы для работы с текстурами, с вершинными буферами и др. Также содержит классы для упрощения создания приложений: классы для редактирования изображений, для работы с динамическими массивами, для работы с файлами и др. А также классы, которые решают более сложные задачи: загрузка и сохранение трёхмерных сцен (карт) и отображение их на экране.

Данная работа может служить основой для трёхмерного отображения моделируемых объектов (в физике, например), для создания игры и др.

14²⁰ – 17²⁰

13. Зотов Алексей (МГТУ, 1 курс). Разработка аппаратной и программной части цифрового датчика температуры.

Основной целью проекта является подключение к ПК дополнительных устройств. Одной из задач было присоединение к ПК измерительных приборов. В моём случае был создан программно-аппаратный комплекс для измерения температуры. Комплекс состоит из двух основных компонентов: 1) Аппаратная часть на базе АЦП (аналого-цифрового преобразователя) и компьютера; 2) Программная часть, позволяющая обрабатывать полученные данные.

Аппаратная часть представляет собой устройство, подключаемое к порту LPT компьютера (принтерному порту). Реализация базируется на микросхеме 10-битного АЦП с последовательным интерфейсом. Устройство позволяет считывать значение температуры 100 раз в секунду.

Программная часть написана на Visual C++. Программа занимается представлением данных в удобном виде для восприятия и обработки данных. Предусмотрено сохранение данных в файл, совместимый по формату с Microsoft Excel.

Данная работа может служить основой для компьютеризации лабораторных работ по физике.

14. Буданов В.М. (НИИ Механики МГУ). Задачи микропроцессорного управления в робототехнических системах.

Рассматриваются задачи микропроцессорного управления в системах управления мобильными роботами. Как правило, такие задачи распадаются на три класса: работа с различными датчиками (фотодатчики, сенсорные датчики, видеокамеры и т.д.), взаимодействие с внешними устройствами робота (двигатели, электромагниты, пневматика) и собственно задача управления, т.е. решение поставленной перед роботом задачи. Для работы с разными классами задач используются различные аппаратные платформы: для работы с датчиками и внешними устройствами – преимущественно микроконтроллеры PIC, AVR, Motorola; для задач управления – платформы Intel 8x86 (часто в виде ноутбуков), RISC процессоры Texas Instruments и другие. В докладе будут рассмотрены применения микроконтроллеров Motorola для решения задач управления шестиногой шагающей машиной и другими робототехническими устройствами. Эти микроконтроллеры программируются как на ассемблере, так и на Си, имеют широкий спектр портов ввода и вывода, что делает их удобной логической платформой для робототехники. Также будут рассмотрены примеры применения других микроконтроллерных средств.

Доклад адресован в первую очередь тем, кто хочет начать осваивать самостоятельное программирование микроконтроллерных устройств и принять участие в разработке программного обеспечения для робототехнических систем в Институте Механики МГУ.

15. Хохлова Марина (9^Г). Компьютерный скальпель.

Задача данного проекта состоит в том, чтобы построить сечение прямой для некоторой составной фигуры, расположенной на плоскости. Данная фигура должна состоять из участков разных цветов, которая может задаваться пользователем с помощью формулы, или являться набором кружков разных диаметров и цветов, которые могут ставиться на плоскость пользователем с помощью мышки. Пользователь может пересечь данную фигуру прямой, поставив две точки, которые соединятся линией. Сечение будет представлять собой набор отрезков прямой различной длины и цветов.

Возможные приложения данной задачи связаны с визуализацией характерного расположения атомов и молекул при компьютерном моделировании молекулярных систем, а также с изучением вида различных сечений живых организмов. Возможны приложения при изучении курса геометрии.

16. Рузда Валентин, Чикунев Егор (9^Г). Программа – сетевой чат «VISQ».

Поставленной целью было создание чата для общения разных пользователей с различных компьютеров сети ROOM2X лица

Задачи, которые встали в процессе создания программы: эффективная работа с файлами, простой, красивый интерфейс, удобное администрирование. Работа с файлами – одна из самых серьезных проблем является работа с файлами, так как нельзя открывать файл с двух компьютеров одновременно. Чтобы пользователю было удобнее общаться с другими людьми, было необходимо сделать простой интуитивно понятный простому пользователю интерфейс. Удобное администрирование позволяют модерировать чат, причем делать это могут не только авторы проекта, но и пользователи, которым были даны такие права. Из этого следует, что в программе присутствует зародышевый вариант иерархической системы чата.

Программа написана под операционную систему MS-DOS. Благодаря невысоким требованиям к компьютеру, она будет работать даже на самых слабых машинах.

Планы на будущее: поддержка нескольких интерфейсов, звуковое сопровождение, передача сообщений через протокол TCP/IP, шифрование пакетов.

17. Чашкин Виктор (9^Г). Моделирование игры «Жизнь».

Основной целью разработки являлось создание интерактивной графической среды для игры «Жизнь». Эта игра – частный случай клеточных автоматов, стилизованных, синтетических миров, определенных простыми правилами, подобными правилам настольных игр. Она описывает динамику популяции стилизованных организмов во времени под действием противоборствующих процессов размножения и вымирания.

Так как данная программа изначально ориентировалась на пользователя, была очевидной необходимость создания удобного пользовательского интерфейса и специальных средств для быстрого редактирования имеющихся и создания новых конфигураций. Этими средствами являются редактор конфигураций с возможностью размножения фрагмента изображения («штампа») и библиотека базовых конфигураций, которую пользователь может дополнять при помощи редактора. Реализована такая важная возможность, как сохранение и загрузка конфигураций из библиотеки.

Программа изначально писалась под операционную систему MS-DOS. Ведется разработка Windows-версии.

18. Рахманов Михаил (10^Г). Разработка поисковой машины Search Server.

Программа Search Server – это поисковая программа, которая работает наподобие современных поисковых серверов и имеет свою базу данных. Search Server принимает в качестве параметра слово или набор слов. Программа проверяет свою базу данных на наличие этих слов и выдает идентификационные номера тех файлов, в которых содержатся все слова из заданного набора (каждому файлу в базе данных присвоен свой идентификационный номер).

При большом объеме базы данных, примерно 10^{10} документов, количество документов, где содержится определенное слово или набор слов, обычно превосходит 10^8 документов. Следовательно, список идентификационных номеров занимает примерно 400 Мб, т. к. каждый идентификационный номер занимает 4 байта (unsigned long). Очевидно, при достаточно большом количестве запросов может произойти переполнение жесткого диска. Для этого был использован алгоритм сжатия получающихся списков. Благодаря этому алгоритму размер файла уменьшается в полтора-два раза.

Пути дальнейшего развития: написать подобие поискового Интернет сервера в локальной сети и создать GUI для данной программы.

19. Кожанов А., Пахомов В.Б. (НИИ Механики МГУ). Одноколесный автономный мобильный робот.

Будущее робототехники в современном мире состоит в создании полностью автономных, самостоятельно ориентирующихся, двигающихся, способных принимать решения мобильных роботов. Несмотря на кажущееся сходство с радиоуправляемыми моделями эти роботы представляют собой принципиально новые устройства. Они должны быть наделены органами восприятия окружающей среды, манипуляторами, а также быть очень маневренными. Наиболее экономичны и удобны в разработке колесные роботы. В то же время, нетрудно заметить, что маневренность улучшается с уменьшением числа колес: ясно, что двухколесный велосипед проедет там, где не проедет трехколесный и т.п. Поэтому наиболее маневренным является робот, состоящий всего лишь из одного колеса.

20. Теплов Сергей (10^Г). Программа моделирования Солнечной системы.

Целью работы является реализация модели солнечной системы с дружественным пользовательским интерфейсом для возможного моделирования и анализа реальных физических ситуаций, происходящих в солнечной системе, а также в любой ситуации, связанной с движением планет, которую может задать пользователь. В процессе работы над проектом были поставлены следующие задачи: Создание приложения принимающего у пользователя с клавиатуры и позволяющее ему создавать объекты с заданными пользователем параметрами (реальными и применимые в реальных физических задачах). Приложение также должно предоставлять пользователю возмож-

ность наблюдать изменения модели, а также вносить изменения в процессе работы программы. Для расчетов связанных с силой возникающей между объектами силами притяжения должен использоваться физический алгоритм связанный с законом всемирного притяжения, все расчеты должны производиться на базе физических вычислений. Приложение также должно обладать дружественным пользовательским интерфейсом для удобства работы.

В данный момент создано ДОС-приложение, использующее библиотеку классов для хранения данных и реализации взаимодействий между объектами модели, библиотеку VGI для визуализации модели, ввод с клавиатуры для задания параметров объектов системы. Пользователь может создавать и удалять объекты в реальном времени, не прерывая работы программы.

В ближайшем будущем планируется создание дружественного интерфейса для приложения, включающего в себя возможности изменения масштаба при просмотре взаимодействий модели пользователем, редактора, позволяющего отредактировать создаваемые в реальном времени объекты. Также будет добавлена возможность просмотра пользователем численных значений физических величин, изменяющихся во время действия модели. В дальнейшем будет осуществлен перенос модели в ОС Windows, внедрение новой графической библиотеки, а также более точный расчет физических величин.

21. Девятов Ростислав (10^A). Реализация индексов на языке Borland C++ 3.1.

Индексы – "мнимые" массивы, которые на самом деле представлены с помощью функций *get* и *set*, хотя пользователь обращается к ним посредством обычного синтаксиса для массивов. Функция *get* принимает набор индексов и возвращает элемент мнимого массива, функция *set* принимает набор индексов и значение элемента массива и должна установить его.

Это позволяет: 1) Предлагать пользователю несколько вариантов доступа к данным, храня в памяти лишь один из них. 2) Хранить несколько представлений данных и автоматически обновлять их все при изменении данных. 3) Предлагать данные пользователю в более удобном виде, чем они хранятся в памяти. 4) Протоколировать изменения данных и обращения к ним.

22. Сугробов Виталий (10^B). Реализация звукового редактора.