Министерство образования и науки Российской Федерации

**Орский гуманитарно-технологический институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования «Оренбургский государственный университет»**

ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ РАЗРАБОТКИ, ВНЕДРЕНИЯ И ЭКСПЛУАТАЦИИ ПРОГРАММНЫХ СРЕДСТВ

Материалы
III Всероссийской научно-практической конференции

Орск 2016

УДК 004.4
ББК 32.973
Т 33

Печатается по решению редакционно-издательского совета Орского гуманитарно-технологического института (филиала) ОГУ

Т33 Теоретические вопросы разработки, внедрения и эксплуатации программных средств : материалы III Всероссийской научно-практической конференции / отв. ред. В. С. Янё. – Орск : Издательство Орского гуманитарно-технологического института (филиала) ОГУ, 2016. – 60 с. – ISBN 978-5-8424-0828-3.

В сборнике представлены материалы Всероссийской научно-практической конференции, посвященной вопросам разработки, внедрения и эксплуатации программных средств.

ISBN 978-5-8424-0828-3

Содержание

АЛГОРИТМ РАСЧЁТА И ПОСТРОЕНИЯ ГЛАДКОЙ НЕПРЕРЫВНОЙ ПЛОСКОЙ СОВОКУПНОСТИ ДУГ ОКРУЖНОСТЕЙ, ЗАДАННОЙ КОНЕЧНОЙ ПОСЛЕДОВАТЕЛЬНОСТЬЮ ОПОРНЫХ ТОЧЕК И КАСАТЕЛЬНЫХ

М. В. Иванюк, А. В. Кузьмин, В. А. Лысов, А. В. Щеголев
АО «Механический завод», г. Орск

В. С. Янё
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

В различных сферах деятельности промышленного предприятия машиностроительного профиля актуальной является задача представления (аппроксимации) гладкими кривыми с заданной точностью криволинейных контуров разного рода и назначения.

В качестве производственных приложений можно привести следующие.

Основное производство

Графоаналитическое моделирование:

- динамики механических свойств металлов [7];
- пооперационной динамики наклёпа [8].

Инструментальное производство

Профилирование осевых сечений инструмента (пуансонов и матриц) в технологических процессах изготовления глубокой вытяжкой осесимметричных цельнотянутых изделий:

- гладкой совокупностью дуг кривых третьего порядка с целью изучения динамики напряжённо-деформированного состояния;
- гладкой совокупностью дуг окружностей с целью изготовления рабочих поверхностей пуансонов и матриц на токарном оборудовании с ЧПУ.

Вспомогательное производство

Аппроксимация контура рабочей поверхности кулаков, заданного последовательностью опорных точек, гладкой совокупностью дуг окружностей с целью изготовления на вертикально-фрезерном оборудовании с ЧПУ.

1. Постановка задачи

Необходимо представить с заданной точностью произвольный плоский криволинейный контур гладкой кривой, образованной дугами окружностей, проходящей через все опорные точки контура по заданным касательным.

Контур может быть выпуклым, вогнутым, выпукло-вогнутым (содержащим точки перегиба), а также замкнутым, отображающим, например, рабочую поверхность кулаков, являющихся частью механизма прессового оборудования.

Для этого необходимо разработать алгоритм расчёта и построения с заданной точностью гладкой непрерывной плоской совокупности дуг окружностей. Причём искомая совокупность определена конечной последовательностью опорных точек и касательных.

Полученный таким образом алгоритм должен удовлетворять требованиям поставленной задачи.

2. Обоснование существования решения

Теоретической основой расчёта является следующее обоснованное утверждение [6].

Утверждение

Пусть $A_1, A_2, \ldots, A_n$ – некоторая последовательность точек, определяющая какой-либо выпукло-вогнутый плоский криволинейный контур, возможно замкнутый. Через точки $A_1, A_2, \ldots, A_n$ проходят регламентирующие касательные $a_1, a_2, \ldots, a_n$ (рис. 1).

Тогда существует гладкая непрерывная плоская кривая, образованная дугами окружностей, последовательно проходящая через точки $A_1, A_2, \ldots, A_n$, касаясь прямых $a_1, a_2, \ldots, a_n$ соответственно.

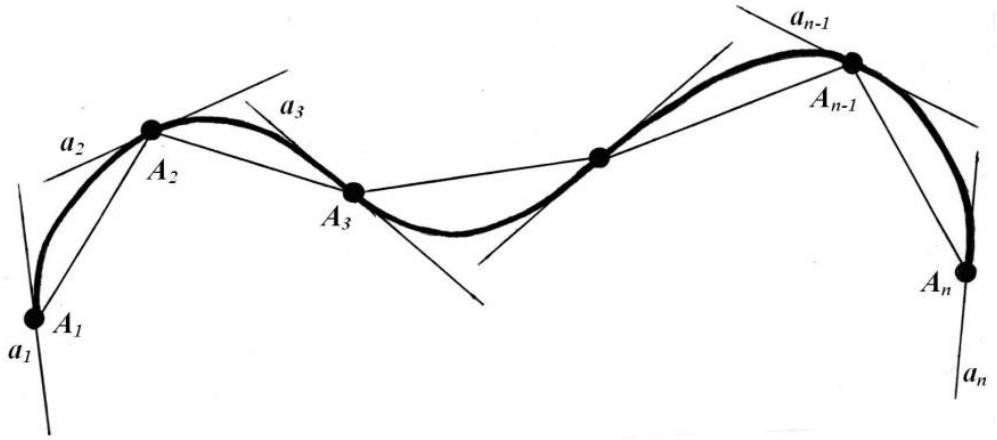

Рис. 1. Схема описания контура опорными точками и регламентирующими касательными

Точки $A_1, A_2, \ldots, A_n$ определены в прямоугольной декартовой или полярной системе координат. Возможны два варианта взаимного расположения точек A_1, A_2, представленные на рисунках 2 и 3.

В том и другом случае треугольник A_1XA_2 определяет равнобедренные треугольники A_1O_1X и A_2O_2X так, что точки O_1, X, O_2 лежат на одной прямой.

Таким образом, показано существование точки X такой, что дуга A_1X окружности с центром в точке O_1 радиуса $O_1A_1 = O_1X$ и дуга A_2X окружности с центром в точке O_2 радиуса $O_2A_2 = O_2X$ в точке X имеют общую касательную, То есть имеет место сопряжение.

Последовательность таковых пар дуг для всех секторов представляет искомую гладкую кривую, образованную дугами окружностей.

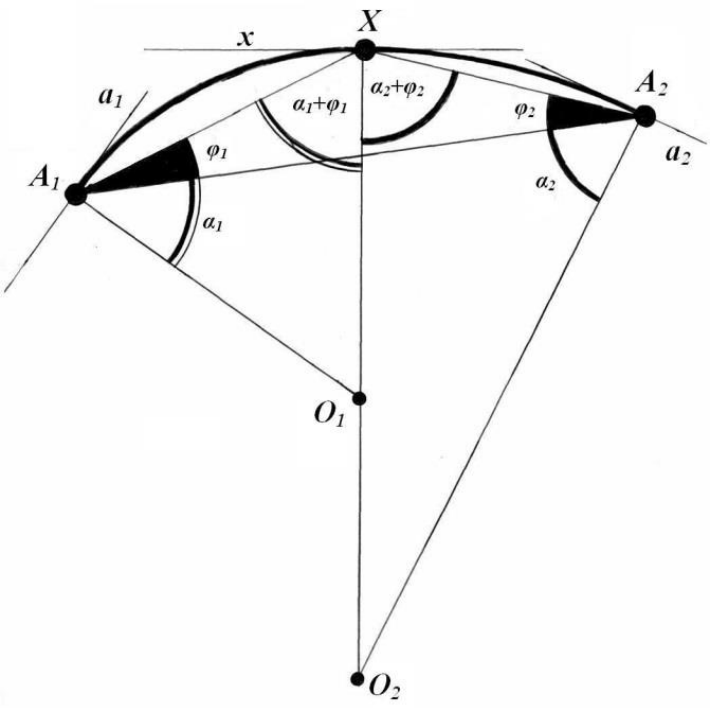

Рис. 2. Схема расположения точки пересечения регламентирующих касательных контура внутри сектора

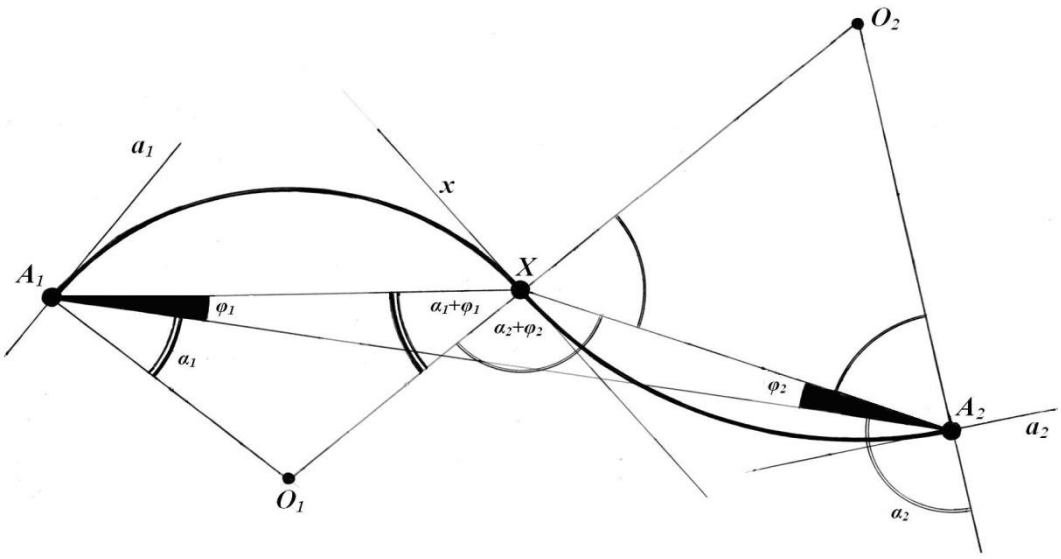

Рис. 3. Схема расположения точки пересечения регламентирующих касательных контура вне сектора

3. Математическая модель расчёта и построения

Рассмотрим один из вариантов математической модели автоматизированного проектирования (расчёта и построения) криволинейного контура, заданного дискретным множеством опорных точек A_1, A_2, ..., A_n (рис. 1). Точки определены в прямоугольной декартовой или полярной системе координат.

С помощью прикладного аппарата линейной алгебры [1], математического анализа [2, 5], математического моделирования производственных процессов [3], аналитической геометрии [4] производится расчёт параметров проектируемого криволинейного контура следующим образом.

Вначале осуществляется проектирование первого фрагмента (сектора). Далее рекуррентно для каждого последующего.

Основу расчёта и построения составляет следующее свойство биссектрис треугольника.

Свойство

В произвольном треугольнике $A_1B_1A_2$ (рис. 4) из вершин A_1, A_2 проведены биссектрисы, пересекающиеся в точке X_1. Построены отрезки: A_1O_1 перпендикулярно A_1B_1, A_2O_2 перпендикулярно A_2B_1.

Из точки X_1 опущен перпендикуляр на основание треугольника A_1A_2. Построена прямая x_1 параллельно A_1A_2.

В этом случае имеет место следующее:

- $A_1O_1 = X_1O_1$, $A_2O_2 = X_1O_2$;
- дуги A_1X_1 и A_2X_1 являются дугами окружностей, сопрягающихся в точке X_1 по прямой x_1.

Построенный таким образом фрагмент $A_1X_1A_2$ является гладким, то есть удовлетворяет условию поставленной выше задачи.

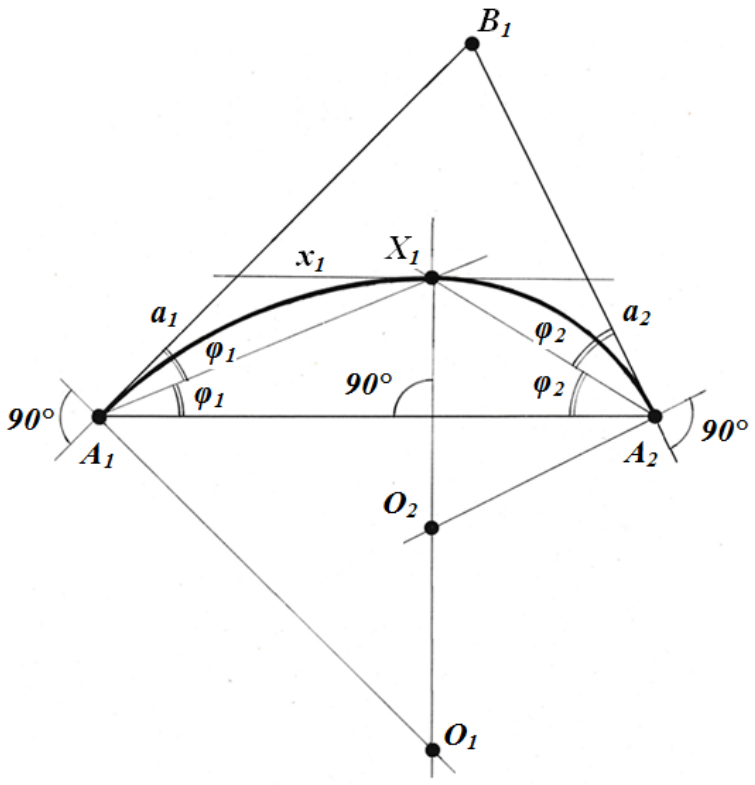

Рис. 4. Геометрическая модель построения гладкой аппроксимирующей кривой, образованной дугами окружностей, на дискретном множестве опорных точек

4. Практические особенности расчёта и построения

Предложенный в работе алгоритм допускает варьирование угловыми коэффициентами регламентирующих касательных в пределах угловых коэффициентов соседних отрезков $A_{i-1}A_i$, A_iA_{i+1} (i = 2, ... , n-1).

За счёт этого обеспечено получение множества аппроксимирующих кривых, заданных конкретной последовательностью опорных точек. Из полученного множества возможен выбор кривой технологически приемлемой.

Таким образом, создана основа для пооперационного проектирования контуров осевых сечений пуансонов и матриц в технологических процессах изготовления осесимметричных цельнотянутых изделий глубокой вытяжкой.

Необходимо отметить следующее.

- для заключительной вытяжной операции положение регламентирующих касательных a_1, a_2, ..., a_n определено контуром осевого сечения изделия согласно чертежу. В этом случае следует только лишь вычислить угловые коэффициенты касательных;
- для каждой предыдущей операции, начиная с первой вытяжной, необходимо определить положение касательных по предложенному алгоритму и вычислить их угловые коэффициенты.

Если регламентирующие касательные a_1, a_n в точках A_1, A_n не заданы, то пользователь вправе определить их положение. Например, для выпуклого фрагмента параллельно оси ординат, для вогнутого фрагмента параллельно оси абсцисс или задать технологически обоснованные значения угловых коэффициентов касательных a_1, a_n.

Если контур имеет точки перегиба, то целесообразно сделать расчёт для каждого выпуклого или вогнутого фрагмента отдельно (рис. 5).

Для повышения точности проектирования рекомендуется ввести промежуточные опорные точки, в частности, если имеет место перегиб, а точка перегиба, как опорная, не задана.

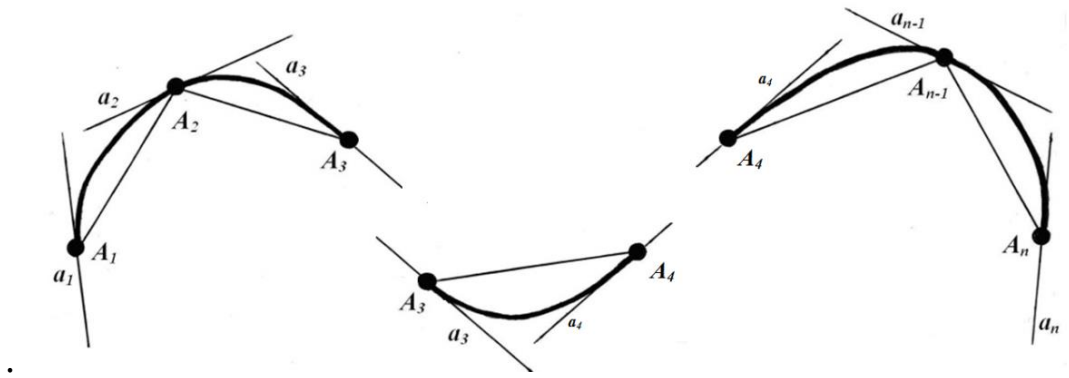

Рис. 5. Схема описания контура опорными точками и касательными по выпуклым и вогнутым фрагментам

5. Методика автоматизированного проектирования

Разработанный теоретический подход к автоматизированной аппроксимации рассмотренных выше криволинейных контуров и его реализация в виде методики проектирования плоских гладких кривых на дискретном множестве опорных точек представлен в виде практической методики в таблице 1.

Таблица 1

Методика автоматизированной аппроксимации плоских криволинейных контуров гладкими непрерывными кривыми

1	**Постановка задачи**
1	2
1.1	**Формулировка** Необходимо представить произвольный плоский криволинейный контур гладкой кривой, образованной дугами окружностей, проходящей через все опорные точки контура по заданным касательным
1.2	**Область применения** Подготовка технологических процессов изготовления осесимметричных изделий, имеющих криволинейные контуры
1.3	**Теоретическое обоснование решения** Существует гладкая непрерывная плоская кривая, образованная дугами окружностей, проходящая через все заданные точки по регламентирующим касательным
1.4	**Виды решения** Для данной задачи существует два вида решений (рисунки 3, 4)
2	**Выбор оптимальных решений**
2.1	**Производственная необходимость поиска оптимальных решений** Критерии оптимальности: 1. Наименьшая длина кривой, определяющей рабочий контур для обеспечения наиболее быстрого перехода между опорными точками. Критерий оптимальности: суммарная длина дуг фрагментов по секторам наименьшая. 2. Наиболее плавный переход между опорными точками с целью снижения износа оборудования. Критерий оптимальности: наименьшая разность по модулю длин радиусов дуг фрагментов по секторам
2.2	**Обоснование существования решения. Выбор характеризующих величин** Существует семейство гладких кривых, образованных дугами окружностей, проходящих через заданные опорные точки по регламентирующим касательным
2.3	**Выбор зависимых и независимых переменных** Определение областей, характера и шага изменения. Величина шага определяется технической потребностью, зависит от класса точности металлорежущего оборудования

Окончание таблицы 1

1	2
3	**Получение и анализ результатов**
3.1	С помощью прикладного аппарата аналитической геометрии и математического анализа методами эмпирического математического моделирования производится расчёт для каждого сектора в декартовой или полярной системе координат (рисунок 5): • координат точек O_1, O_2, X_1, B_1; • длин радиусов O_1A_1, O_2A_2; • длин дуг A_1X_1, A_2X_1; • угловых коэффициентов касательных a_1, a_2, x_1; • значений углов φ_1, φ_2 в градусной и радианной мере. В соответствии с заданным критерием оптимальности выбираются и выводятся в файл определенной структуры результаты совместно с контрольными величинами. Оценивается погрешность вычислений
4	**Алгоритмы и программное обеспечение**
4.1	Основные требования к алгоритмам и программному обеспечению: • приемлемые временные характеристики; • визуализация процесса; • поэтапный расчёт по фрагментам (секторам); • приостановка и возобновление процесса конкретного сектора; • устранения нештатных ситуаций в интерактивном режиме

Рассмотренная методика в течение эксплуатации с 1993 года получила широкое применение в основном, инструментальном и вспомогательном производстве предприятия машиностроительного профиля АО «Механический завод» (г. Орск, Оренбургская область), в технологических процессах изготовления контрольно-мерительного инструмента, рабочих поверхностей кулаков, поверочных шаблонов.

Практические результаты

В результате внедрения методики полностью исключено появление выступов и впадин в точках сопряжения. В процессе эксплуатации получено снижение станкоёмкости изготовления на операциях черновой и чистовой обработки на металлорежущих станках с ЧПУ.

Наиболее значимым является применение методики в инструментальном производстве для технологических процессов изготовления цилиндрических деталей методом глубокой вытяжки.

В процессе эксплуатации показана эффективность разработанной методики, что позволило перейти к разработке алгоритмов автоматизированной аппроксимации и реализации в виде подсистемы САПР ТП.

6. Оптимальные решения

В соответствии с заданным критерием оптимальности выбираются и выводятся в файл определённой структуры результаты совместно с контрольными величинами и оценкой погрешности вычислений.

7. Заключение

Разработан инструментарий, позволяющий дискретные цифровые данные, например, с бумажных носителей преобразовать в аналитический и графический формат непрерывных функциональных зависимостей и гладких аппроксимирующих кривых.

Универсальность разработанного инструментария с возможностями интерактивного доступа позволяет применять его для решения широкого круга производственных задач.

Библиографический список

1. Амосов, А. А. Вычислительные методы для инженеров : учебное пособие / А. А. Амосов, Ю. А. Дубинский, Н. В. Копчёнова. – 2-е изд., доп. – М. : Издательство МЭИ, 2003. – 569 с., ил.

2. Данко, П. Е. Высшая математика в упражнениях и задачах : учеб. пособие для студентов втузов. Часть 1 / П. Е. Данко, А. Г. Попов, Т. Я. Кожевникова. – М. : Высш. шк., 1986. – 304 с.

3. Зарубин, В. С. Математической моделирование в технике : учеб. для вузов / под ред. В. С. Зарубина, А. П. Крищенко. – 2-е изд., стереотип. – М. : Изд-во МГТУ им. П. Э. Баумана, 2003. – 496 с. (Сер. Математика в техническом университете; Вып. XXI, заключительный).

4. Моденов, П. С. Аналитическая геометрия / П. С. Моденов. – М. : МГУ, 1969. – 699 с.

5. Пискунов, Н. С. Дифференциальное и интегральное исчисления для ВТУЗов / Н. С. Пискунов. – М. : Наука, 1978. – Том 1. – 416 с.

6. Лысов, В. А. Аппроксимация плоских криволинейных контуров гладкими кривыми на дискретном множестве опорных точек / В. А. Лысов, О. В. Шевченко, А. В. Щеголев // Научно-технический вестник Поволжья. – 2011. – № 3. – С. 145-149.

7. Лысов, В. А. Графоаналитическое моделирование динамики механических свойств металлов в технологических процессах изготовления глубокой вытяжкой / В. А. Лысов, А. И. Сердюк, О. В. Шевченко, А. В. Щеголев // Информационные технологии в проектировании и производстве. – 2012. – № 4. – С. 46-53.

8. Лысов, В. А. Графоаналитическое моделирование пооперационной динамики наклёпа в составе сервисных процедур САПР ТП глубокой вытяжки / В. А. Лысов, А. В. Щеголев // Оборонный комплекс – научно-техническому прогрессу России. – 2014. – № 2. – С. 22-29.

ВОЗМОЖНОСТИ КОМПЬЮТЕРНОЙ ГРАФИКИ В ЯЗЫКЕ PASCAL

А. Т. Аразашвили
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

Компьютерная (машинная) графика – область деятельности, в которой компьютеры используются в качестве инструмента как для создания изображений, так и для обработки визуальной информации, полученной из реального мира

В стандарте языка Pascal графический вывод не предусмотрен. Однако на разных типах компьютеров, в разных реализациях языка Pascal существуют различные программные средства графического вывода – специальные наборы данных, функций, процедур. Несмотря на такое разнообразие, имеются общие понятия и средства, свойственные любому варианту реализации графики в любом языке программирования.

Начиная с четвертой версии Turbo Pascal для IBM PC появилась мощная графическая библиотека, организованная в модуль Graph.

Любая программа, использующая графический режим, будет иметь одну и ту же структуру:

1) определение графического драйвера;
2) установка графического режима;
3) инициализация графического режима;
4) построение графических объектов;
5) закрытие графического режима.

Для подключения библиотеки графических функций и процедур в языке Pascal необходимо подключить к программе модуль Graph следующей строкой: Uses graph;

Взаимодействие программы и видеосистемы в графических режимах обеспечивают драйверы. Драйверы собраны в файлах, имеющих расширение BGI: CGA.BGI, EGAVGA.BGI, HERC.BGI, IBM8514.BGI, ATT.BGI, PC3270.BGI и другие. Драйвер – это специальная программа, осуществляющая управление тем или иным техническим средством ПК. Графический драйвер управляет графическим адаптером в графическом режиме.

Графические возможности конкретного адаптера определяются разрешением экрана, то есть общим количеством пикселей, а также количеством цветов. Кроме того, многие адаптеры могут работать с несколькими графическими страницами.

Для инициализации графического режима используется процедура: InitGraph (var Driver, Mode: integer; Path:string;), где Driver – переменная, определяющая тип графического драйвера; Mode – переменная, задающая режим работы графического адаптера; Path – выражение, содержащее путь доступа к файлу драйвера.

Для проверки успешности инициализации графического режима существует функция GraphResult, которая имеет тип результата integer, в котором

закодирован результат последнего обращения к графическим процедурам. Если ошибка не обнаружена, значением функции будет ноль, в противном случае – отрицательное число.

Завершает работу адаптера в графическом режиме и восстанавливает текстовый режим работы экрана процедура CloseGraph.

Процедуры и функции, входящие в модуль Graph, позволяют устанавливать цвета фона экрана и выводимого текста, перемещать указатель, устанавливать точки, рисовать линии, окружности, эллипсы, дуги, прямоугольники, параллелепипеды, закрашивать фигуры и выводить текст.

Для демонстрации возможностей компьютерной графики в языке Pascal с использованием модуля Graph приведём пример программы для построения кривой с названием «безумие» (madness).

```
Program Madness;
Uses CRT, Graph;
Var d,m,x,y : integer; t: real;
Begin
d:=detect; m:=detect; InitGraph(d,m,' '); t:=0;
repeat
x:=320+round(180*(sin(0.0099*t)-0.7*cos(0.0301*t)));
y:=240-round(200*(0.1*sin(0.1503*t)+cos(0.0101*t)));
PutPixel(x,y,white); t:=t+0.01;
until KeyPressed;
CloseGraph;
End.
```

В результате работы программы на экране монитора будет вычерчиваться кривая, показанная на рисунке 1, до тех пор, пока не будет нажата какая-либо клавиша на клавиатуре.

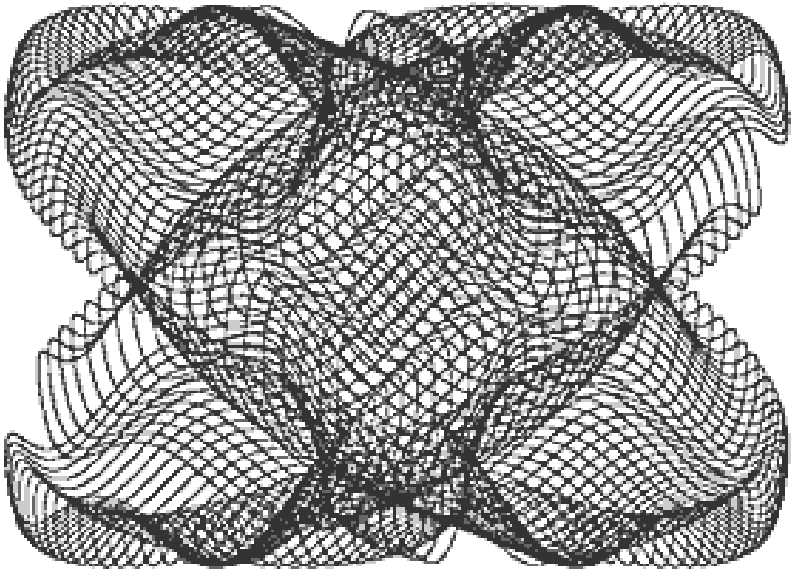

Рис. 1. Вид кривой «безумие»

С помощью процедур и функций модуля Graph можно имитировать движение или вращение графических объектов. Движение создается за счёт стирания изображения на старом месте и рисования его на новом месте (с небольшим сдвигом). При составлении таких программ широко используется задержка.

Одним из приложений компьютерной графики в языке Pascal является наглядное представление результатов математических расчётов. Нарисованный график функции, гистограмма или секторная диаграмма делают результаты математических расчётов обозримее, нагляднее и понятнее.

ОСНОВНЫЕ ВОПРОСЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Д. Ю. Блиничкин
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

С быстрым развитием информационных и коммуникационных технологий, с началом информационной революции особую актуальность и практическую значимость приобретает понятие «информационная безопасность».

Под информационной безопасностью (ИБ) понимают защищенность информации и поддерживающей инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера, которые могут нанести неприемлемый ущерб субъектам информационных отношений, в том числе владельцам и пользователям информации и поддерживающей инфраструктуры.

Защита информации – это комплекс мероприятий, направленных на обеспечение информационной безопасности. Угрозы ИБ – это оборотная сторона использования информационных технологий.

Из этого положения можно вывести два важных следствия:

1. Трактовка проблем, связанных с информационной безопасностью, для разных категорий субъектов может существенно различаться. Для иллюстрации достаточно сопоставить режимные государственные организации и учебные институты. В первом случае «пусть лучше все сломается, чем враг узнает хоть один секретный бит», во втором – «да нет у нас никаких секретов, лишь бы все работало».

2. Информационная безопасность не сводится исключительно к защите от несанкционированного доступа к информации, это принципиально более широкое понятие. Субъект информационных отношений может пострадать (понести убытки или получить моральный ущерб) не только от несанкционированного доступа, но и от поломки системы, вызвавшей перерыв в работе. Более того, для многих открытых организаций, например, учебных, собственно защита от несанкционированного доступа к информации стоит по важности отнюдь не на первом месте.

При изучении информационной безопасности очень важно понимать, что это не только техническое явление, а, в большей степени, явление социальное.

Помимо компьютеров информационная безопасность зависит от поддерживающей инфраструктуры, к которой можно отнести системы электро-, водо- и теплоснабжения, а также средства коммуникаций и другое. Информационная безопасность – это многогранная, даже многомерная, область деятельности человека, для обеспечения которой успех может принести только системный комплексный подход.

При анализе проблематики, связанной с информационной безопасностью, необходимо учитывать специфику данного аспекта безопасности, состоящую в том, что информационная безопасность есть составная часть информационных технологий – области, развивающейся беспрецедентно высокими темпами. Здесь важны не столько отдельные решения: законы, учебные курсы, программно-технические изделия, находящиеся на современном уровне, сколько механизмы генерации новых решений, позволяющие жить в темпе технического прогресса.

К сожалению, современная технология программирования не позволяет создавать безошибочные программы, что не способствует быстрому развитию средств обеспечения ИБ. Следует исходить из того, что необходимо конструировать надежные системы информационной безопасности с привлечением ненадежных компонентов – программ. В принципе, это возможно, но требует соблюдения определенных архитектурных принципов и контроля состояния защищенности на всем протяжении жизненного цикла информационных систем.

В таких условиях системы информационной безопасности должны уметь противостоять разнообразным атакам, как внешним, так и внутренним, атакам автоматизированным и скоординированным. Иногда нападение длится доли секунды; порой прощупывание уязвимых мест ведется медленно и растягивается на часы, так что подозрительная активность практически незаметна. Целью злоумышленников может быть нарушение всех составляющих ИБ – доступности, целостности или конфиденциальности.

Для защиты интересов субъектов информационных отношений необходимо сочетать меры следующих уровней:

- законодательного;
- административного: приказы и другие действия руководства организаций, связанных с защищаемыми информационными системами;
- процедурного: меры безопасности, ориентированные на людей;
- программно-технического.

Законодательный уровень является важнейшим для обеспечения информационной безопасности.

На законодательном уровне различаются две группы мер:

- меры, направленные на создание и поддержание в обществе негативного, в том числе с применением наказаний, отношения к нарушениям и нарушителям информационной безопасности, их называют мерами ограничительной направленности;

– направляющие и координирующие меры, способствующие повышению образованности общества в области информационной безопасности, помогающие в разработке и распространении средств обеспечения информационной безопасности, это меры созидательной направленности.

На взгляд специалистов в области ИБ, самое важное на законодательном уровне – создать механизм, позволяющий согласовать процесс разработки законов с реалиями и прогрессом информационных технологий.

Безопасность информационной системы зависит от окружения, в котором она функционирует. Необходимо принять меры для защиты зданий и прилегающей территории, поддерживающей инфраструктуры, вычислительной техники, носителей данных.

Различают четыре уровня защиты информации:

1) предотвращение – доступ к информации и технологии имеет только персонал, который получил допуск от собственника информации;

2) обнаружение – обеспечивается раннее обнаружение преступлений и злоупотреблений, даже если механизмы защиты были обойдены;

3) ограничение – уменьшается размер потерь, если преступление все-таки произошло, несмотря на меры по его предотвращению и обнаружению;

4) восстановление – обеспечивается эффективное восстановление информации при наличии документированных и проверенных планов по восстановлению.

Ранее контроль над защитой информации был заботой технических администраторов. Сегодня контроль над информацией стал обязанностью каждого пользователя. Это требует от пользователя новых знаний и навыков. Так, например, хороший контроль над информацией требует понимания возможностей совершения компьютерных преступлений и злоупотреблений, чтобы можно было в дальнейшем предпринять контрмеры против них.

ИБ подчеркивает важность информации в современном обществе – понимание того, что информация – это ценный ресурс, нечто большее, чем отдельные элементы данных. Информационной безопасностью называют меры по защите информации от неавторизованного доступа, разрушения, модификации, раскрытия и задержек в доступе. ИБ включает в себя меры по защите процессов создания данных, их ввода, обработки и вывода. Целью ИБ является обезопасить ценности системы, защитить и гарантировать точность и целостность информации, минимизировать разрушения, которые могут иметь место, если информация будет модифицирована или разрушена. ИБ требует учета всех событий, в ходе которых информация создается, модифицируется, к ней обеспечивается доступ или она распространяется.

ИСПОЛЬЗОВАНИЕ СРЕДСТВ ТЕХНОЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ НА НЕФТЕПЕРЕРАБАТЫВАЮЩИХ ПРЕДПРИЯТИЯХ

В. С. Богданова
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

Моделирование процессов и производств на современном предприятии невозможны без современных средств технологического моделирования.

Используя средства технологического моделирования, инженеры, работающие на производстве или в проектных организациях, получают современный мощный инструмент технологического расчета, позволяющий в несколько раз повысить скорость выполнения инженерных расчетов (количество) и глубже понять поведение установки, найти оптимальную совокупность рабочих параметров (качество). С 2005 года на ОАО «Орскнефтеоргсинтез» введена в эксплуатацию модель PIMS.

Модель содержит описание основных технологических установок НПЗ топливного и масляного производств, схему возможных материальных потоков, используемые виды сырья и вспомогательных материалов (утилит), схему смешения конечных продуктов.

Параметры модели:

– ограничения по производительности установок, отборы в каждом технологическом режиме работы;

– потребление и стоимость вспомогательных материалов;

– параметры качества нефти, полупродуктов и ограничения по качеству конечных продуктов;

– цены на сырье и продукты, ограничения по спрос.

Программа может быть использована как инструмент анализа сравнительной эффективности производства различных нефтепродуктов (полуфабрикатов и товарных), поскольку одним из побочных результатов оптимизации являются так называемые «теневые цены»; графа оценки содержит так называемые «теневые цены». Они формируются в результате решения задачи оптимизации и характеризуют, по какой цене можно было бы купить нефтепродукт на стороне, то есть какова предельная (с точки зрения ситуация на предприятии) цена этого продукта. По ним можно судить о том, насколько выгодно или невыгодно производить и использовать соответствующий продукт.

Для планирования на модели PIMS на предприятии существует специальная служба. Функции, которые выполняет Служба поддержки модели PIMS:

1. Построение модели и ее поддержка – актуальность и корректность.
2. Планирование НПЗ для различных периодов и ревизия планов:

– месяц;

– годовой план;

– пятилетний план.

3. Сравнение плана с фактом.

Анализ отклонения по различным причинам:

– изменение сырья;

– изменение спроса;

– изменение цен;

– изменение по заводским причинам;

– изменение из-за неисправности модели.

4. Определение и ревизия плановых значений КПЭ для завода и для отдельных установок с использованием модели.

5. Оценка и расчет экономических показателей для всех коммерческих проектов и проектов, связанных с изменением структуры и технологии завода.

6. Оценка будущего изменения качества, рыночных изменений и разные варианты развития завода.

7. Оценка ограничения в различных процессах планирования и разработка предложения для их устранения в тесном контакте с производственным отделом и с техническим/ технологическим отделом.

8. Отслеживание выполнения плана.

9. Отслеживание выходов установок и ключевых показателей качества продуктов и компонентов и запаса качества. Отображение этих изменений в модели и инициирование процесса улучшения.

10. Формулировка плана для внутреннего пользования в виде отчетов для производственного отдела и определение основных рецептур для смешения.

В модели заданы ограничения по мощности производственных установок в соответствии с их характеристиками, и по параметрам качества производимых продуктов – в соответствии со стандартами (ГОСТы и ТУ). Модель создана в декабре 2004 – январе 2005 года на основе существовавшей модели линейного программирования RPMS специалистами компаний Aspentech и ТНК.

Для эффективного планирования производства с применением средств оптимизации жизненно необходимо, чтобы работе с моделью были обучены и использовали ее при планировании производственной деятельности специалисты.

Сбор данных осуществляется при использовании таблиц, электронной почты и телефона, которые затем заносятся в таблицы, совместимые с моделью.

Необходимая для проверки расчетов исходная информация включает в себя следующие группы:

– список нефтепродуктов. В этот список входят сырье (нефть и нефтепродукты, поступающие со стороны), полуфабрикаты, товарная продукция. На сырье указываются цены и объемы поставок. На товарные продукты – предполагаемые цены реализации и, возможно, ограничения на объемы производства. Кроме того, задаются требования по качеству товарных продуктов, получаемых смещением, и качеству компонентов смеси;

– список установок и описание каждой установки в отдельности (номинальная мощность, входные и выходные нефтепродукты, коэффициенты отбора, нормы расхода производственных ресурсов). По описанию установок автоматически формируется схема материальных потоков;

– сведения, необходимые для расчета и анализа основных технико-экономических показателей.

Программа выдаёт оптимальный технико-экономический план при заданных условиях, а также рекомендации о том, как нужно изменить эти условия (разумеется, если это возможно), чтобы улучшить основные технико-экономические показатели. Если условия, заданные пользователем, окажутся противоречивыми, программа предлагает компромиссный вариант, ослабляет невыполнимые требования. Выходная информация включает в себя:

– оптимальный план производства товарной продукции в натуральном и стоимостном выражении;

– загрузку и баланс каждой установки. Оценку каждой установки с точки зрения ее роли в достижении максимального экономического результата;

– материальный баланс по каждому нефтепродукту и всему производству в целом. Сведения, позволяющие судить о том, насколько выгодно или невыгодно производить каждый нефтепродукт.

Перспективы развития модели:

1. Информационная интеграция системы планирования с системами корпоративного учета и контроля на базе существующих ERP-систем.

2. Оперативное планирование нефтепереработки:

а) маржинальный анализ производства, анализ чувствительности производства;

б) оптимизация межзаводской логистики для групп НПЗ;

в) выявление и экономико-технологический анализ «узких мест» производства и оценка мероприятий по их «расшивке»;

г) ретроспективный (факторный) анализ экономики производства.

4. Среднесрочное планирование распределения нефти и нефтепродуктов с целью определения множества вероятностных сценариев развития рынка для подготовки принятия оперативных решений:

а) анализ инвестиционных проектов;

б) анализ потенциального эффекта от реализации инфраструктурных компаний-конкурентов и транспортных операторов;

в) оперативная оценка и адаптация к изменениям в тарифной политике транспортных операторов;

г) анализ экономической целесообразности принятия политических решений (например, доли рынка по регионам присутствия);

д) детализированный ситуационный анализ положения компании на региональных рынках относительно конкурирующих компаний (при наличии оптимизационной модели отрасли в целом).

5. Стратегическое планирование распределения нефти и нефтепродуктов в качестве аналитического инструмента для поддержки принятия решений по стратегии развития компании с учетом стратегий конкурентов и планов государственного развития нефтяной отрасли:

а) визуализация и анализ всех возможных логистических цепочек по всему рынку России и СНГ;

б) анализ эффективности стратегических планов компании в зависимости от планов конкурентов;

в) поддержка принятия решений по участию в совместных долгосрочных инфраструктурных проектах (новые НПЗ, маршруты, терминалы);

г) экономическая аргументация при диалоге с уполномоченными государственными структурами (например, при принятии новых тарифных политик, условий налогообложения);

д) расчеты по различным конъюнктурным сценариям на основе макропараметров (например, динамики потребления, профилей добычи , прогнозов цен на сырьё).

МОДЕЛИРОВАНИЕ ПРОЦЕССА ОБСЛУЖИВАНИЯ ЗАЯВОК В ТРАНСПОРТНОЙ ОРГАНИЗАЦИИ

А. А. Жигарева
Национальный исследовательский университет «МЭИ», г. Смоленск

При решении многих сложных практических задачи одним из наиболее действенных и распространённых методов является метод моделирования. Конечно, данный метод представляется весьма разноплановым, базирующимся на широком спектре научных методологий и подходов. Однако даже поверхностный анализ предметных областей, в рамках которых планируется применить моделирование, может выявить также особенности, которые могут склонить исследователя к выбору в пользу имитационного решения задач. К таким особенностям, в частности, можно отнести наличие факторов, аналитическое описание которых неизвестно или слишком сложно.

Одной из таких предметных областей выступает обслуживание заявок транспортной организации ПАО «Колесо», занимающейся грузовыми и пассажирскими перевозками. К основным средствам организации относятся 5 мини-ЭВМ, соединённых в локальную сеть. На этих рабочих местах находятся диспетчеры, которые обслуживают заявки на перевозку. В часы пик и праздничные дни часто возникает ситуация «аврального режима», характеризуемая большим потоком заявок на обслуживание. В этой связи была предложена попытка оптимизации структуры вычислительного процесса, которая позволила бы минимизировать издержки во время авральных режимов.

На первом этапе решения данной задачи был проведён анализ предметной области, заключавшийся в изучении интенсивности потоков заявок, времени обслуживания заявок диспетчерами, сбора характеристики об очередях на обслуживание, времени ожидания и т. д.

В ходе этого анализа был сделан вывод о том, что в рассматриваемой предметной области для описания ряда факторов удобно использовать случайные величины. Особенностью рассматриваемой проблемы является то, что по-

мимо детерминированных факторов, которые могут быть учтены аналитическими методами, присутствуют ещё и случайные воздействия и возмущения [1].

При разработке модели в данной ситуации была выбрана одна из концепций имитационного моделирования, базирующаяся на теории массового обслуживания, в ее основе лежит представление предметной области в виде совокупности источников сообщений, очередей, каналов обслуживания.

Следующим этапом разработки был выбор среды создания модели. Проанализировав такие системы, как SLX, Arena, MODSIM, GPSS, был сделан выбор в сторону программной среды GPSS. Данный выбор обоснован тем, что имеется возможность получения бесплатной студенческой версии, которой в то же время достаточно для решения многих прикладных задач.

Задача оптимизации работы системы обработки информации была поставлена в службе такси, где имеется 5 ЭВМ. Сообщения с периодичностью 26±10 микросекунд поступают на мультиплексорный канал, где они обрабатываются в течение 20±5 микросекунд, а затем разделяются по пяти очередям. Сообщение поступает на ту ЭВМ, где очередь наименьшая, при этом очередь каждой ЭВМ не может превышать 5. Если заявке отказано в обработке, что несёт за собой 120 рублей убытка, то возможно увеличение ёмкости входных накопителей ЭВМ, единица ёмкости одного входного накопителя обходится в 18 рублей. Во-вторых, если суммарное количество сообщений во всех входных накопителях всех ЭВМ превышает значение 15, то автоматически происходит переход всех ЭВМ в авральный режим. При этом убытки за каждое сообщение составят k*6 рублей, а за каждую секунду работы – 4 рубля. Значение константы k выбиралось из диапазона [10; 135]. В ходе эксперимента было выяснено, при каких емкостях входных накопителей ЭВМ достигается минимум суммарных затрат. Значение k было взято 25. Сообщения поступают в модель, попадают в очередь в мультиплексорный канал, где буферизуются и предварительно обрабатываются. После этого происходит выбор самой короткой из пяти очередей ЭВМ, заявка попадает в ЭВМ и обрабатывается, после чего покидает модель. В случае отказа в обработке заявки, во-первых, считаются убытки, во-вторых, ёмкость входных каналов всех ЭВМ увеличивается вместе с затратами.

Сообщения поступают в модель, попадают в очередь в мультиплексорный канал, где буферизуются и предварительно обрабатываются. После этого происходит выбор наименьшей из пяти очередей ЭВМ, транзакт попадает в мини-ЭВМ и обрабатывается, после чего покидает модель. В случае отказа в обработке транзакта, во-первых, считаются убытки, во-вторых, ёмкость входных каналов всех мини-ЭВМ увеличивается вместе с затратами. Если сумма всех транзактов в очередях ко всем мини-ЭВМ превышает 15, то происходит переход в авральный режим, что также, в свою очередь, приводит к затратам. В результате проведения экспериментов стало возможным определить ёмкость входных накопителей, при которых суммарные издержки становятся минимальными. Для этого эксперимент был проведён несколько раз, пока не было получено минимальное значение.

После разработки модели было выполнено планирование эксперимента, которое позволило оптимальным образом решить поставленную задачу оптимизации вычислительной структуры. Учитывая, что нас интересуют всевозможные сочетания уровней факторного пространства, выбор был сделан в сторону полного факторного эксперимента. Факторное пространство включает в себя: коэффициент k, ёмкость входных накопителей ЭВМ. Целевой функцией являются суммарные затраты.

Разработанный текст программы имеет стандартную структуру, характерную для событийно-ориентированного языка GPSS. Особый интерес представляет фрагмент кода, представленный ниже, реализующий процесс выбора ЭВМ с минимальной длиной очереди:

```
SELECT MIN EVM,1,5,,Q ; выбор наименьшей очереди
TEST LE Q*$EVM,5,M_emkost ; проверяем длину очереди, которая не должна быть > 5
Queue P$EVM ; объявление очереди
seize P$EVM ; транзакт поступает в эвм
depart P$EVM ; транзакт выходит из очереди
advance 135 ; обработка в ЭВМ в течение 135 ед. машинного времени
release P$EVM ; освобождается эвм
transfer ,KILL ; выход транзакта из модели.
```

В приведённом фрагменте кода представлен выбор минимальной очереди к мини-ЭВМ, для чего применяется оператор SELECT. Проводим проверку, длина очереди не должна быть больше 5. Если длина очереди превышает допустимое значение, то происходит переход на метку M_emkost, где происходит увеличение ёмкости входных накопителей. Далее объявляется очередь, которая является минимальной из пяти очередей, занимается соответствующая мини-ЭВМ, после чего транзакт выходит из очереди, обрабатывается в соответствующей мини-ЭВМ 135 единиц машинного времени, потом освобождаем мини-ЭВМ. После этого транзакт переходит на метку KILL, где выходит из модели.

В ходе проведения работы были получены следующие результаты:

– оптимальная ёмкость входного накопителя для каждой ЭВМ равна 8;

– суммарные затраты (убытки от отказов транзактов в обслуживании, затраты на увеличение ёмкости, поддержка аврального режима) составляют 30762 единиц стоимости.

Применённый подход к решению задачи минимизации не является единственно возможным для решения данной задачи, но обладает такими явными преимуществами, как простота и эффективность.

Библиографический список

1. Акопов, А. С. Имитационное моделирование : учебник и практикум для академического бакалавриата / А. С. Акопов. – М. : Юрайт, 2014. – 388 с.

МЕТОД АВТОМАТИЗИРОВАННОГО ПОСТРОЕНИЯ ЦИКЛОГРАММ В ИССЛЕДОВАНИИ ГИБКИХ ПРОИЗВОДСТВЕННЫХ СИСТЕМ

Е. В. Иванникова
Оренбургский государственный университет, г. Оренбург

Изготовление любого изделия характеризуется определенным производственным циклом, то есть периодом времени, за который материал или заготовка проходит все операции производственного процесса (или его части) и превращается в готовую продукцию (или в готовую ее часть).

Графически производственный цикл можно представить с помощью цикловых диаграмм, или *циклограмм*. Циклограммой, или цикловой диаграммой, называется схема согласованности перемещений оборудования или их исполнительных органов во времени в сложных технологических системах, работающих по заданному циклу. Построение циклограмм работы оборудования визуализирует качество работы, позволяя оценить время простоев, относительно времени работы оборудования. Циклограмма также позволяет выявить причины простоев и, соответственно, найти путь к их сокращению до минимально необходимых, например, для профилактического осмотра или ремонта. Циклограммы работы оборудования применяются при проектировании и анализе работы производственных систем, в которых процессы имеют детерминированный, упорядоченный характер, например, роботизированных технологических комплексах, автоматических линиях и гибких производственных системах. Такие производственные системы имеют в своем составе оборудование, работа которого должна быть отлаженной и ритмичной. Примером может служить гибкий автоматизированный участок, состоящий из 2-8 станков, обслуживаемый несколькими транспортными средствами, изображенный на рисунке 1.

Для анализа его работы можно применить программный пакет «Комплекс v.2.1», который является доработанной версий «Комплекс v.1.1». Ввод паспортных данных оборудования и выпускаемой продукции, а также результаты моделирования показаны на рисунке 2.

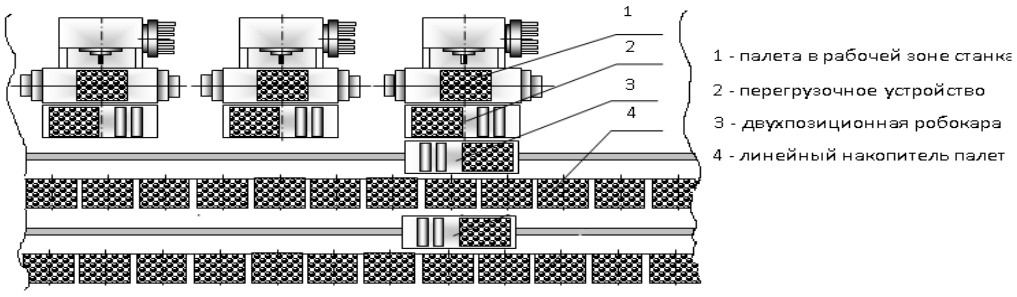

Рис. 1. Схема ГАУ, обслуживаемого двумя робокарами и линейным двухрядным накопителем паллет

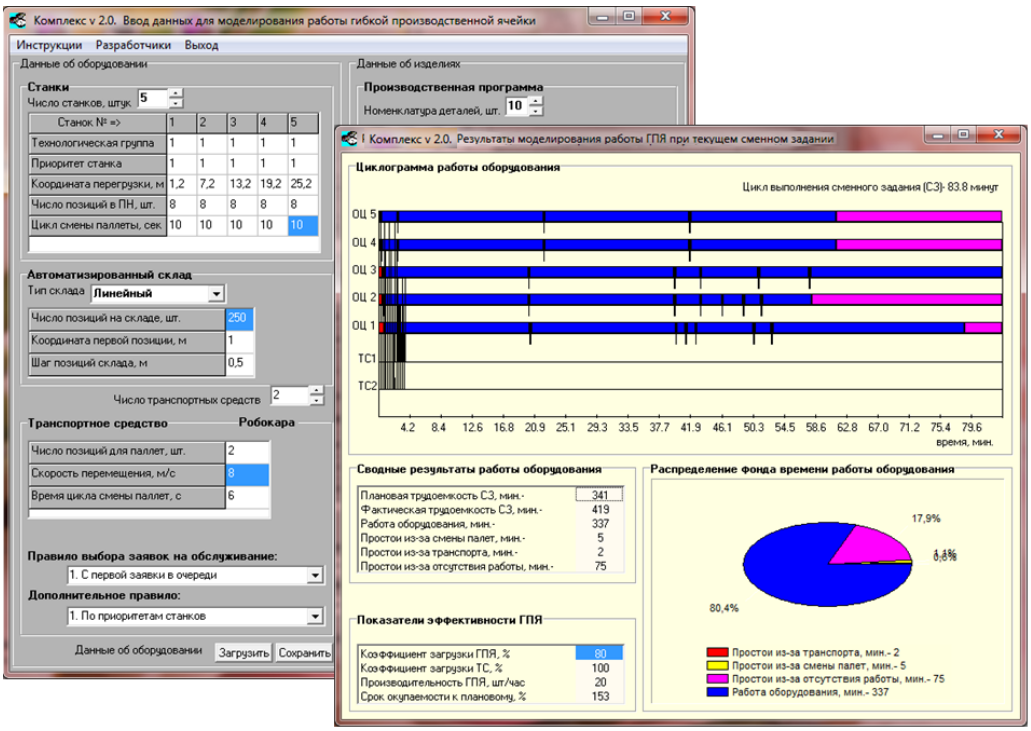

Рис. 2. Ввод данных и результаты моделирования

В данном примере моделирование проводилось для 5 единиц основного технологического оборудования и двух единиц транспортных средств. Циклограмма работы оборудования ГАУ и круговая диаграмма представлены на рисунке 3. Варьируя значения скоростей транспортных средств, можно изменить коэффициент загрузки оборудования до необходимого, который обычно составляет $k_{ГАУ} = 0{,}85…0{,}95$.

ОБЗОР ИНТЕГРИРОВАННЫХ СРЕД РАЗРАБОТКИ ДЛЯ ПРОГРАММИРОВАНИЯ НА ЯЗЫКАХ C/C++

И. И. Карманович
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

Интегрированная среда разработки (IDE – Integrated development environment) – это комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО).

Среда разработки включает в себя:

– текстовый редактор;
– компилятор и/или интерпретатор;
– средства автоматизации сборки;
– отладчик.

Иногда IDE содержит также средства для интеграции с системами управления версиями и разнообразные инструменты для упрощения конструирова-

ния графического интерфейса пользователя. Многие современные среды разработки также включают браузер классов, инспектор объектов и диаграмму иерархии классов для использования при объектно-ориентированной разработке ПО.

Интегрированные среды разработки были созданы для того, чтобы максимизировать производительность программиста благодаря тесно связанным компонентам с простыми пользовательскими интерфейсами. Это позволяет разработчику сделать меньше действий для переключения различных режимов, в отличие от дискретных программ разработки.

IDE обычно представляет собой единственную программу, в которой проводится вся разработка. Она, как правило, содержит много функций для создания, изменения, компилирования, развертывания и отладки программного обеспечения. Цель интегрированной среды заключается в том, чтобы объединить различные утилиты в одном модуле, который позволит абстрагироваться от выполнения вспомогательных задач, тем самым давая возможность программисту сосредоточиться на решении собственно алгоритмической задачи и избежать потерь времени при выполнении типичных технических действий (например, вызове компилятора). Таким образом повышается производительность труда разработчика. Также считается, что тесная интеграция задач разработки может далее повысить производительность за счёт возможности введения дополнительных функций на промежуточных этапах работы. Например, IDE позволяет проанализировать код и тем самым обеспечить мгновенную обратную связь и уведомить о синтаксических ошибках.

Для создания программ на языке C/C++ обычно используются следующие интегрированные среды разработки:

– GCC (GNU Compiler Collection) – набор компиляторов проекта GNU (фонд разработки свободного программного обеспечения) с поддержкой таких языков программирования, как: Ada, C, C++, Fortran, Java, Objective-C, Objective-C++, Go. Используется как компилятор для большинства операционных систем семейства Linux. Использование этого компилятора удобно в случае, если планируется создавать кросс-платформенное приложение либо использовать в своей программе библиотеки, созданные в рамках сообщества разработчиков свободного программного обеспечения.

– Microsoft Visual C++ – это C/C++ компилятор и компоновщик, библиотека стандартных шаблонов (STL) и .NET runtime.

– Borland C++ – одна из качественных IDE, включающая сам компилятор языка C/C++, компоновщик, компилятор ресурсов, C++ Win32 препроцессор, утилиту для создания lib файлов из dll и другие возможности для создания программ под ОС семейства Win32.

– Open Watcom – проект сообщества открытого кода по поддержке и развитию многоплатформенных компиляторов Watcom C, C++ и Fortran и сопутствующих программ. Этот компилятор генерирует компактный и быстрый код, но на текущий момент он не поддерживает полностью стандарт C++. В отличие от всех представленных в данном обзоре компиляторов, Open Watcom заметно

отличается наличием простого графического редактора, графического дебагера, редактора ресурсов и других утилит.

– Digital Mars C++ Compiler – набор программного обеспечения для написания программ на C/C++, который, помимо самого компилятора, содержит графическую среду разработки совместно с дебагером, справку, различные библиотеки. Данный компилятор работает только в ОС Windows.

– Dev-C++ – бесплатная интегрированная среда для программирование на языках C/C++ с поддержкой компилятора GCC, интегрированной отладкой, менеджером проекта, настраиваемым редактором кода с подсветкой синтаксиса, просмоторщиком классов, поддержкой профилей и шаблонов.

– NetBeans IDE – интегрированная среда разработки приложений, бесплатная IDE с открытым исходным кодом. Предназначена для профессиональной разработки десктоп приложений, web-приложений, корпоративных систем, программ для мобильных устройств. NetBeans – единственная IDE, которая устроит и начинающего разработчика, и профессионала.

– Eclipse – бесплатная программная платформа с открытым исходным кодом, контролируется организацией Eclipse Foundation. Написана на языке программирования Java и основной целью её создания является повышение продуктивности процесса разработки программного обеспечения.

IDE, разработанные на базе платформы Eclipse, применяются для создания программного обеспечения на различных языках программирования. Eclipse является платформой для разработки любых интегрированных сред программирования и практически любого клиентского программного обеспечения.

– Microsoft Visual Studio Express – набор ограниченных средств разработки для языков Visual Basic, C#, C++ и Visual Web Developer Express. Позиционируется Microsoft как IDE начального уровня для лиц, не занимающихся профессионально программированием. Графический интерфейс среды позволяет создавать оконные, а подключив и MSDN, можно пользоваться справочным пособием по языку C/C++ и API Windows.

БИОМЕТРИЧЕСКАЯ СИСТЕМА ЗАЩИТЫ ИНФОРМАЦИИ

Д. Г. Кудлай, О. С. Вежлева
Оренбургский государственный университет, г. Оренбург

Во все времена была проблема осуществлять санкционированный допуск клиента к объекту или секретной информации, признавать своих и не допускать чужих.

Для этого использовали всевозможные средства: ключи, чипы, пропуски с фотографией личности, паспорт с визой, карточки, пароли и пр. Аутентификация – проверка соответствия традиционными методами – ущербна и часто дает сбои: можно подделать, потерять, забыть.

В криминалистике давно уже применяют отпечатки пальцев, исследование ДНК, и это оправдывает себя. Биометрическая система защиты позволяет идентифицировать пользователя по биологическим характеристикам.

Безопасность – это субстанция, которую трудно оценить количественно, поскольку сложно представить себе клиента, жертвующего собственной безопасностью из соображений экономии. Рост террористической угрозы и необходимость совершенствования систем обеспечения безопасности привели к тому, что объем рынка биометрического оборудования в последнее время начал быстро расти.

Уже в 2006 г. граждане стран Евросоюза стали обладателями так называемых электронных паспортов – документов, построенных на специальной микросхеме, в которой записаны некоторые биометрические данные владельца (например, информация об отпечатках пальцев, радужной оболочке глаза), а также сопутствующие гражданские данные (номера карточки страхования, водительского удостоверения и т. п.). Область применения таких документов практически не ограничена: их можно использовать как международные удостоверения личности, кредитные карты, медицинские карты, страховые полисы, пропуска – список можно продолжать и продолжать.

В некоторых областях биометрия уже активно используется на протяжении нескольких лет. И одна из таких областей – компьютерная безопасность. Самое распространенное решение на базе биометрических технологий – это идентификация (или верификация) по биометрическим характеристикам в корпоративной сети или при запуске рабочей станции (ПК, ноутбук и т. д.).

Биометрическое распознавание объекта заключается в сравнении физиологических или психологических особенностей этого объекта с его характеристиками, хранящимися в базе данных системы. Главная цель биометрической идентификации заключается в создании такой системы регистрации, которая бы крайне редко отказывала в доступе легитимным пользователям и в то же время полностью исключала несанкционированный вход в компьютерные хранилища информации. По сравнению с паролями и карточками такая система обеспечивает гораздо более надежную защиту, ведь собственное тело нельзя ни забыть, ни потерять.

Достаточно широкое распространение получили средства криптографической защиты, в которых доступ к ключам шифрования предоставляется только после биометрической идентификации их владельца. Надо отметить, что в сфере компьютерной безопасности шаблон используемой биометрической характеристики, как правило, подвергается одностороннему преобразованию, то есть из него нельзя путем обратной процедуры восстановить отпечаток пальца или рисунок радужной оболочки глаза.

Как известно, аутентификация подразумевает проверку подлинности субъекта, которым в принципе может быть не только человек, но и программный процесс.

Биометрические технологии аутентификации можно разделить на две большие категории – физиологические и психологические. К первой относятся

методы, основанные на физиологической характеристике человека, то есть неотъемлемой, уникальной характеристике, данной ему от рождения. Здесь анализируются такие признаки, как черты лица, структура глаза (сетчатки или радужной оболочки), параметры пальцев (папиллярные линии, рельеф, длина суставов и т. д.), ладонь (ее отпечаток или топография), форма руки, рисунок вен на запястье или тепловая картина.

К группе психологических относят так называемые динамические методы, которые основываются на поведенческой (динамической) характеристике человека. Иными словами, они используют особенности, характерные для подсознательных движений в процессе воспроизведения какого-либо действия. К таким характеристикам относятся голос человека, особенности его подписи, динамические параметры письма, особенности ввода текста с клавиатуры и т. д.

Любая биометрическая система позволяет распознавать некий шаблон и устанавливать аутентичность конкретных физиологических или поведенческих характеристик пользователя. Логически биометрическую систему можно разделить на два модуля: регистрации и идентификации.

Модуль регистрации отвечает за то, чтобы система научилась идентифицировать конкретного человека. На этапе регистрации биометрические датчики сканируют его необходимые физиологические или поведенческие характеристики, создавая их цифровое представление. Специальный модуль обрабатывает это представление, с тем чтобы выделить характерные особенности и сгенерировать более компактное и выразительное представление, называемое шаблоном. Для изображения лица такими характерными особенностями могут быть размер и относительное расположение глаз, носа и рта.

Модуль идентификации отвечает за распознавание человека. На этапе идентификации биометрический датчик регистрирует характеристики человека, идентификация которого проводится, и преобразует эти характеристики в тот же цифровой формат, в котором хранится шаблон. Полученный шаблон сравнивается с хранимым, с тем чтобы определить, соответствуют ли эти шаблоны друг другу. При использовании в процессе аутентификации технологии идентификации отпечатков пальцев имя пользователя вводится для регистрации, а отпечаток пальца заменяет пароль. Эта технология использует имя пользователя в качестве указателя для получения учетной записи пользователя и проверки соответствия "один к одному" между шаблоном считанного при регистрации отпечатка и сохраненным ранее шаблоном для данного имени пользователя. В другом случае введенный при регистрации шаблон отпечатка пальца сопоставляется со всем набором сохраненных шаблонов.

Основные статические методы защиты информации:

1. По отпечатку пальца

В основе этого метода лежит уникальность рисунка папиллярных узоров на пальцах у каждого человека. Отпечатки пальцев – наиболее точная, дружественная к пользователю и экономичная биометрическая характеристика из всех, используемых в компьютерных системах идентификации. Устраняя для пользователей потребность в паролях, технология распознавания отпечатков

пальцев сокращает число обращений в службу поддержки и снижает расходы на сетевое администрирование.

Преимущества доступа по отпечатку пальца – простота использования, удобство и надежность. Вероятность ошибки при идентификации пользователя намного меньше, чем у других биометрических методов.

2. По геометрии руки

В этой технологии оценивается несколько десятков различных характеристик, включая размеры самой ладони в трех измерениях, длину и ширину пальцев, очертания суставов и т. п. С помощью специального устройства, состоящего из камеры и нескольких подсвечивающих диодов (включаясь по очереди, они дают разные проекции ладони), строится трехмерный образ кисти руки.

3. По геометрии лица

Идентификация человека по лицу, без сомнения, самый распространенный способ распознавания в обычной жизни. Но в плане технической реализации она представляет собой более сложную задачу, нежели распознавание отпечатков пальцев, и требует более дорогостоящей аппаратуры.

4. По радужной оболочке глаз

Довольно надежное распознавание обеспечивают системы, анализирующие рисунок радужной оболочки глаза человека. Дело в том, что эта часть человеческого организма весьма стабильна. Она практически не меняется в течение всей жизни, не зависит от одежды, загрязнений и ран. Заметим также, что оболочки правого и левого глаза по рисунку существенно различаются.

5. По сетчатке глаза

Для того, чтобы зарегистрироваться в специальном устройстве, достаточно посмотреть в глазок камеры менее минуты. За это время система успевает подсветить сетчатку и получить обратно отраженный сигнал. Для сканирования сетчатки используется инфракрасное излучение низкой интенсивности, направленное через зрачок к кровеносным сосудам на задней стенке глаза. Из полученного сигнала выделяется несколько сотен первоначальных характерных точек, информация о которых усредняется и сохраняется в кодированном файле.

К динамическим методам защиты относятся:

1. По голосу

Существует достаточно много способов построения кода идентификации по голосу; как правило, это различные сочетания частотных и статистических характеристик голоса. Здесь могут оцениваться такие параметры, как высота тона, модуляция, интонация и т. п.

2. По рукописному почерку

Как оказалось, подпись – это такой же уникальный атрибут человека, как и его физиологические характеристики. Кроме того, метод идентификации по подписи более привычен для любого человека, поскольку он, в отличие от снятия отпечатков пальцев, не ассоциируется с криминальной сферой.

Биометрическая система защиты данных уже сегодня решает ряд важных задач на самом высоком уровне (голосование, вопросы миграции, работа спецслужб). Биометрия является лучшей альтернативой традиционным методам, уже сегодня в мире более 300 компаний занимаются этими разработками и производством, емкость рынка – десятки миллиардов долларов. Это впечатляет, но не все так радужно, существуют проблемы в вопросах биометрической защиты информации.

Любые сканеры можно обмануть, оцифрованную биометрическую информацию можно потерять. И это пока самые уязвимые места биометрии. Невысока надежность сканеров при считывании биологических параметров человека, на их работу может влиять запыленность, влажность помещения.

Грязные руки, плохое освещение, плохо выставленное к объективу лицо, измененный из-за болезни или стресса голос приводят к искажению информации.

Страхи и предубеждения людей, которые не лишены основания, тоже можно отнести к минусам биометрических систем безопасности.

Таким образом, биометрическое распознавание обеспечивает более надежную аутентификацию пользователей, чем пароли и удостоверяющие личность документы, и является единственным способом обнаружения самозванцев. Хотя биометрические системы не являются абсолютно надежными, исследователи сделали значительные шаги вперед по пути идентификации уязвимостей и разработки мер противодействия им. Новые алгоритмы для защиты биометрических шаблонов частично устраняют опасения по поводу защищенности систем и приватности данных пользователя, но понадобятся дополнительные усовершенствования, прежде чем подобные методы будут готовы к применению в реальных условиях.

КЛАССИФИКАЦИЯ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ИЗДЕЛИЮ

М. А. Кузниченко
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

Цель разработки программного изделия состоит в том, чтобы уложившись в отведённое время и бюджет, разработать качественное программное обеспечение, удовлетворяющее реальные потребности пользователей. Успех таких проектов зависит от хорошо организованного управления требованиями. Ошибки в формулировке требований – наиболее часто встречающийся тип ошибок при разработке программных систем, а их устранение является наиболее дорогостоящим.

В зависимости от того, где и когда при работе над проектом программного приложения был обнаружен дефект, цена его может разниться в 50-100 раз. Причина состоит в том, что на его исправление придётся затратить средства на повторную спецификацию, повторное проектирование, повторное кодирование,

повторное тестирование, внесение исправлений, возврат дефектных версий и замену их на обновлённые, на обслуживание, создание нового пакета документации и многое другое.

Таким образом, ошибки и неточности в определении требований приводят к затратам, составляющим 25%-40% бюджета проекта разработки в целом.

Требование – это условие, которому должно удовлетворять программное обеспечение, или свойство, которым оно должно обладать, чтобы, с одной стороны, удовлетворить потребность пользователя в решении некоторой задачи, а с другой стороны, удовлетворить требования контракта, спецификации или стандарта. Управление требованиями – это систематический процесс выявления, организации и документирования требований к любой сложной системе, а также процесс, в ходе которого вырабатывается и обеспечивается соглашение между заказчиком и выполняющей проект группой.

Учитывая, что к программному изделию будут предъявлены сотни или даже тысячи требований, очень важно организовать и классифицировать их. Требования следует формулировать так, чтобы они были доступны для ознакомления и понимания участниками проекта.

Спецификация требований к программному изделию является основным документом, определяющим план его разработки. Все требования, указанные в спецификации, делятся на две большие группы: функциональные и нефункциональные. Функция – это предоставляемое системой обслуживание для удовлетворения одной или нескольких потребностей клиента.

Функциональные требования определяют действия, которые должна выполнять система, без учета ограничений, связанных с ее реализацией. Тем самым функциональные требования определяют поведение системы в процессе обработки информации. Функциональные требования определяют, что должно выполнять программное изделие, и выводятся непосредственно из логической модели, которая вытекает из требований пользователя. Для количественного выражения некоторые из функциональных требований могут включать атрибуты эксплуатационных характеристик, например, производительность, емкость.

Эксплуатационные требования определяют значения измеряемых переменных, характеризующих работу программного изделия. Представляются в виде отдельных требований или количественных атрибутов функциональных требований. Например, количественные требования записывать: «время ответа должно быть не более х сек».

Нефункциональные требования не определяют поведение системы, но описывают атрибуты системы или атрибуты системного окружения. Нефункциональные требования к программному изделию подразделяются на следующие категории: требования к пользовательскому интерфейсу, операционные требования, требования к ресурсам, требования к приёмному тестированию, качеству, надёжности, защите информации, сопровождению, безопасности и другие.

Требования к интерфейсам описывают элементы технических средств, программного обеспечения, баз данных, с которыми должен взаимодействовать

программный продукт. Требования к интерфейсам с техническими средствами определяют необходимую их конфигурацию. Требования к программному обеспечению могут включать требования к типу и версии операционной системы, прикладным пакетам, типу СУБД.

Требования к внешним интерфейсам могут обусловить использование конкретного сетевого протокола передачи информации, определенного языка описания документов и т.п.

Требования к интерфейсам можно проиллюстрировать с помощью специальных структурных схем, описывающих взаимодействие программного изделия с окружающей обстановкой.

Операционные требования регламентируют, как будет работать система, как она будет связываться с операторами или пользователями программного изделия.

Требования должны включать все интерфейсы пользователя и требования к человеко-машинному взаимодействию. Например, форматы экранов, содержание сообщений об ошибках, справочная информация, выдаваемая в качестве подсказок пользователю, цветовое оформление.

Требования к ресурсам устанавливают верхние пределы для характеристики технических средств: скорость процессора, емкость внешней и оперативной памяти.

Требования на верификацию программного изделия и на приемное тестирование описывают, как проверяется корректность принимаемых решений на каждом этапе жизненного цикла программного изделия, могут включать требования к моделированию окружающей обстановки, интерфейсов программного изделия. Требования к приемному тестированию определяют условия проведения аттестации разработанного программного изделия.

Требования к защите информации включают требования к обеспечению конфиденциальности и целостности информации: команды блокировки, системы паролей, защита от вирусов, ограничение доступа к данным и запрещение отдельных операций с данными для разных категорий пользователей.

Требования к качеству охватывают специфические атрибуты программного изделия, которые гарантируют, что функционирование изделия будет соответствовать поставленным целям. Показатели качества должны быть выражены в количественных величинах.

Показатели: надежность программного изделия, пригодность его к сопровождению, безопасность – описываются отдельно. Требования надежности определяются либо значением допустимого среднего времени между отказами, либо значениями минимального времени между отказами. Здесь необходимо предусмотреть возможности восстановления работы программы после сбоев.

Требования на пригодность к сопровождению – требования простоты исправления ошибок при отказах, легкости адаптации к конкретным операционным условиям и простоты модернизации программного изделия при изменении требований пользователя и при совершенствовании программного изделия в процессе его эксплуатации.

По возможности, требования представляются количественными показателями: время исправления отказа, коэффициент готовности. Требования могут включать ряд ограничений, отражающих возможность организации, занятой сопровождением.

Требование к безопасности – ряд дополнительных требований к программному изделию, которые обусловлены опасностью отказов программного изделия.

Требования к документации дополняют требования, содержащиеся в стандартах на документацию.

Для корректного и полного выявления требований к программному изделию наиболее подходящим является метод моделирования бизнес-процессов. Существует множество подходов к моделированию, среди которых наиболее популярны структурный и объектно-ориентированный подходы. Всё более востребованным среди разработчиков является унифицированный язык моделирования UML, имеющий в своём арсенале множество диаграмм, описывающих систему с разных точек зрения.

Для описания требований к будущей системе и моделирования бизнес-процессов наиболее наглядной является модель прецедентов (business use case model) и модель объектов бизнес-процесса (business object model). Модель прецедентов бизнес-процесса представляет собой модель предполагаемых функций бизнес-единицы и используется в качестве исходной информации для выявления ролей и взаимосвязей в организации. Модель объектов бизнес-процесса описывает сущности и то, как они взаимодействуют в процессе создания функциональных требований.

Библиографический список

1 Лэффингуэл, Д. Принцип работы с требованиями к программному обеспечению. Унифицированный подход ; пер. с англ. / Дин Лэффингуэл, Дон Уидриг. – Москва : Вильямс, 2010. – 448 с. : ил. – ISBN 5-8459-0275-4.

2 Вигерс, К. Разработка требований к программному обеспечению ; пер. с англ. / Карл Вигерс, Джой Битти. – 3-е изд., доп. – М. : Издательство «Русская редакция», 2014. – 736 с. : ил. – ISBN 978– 5-7502– 0433-5.

НЕОБХОДИМОСТЬ АНТИВИРУСНОЙ ЗАЩИТЫ КОМПЬЮТЕРНЫХ УСТРОЙСТВ

А. А. Мельникова

Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

С каждым днем в нашей жизни появляется все больше и больше новейшей современной техники: компьютеры, нетбуки, смартфоны и другое. Ежедневно современному человеку приходится работать со всем этим. Но мы должны быть

уверены, что наши устройства и информация, хранящаяся на них, должны быть исправны.

Современный Интернет заполонен разнообразными вирусами – червями, троянами, вредоносными кодами, готовыми в любой момент пробить брешь в защите электронного устройства, и поэтому для того, чтобы его защитить, необходима антивирусная защита.

Вирус – это определенная программа или код, который копирует себя в Windows, в результате чего происходят изменения в реестре системы. Эти изменения могут привести не только к неправильной работе каких-либо приложений, но и всей операционной системы.

Какой вред могут нанести вирусы компьютеру? На этот вопрос ответить достаточно сложно, потому что каждый день появляется от 10 до 100 новых вирусов, которые существенно отличаются друг от друга. Но одно можно сказать с уверенностью: вирус дублирует себя в системные файлы или заражает другие программы, в итоге происходит потеря данных.

Как же понять, что компьютер оказался зараженным? Это помогают определить различные признаки:

1. Некоторые программы перестают работать или работают с ошибками.
2. Размер некоторых исполнимых файлов и время их создания изменяются. В первую очередь это происходит с командным процессором, его размер увеличивается на величину размера вируса.
3. На экран выводятся посторонние символы и сообщения, появляются странные видео и звуковые эффекты.
4. Работа компьютера замедляется, и уменьшается размер свободной оперативной памяти.
5. Некоторые файлы и диски оказываются испорченными (иногда необратимо, если вирус отформатирует диск).
6. Компьютер перестает загружаться с жесткого диска.

Всегда нужно обращать внимание на ошибки и отклонения от работы каких-либо программ или действий, исполняемых компьютером.

Существует огромное множество вирусов, которые могут быть как неопасными, так и очень опасными, они могут привести к потере программ, уничтожению данных, стиранию информации в системных областях диска без возможности восстановления.

Стоит отметить, что путей проникновения вирусов на электронные устройства множество, не стоит думать, что если человек не пользуется интернетом, то вирусная программа не может проникнуть на его компьютер, ноутбук. Вирус может попасть через:

- глобальную сеть Internet;
- электронную почту;
- локальную сеть;
- компьютеры «Общего назначения»;
- пиратское программное обеспечение;
- ремонтные службы;

– съемные накопители, на которых находятся заражённые вирусом файлы;

– жёсткий диск, на который попал вирус;

– вирус, оставшийся в оперативной памяти после предшествующего пользователя.

Как можно защитить свой компьютер от вирусов? Уменьшить ущерб, а также снизить риск заражения вирусом, поможет антивирусная программа и действия, которые нужно стараться выполнять, чтобы сохранить устройство и не утратить информацию, находящуюся на нем. В повседневной работе на компьютере необходимо пользоваться следующими рекомендациями.

1. Установите на свой персональный компьютер современную антивирусную программу.

2. Перед просмотром информации, принесенной на флэш-карте с другого компьютера, проверьте носитель антивирусом.

3. После разархивирования архивных файлов сразу проверьте их на вирусы (не все антивирусные программы могут искать вредоносный код в архивах или могут делать это не корректно).

4. Периодически проверяйте компьютер на вирусы (если активно пользуетесь Интернетом – запускайте раз в неделю, а то и чаще).

5. Как можно чаще делайте резервные копии важной информации (backup).

6. Используйте совместно с антивирусной программой файервол (firewall), если компьютер подключен к Интернет.

7. Настройте браузер (программа просмотра Интернет-страниц – InternetExplorer, Opera, Chrome и другие) для запрета запуска активного содержимого html-страниц.

На сегодняшний день существует много различных антивирусов, например NOD 32, Dr.Web, антивирус Касперского, Norton Antivirus и другие, которые могут помочь противодействовать вирусам. Хотелось бы обратить внимание, что не стоит устанавливать несколько антивирусов одновременно на один компьютер, потому что это значительно снизит производительность работы антивирусов, которые будут заняты сканированием друг друга.

Хотя сегодня в Интернете можно найти бесплатные антивирусные программы, было бы лучше и надёжнее приобрести полную версию антивирусного программного обеспечения, чтобы обеспечить себе полную защиту.

Действия каждого антивируса заключаются в следующем:

1. Найти и удалить инфицированный файл.

2. Заблокировать доступ к инфицированному файлу.

3. Отправить файл в карантин (то есть не допустить дальнейшего распространения вируса).

4. Попытаться «вылечить» файл, удалив вирус из тела файла.

5. В случае невозможности лечения-удаления, выполнить эту процедуру при следующей перезагрузке операционной системы.

Для того чтобы антивирусная программа постоянно успешно работала, необходимо базу сигнатур вирусов периодически загружать, обычно это делается через сеть Интернет.

С помощью антивируса можно вылечить заражённые файлы, но ведь легче предупредить вирус, чем лечить уже инфицированные программы или данные. И стоит отметить, что хорошая антивирусная защита – это не только хорошая антивирусная программа, но и грамотное поведение пользователя в сети и за своим компьютером.

Таким образом, современные условия функционирования технических устройств предъявляют повышенные требования к антивирусной защите, и, значит, каждый пользователь должен стараться в полной мере обеспечивать надежную безопасность своему устройству.

Библиографический список

1. http://www.allpchelp.ru/poleznoe/dly_chego_nyjnu_antivirysu/
2. http://online-academy.ru/antivir/
3. https://ru.wikipedia.org/wiki

БЕЗОПАСНОСТЬ WEB-ПРИЛОЖЕНИЙ

Е. В. Митюшкина, Н. К. Байгужина

Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

Современный мир несет в себе тысячи угроз и потенциальных опасностей буквально на каждом шагу и в каждый момент времени. Всемирная сеть, ставшая неотъемлемой частью нашей жизни, не является исключением.

Количество угроз растет пропорционально росту бизнеса, однако, как показала многолетняя практика, 99% атак происходят через десяток стандартных ошибок валидации входящих данных, либо обнаруженные уязвимости в установленных компонентах программного обеспечения сторонних производителей, либо банально, по халатности системных администраторов, использующих настройки и пароли, установленные по умолчанию. Классификацией векторов атак и уязвимостей занимается сообщество OWASP (Open Web Application Security Project). Это международная некоммерческая организация, сосредоточенная на анализе и улучшении безопасности программного обеспечения.

OWASP создал список из десяти самых опасных векторов атак на Web-приложения, этот список получил название OWASP TOP-10, и в нем сосредоточены самые опасные уязвимости, которые могут стоить некоторым людям больших денег или подрыва деловой репутации, вплоть до потери бизнеса.

1. Инъекции

Все данные, как правило, хранятся в специальных базах, обращения к которым строятся в виде запросов, чаще всего написанных на специальном языке

запросов SQL (Structured Query Language – структурированный язык запросов). Приложения используют SQL-запросы для того, чтобы получать, добавлять, изменять или удалять данные. При недостаточной проверке данных от пользователя, злоумышленник может внедрить в форму web-интерфейса приложения специальный код, содержащий кусок SQL-запроса.

Такой вид атаки называется инъекция, в данном случае самый распространённый – SQL-инъекция. Это опаснейшая уязвимость, позволяющая злоумышленнику получить доступ к базе данных и возможность читать, изменять, удалять информацию, которая для него не предназначена.

2. Недочеты системы аутентификации и хранения сессий

Для того чтобы отличать одного пользователя от другого, web-приложение использует так называемые сессионные куки. После того, как будет введен логин и пароль и пройдет авторизация пользователя, в хранилище браузера сохраняется специальный идентификатор, который браузер в дальнейшем предъявляет серверу при каждом запросе страницы web-приложения.

В случае, если идентификатор был украден злоумышленником, а в системе не были реализованы проверки, например IP-адреса сессии, или проверки наличия более одного соединения в одной сессии, злоумышленник сможет получить доступ в систему с правами вашего аккаунта.

3. Межсайтовый скриптинг

Межсайтовый скриптинг – еще одна ошибка валидации пользовательских данных, которая позволяет передать JavaScript код на исполнение в браузер пользователя. Атаки такого рода часто также называют HTML-инъекциями, ведь механизм их внедрения очень схож с SQL-инъекциями, но, в отличие от последних, внедряемый код исполняется в браузере пользователя.

4. Небезопасные прямые ссылки на объекты

Данный вид уязвимости является также следствием недостаточной проверки пользовательских данных. Суть ее заключается в том, что при выводе каких-либо конфиденциальных данных, например личных сообщений или учетных карточек клиентов, для доступа к объекту используется идентификатор, который передается в открытом виде в адресной строке браузера, и не реализована проверка прав доступа к объектам.

Эксплуатация данной уязвимости очень проста и не требует вообще никаких специальных навыков – достаточно лишь перебирать число в адресной строке браузера и наслаждаться результатом.

5. Небезопасная конфигурация

Безопасность web-приложения требует наличия безопасной конфигурации всех компонентов инфраструктуры: компонентов приложения, web-сервера, сервера баз данных и самой платформы. Настройки компонентов сервера по умолчанию зачастую небезопасны и открывают возможности к атакам.

При правильной настройке сервера и включенной опции cookie_httponly, получить сессионную cookie через JavaScript невозможно, но зачастую эта простая и важная настройка отсутствовала в таких критично важных местах, как личные кабинеты платежных систем.

Кроме того, программное обеспечение должно быть в актуальном состоянии: уязвимости находят каждый день в самых различных программных компонентах – операционной системе, web-серверах, серверах баз данных, почтовых серверах.

6. Незащищенность критичных данных

Многие web-приложения не защищают конфиденциальные данные, такие как кредитные карты и учетные данные для аутентификации. Злоумышленники могут украсть или модифицировать такие слабо защищенные данные для использования в своих корыстных целях.

7. Отсутствие функций контроля доступа

Суть уязвимости, как следует из названия, заключается в отсутствии проверки наличия надлежащего доступа к запрашиваемому объекту. Большинство web-приложений проверяют права доступа, прежде чем отобразить данные в пользовательском интерфейсе. Тем не менее, приложения должны выполнять те же проверки контроля доступа на сервере при запросе любой функции. Ведь есть еще множество вспомогательных служебных запросов, которые зачастую отправляются в фоновом режиме асинхронно, при помощи технологии AJAX.

Если параметры запроса недостаточно тщательно проверяются, злоумышленники смогут подделать запрос для доступа к данным без надлежащего разрешения.

8. Межсайтовая подделка запроса

Вектор атаки CSRF, также известный как XSRF, позволяет злоумышленнику выполнять от имени жертвы действия на сервере, где не реализованы дополнительные проверки.

9. Использование компонентов с известными уязвимостями

Зачастую web-приложения написаны с использованием специальных библиотек или «фреймворков», которые поставляются сторонними компаниями. В большинстве случаев эти компоненты имеют открытый исходный код, доступный для всех пользователей, которые проверяют их исходный код, в том числе и на предмет уязвимостей. И нужно отметить, что делают они это отнюдь не безуспешно.

Также уязвимости ищут в более низкоуровневых компонентах системы, таких как сервер базы данных, web-сервер и, наконец, компоненты операционной системы – вплоть до ее ядра.

Очень важно использовать последние версии компонентов и следить за появляющимися известными уязвимостями на сайтах типа securityfocus.com.

10. Непроверенные переадресации и пересылки

Web-приложения зачастую переадресуют пользователя с одной страницы на другую. В процессе могут использоваться ненадлежащим образом проверяемые параметры с указанием страницы конечного назначения переадресации. Без соответствующих проверок атакующий может использовать такие страницы для переадресации жертвы на подложный сайт, который, к примеру, может

иметь очень схожий или неотличимый интерфейс, но украдет ваши данные кредитной карты или другие критичные конфиденциальные данные.

Этот вид уязвимостей, также как и многие другие перечисленные выше, является разновидностью ошибок проверки входящих данных.

ПРОГРАММНЫЕ АГЕНТЫ И МУЛЬТИАГЕНТНЫЕ СИСТЕМЫ

Ж. В. Михайличенко
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск,

Агент (лат. agere – вести, действовать) – это аппаратная или программная сущность, способная действовать в интересах достижения целей, поставленных перед ним владельцем или пользователем.

Фактически агенты – это программные роботы, предназначенные для работы с информацией. Идея интеллектуальных помощников при общении пользователя с компьютером родилась в середине 70-х годов и была частично воплощена во многих популярных продуктах фирм Microsoft, Apple, Lotus Software и других. Но настоящее развитие программных агентов началось с развитием Интернета. Говорящие агенты могут общаться в виртуальных мирах, информационные агенты доставляют новости и сообщают об изменениях на избранных сайтах, магазинные агенты сравнивают цены на товары в электронных магазинах, агенты-пауки (кроулеры) перемещаются по сети и индексируют информацию для поисковых серверов.

Целесообразное поведение агентов появляется только на интеллектуальном уровне, так как в этом случае необходимо не только наличие целей функционирования, но и возможность использования достаточно сложных знаний о среде, партнёрах и о себе. Отличительная черта таких программных агентов – стремление как можно лучше понять, что от них требуется. Они наблюдают за поведением пользователя, стараясь уловить закономерности и предложить свои услуги для выполнения каких-либо рутинных операций. Достаточно часто компонентом таких агентов являются искусственные нейронные сети, способные обучаться.

Для интеллектуальных агентов характерны следующие свойства:

1) Автономность – способность функционировать без вмешательства со стороны своего владельца и осуществлять контроль внутреннего состояния своих действий.

2) Социальное поведение – возможность взаимодействия и коммуникации с другими агентами.

3) Реактивность – адекватное восприятие среды и соответствующие реакции на её изменение.

4) Активность – способность генерировать цели и действовать рациональным образом для их достижения.

5) Базовые знания – знания агента о себе, окружающей среде, включая других агентов, которые не меняются в рамках жизненного цикла агента.

6) Убеждения – переменная часть базовых знаний, которые могут меняться во времени, хотя агент может об этом и не знать и продолжать их использовать для своих целей.

7) Цели – совокупность состояний, на достижение которых направлено текущее состояние агента.

8) Желания – состояния или ситуации, достижение которых для агента важно.

9) Обязательства – задачи, которые берёт на себя агент по просьбе или поручению других агентов.

10) Намерения – то, что агент должен делать в силу своих обязательств и желаний.

Иногда в этот перечень добавляют и такие свойства, как рациональность, правдивость, благожелательность, мобильность, хотя эти свойства характерны не только для интеллектуальных агентов.

Мультиагентная система (МАС) – это программно-вычислительный комплекс, где взаимодействуют различные агенты для решения задач, которые трудны или недоступны в силу своей сложности для одного агента.

В зависимости от концепции, выбранной для организации МАС, обычно выделяют три класса архитектур: интеллектуальная, реактивная и гибридная.

Организация МАС на принципах искусственного интеллекта имеет преимущества с точки зрения удобства использования методов и средств символьного представления знаний. Однако некоторые свойства агентов (желания, намерения, обязательства по отношению к другим агентам) невозможно выразить в терминах исчисления предикатов первого порядка. Поэтому для представления знаний агентов в рамках данной архитектуры были использованы специальные расширения соответствующих логических исчислений. Недостатком интеллектуальной архитектуры МАС является существенная трудность в создании точной и полной модели представления мира, процессов и механизмов рассуждений.

Принципы реактивной архитектуры возникли как альтернативный подход к архитектуре интеллектуальных агентов. Простым примером реализации реактивной архитектуры можно считать системы, где реакции агентов на внешние события генерируются конечными автоматами. Реактивный подход позволяет наилучшим образом использовать множество достаточно простых образцов поведения для реакции агента на определённые стимулы для конкретной предметной области. Однако применение этого подхода ограничивается необходимостью полного ситуативного анализа всех возможных активностей агентов.

Появление гибридных архитектур обусловлено невозможностью создания оптимальных МАС на основе первых двух подходов. Поэтому в последнее время прослеживается тенденция разработки и использования именно гибридных МАС-архитектур и систем агентов. Но такие МАС обычно слишком специфичны для приложений, под которые они разрабатываются.

Современная теория МАС используется в составных системах обороны, в транспорте, логистике, графике, геоинформационных системах и многих других. Многоагентные системы хорошо зарекомендовали себя в сфере сетевых и мобильных технологий, для обеспечения автоматического и динамического баланса нагруженности, расширяемости и способности к самовосстановлению.

СОВРЕМЕННЫЕ СРЕДСТВА КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ И РАСЧЕТА

А. Д. Михайлов, С. Н. Сергиенко, В. А. Твердохлебов, И. В. Мишуков
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

Современные тенденции перехода на новое качество проектирования изделий и технологий, возникшие в последнее время, приводят к появлению нового интегрированного конструкторско-технологического уклада, основанного на интеграции в промышленности всех этапов работ и реализации концепции информационной поддержки жизненного цикла изделий. Предпосылкой для реализации концепции явился переход в процессе проектирования от бумажной и двухмерной электронной документации к твердотельному 3D-моделированию изделий и их компонентов в CAD-системах (средах).

В Орском гуманитарно-технологическом институте (филиале) Оренбургского государственного университета были созданы условия для эффективного внедрения IT-технологий в учебный процесс. Основой ее явился программный комплекс конструкторско-технологического проектирования (программные продукты фирмы АСКОН – КОМПАС и Вертикаль).

Создание трехмерных моделей позволяет визуализировать объект, оценить собираемость изделия, корректность, размерные цепи и другую информацию, которая при 2D-проектировании не могла быть получена.

Важнейшим этапом подготовки специалиста является его технологическая подготовка. Технологический процесс является сложным многофакторным объектом. Проблемы по выбору заготовки, размерному анализу деталей и автоматизации технологического проектирования – вот небольшой перечень вопросов, которые приходиться решать. Справиться с этим помогают программы «Вертикаль», «КОМПАС» и «Моделирование процесса технологической обработки тел вращения». Если первая и вторая имеют широкое применение как в учебном процессе, так и на предприятиях машиностроения, то последняя еще мало известна.

Разработанная методика моделирования проектирует заготовки, припуски и межоперационные размеры, режимы резания и нормирование. Учитываются особенности структурных характеристик, прогнозы будущей трудоемкости и стоимости изготовления деталей.

Прикладная программа «Моделирование процесса технологической обработки тел вращения» предназначена также для усвоения материала основных

разделов дисциплины «Основы технологии производства и ремонта автомобилей», закрепления базовых знаний и снижения до минимума механических расчетов.

Она включает следующие разделы:

– табличный метод определения заготовки;

– подробный расчет припусков и межоперационных размеров;

– окончательный вариант получения заготовки;

– расчет режимов резания с использованием обширной библиотеки;

– нормирование операций.

Внедренная имитационно-моделирующая программа обеспечивает представление реальных процессов и решений, состоит из учебной информации и программы управления процессами знаний в диалоге с компьютером, обеспечивает взаимосвязь материала курса, подлежащего компьютеризации, с материалами, изучаемыми традиционными способами

Окна программы представлены на рисунках 1, 2.

Рассмотрим пример выбора заготовки.

Перед тем, как приступить к расчетам заготовки и последующих припусков и допусков, нужно перенести деталь с чертежа в программу (чертеж или модель выполняют в программе КОМПАС) и завести ее геометрические характеристики (рис. 1) в соответствующие поля. Первым видом расчета является выбор метода получения заготовки, который позволяет определить оптимальную заготовку как с точки зрения экономии металла, так и ее себестоимости. Рассчитанные методы можно объединить путем выбора сравнительного пункта в списке рекомендуемых методов получения и, если требуется, экспортировать в «Microsoft Word». Далее – расчет припусков, допусков и межоперационных размеров, который также экспортируются в «Microsoft Word».

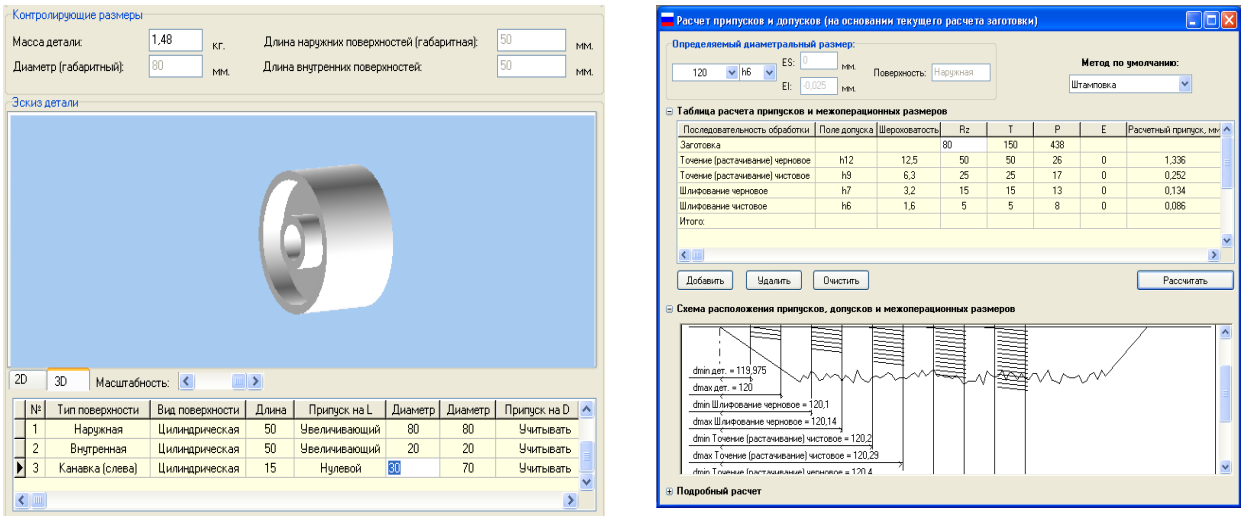

Рис. 1. Расчет заготовки

Следующим видом расчета является расчет режимов резания и норм времени (рис. 2).

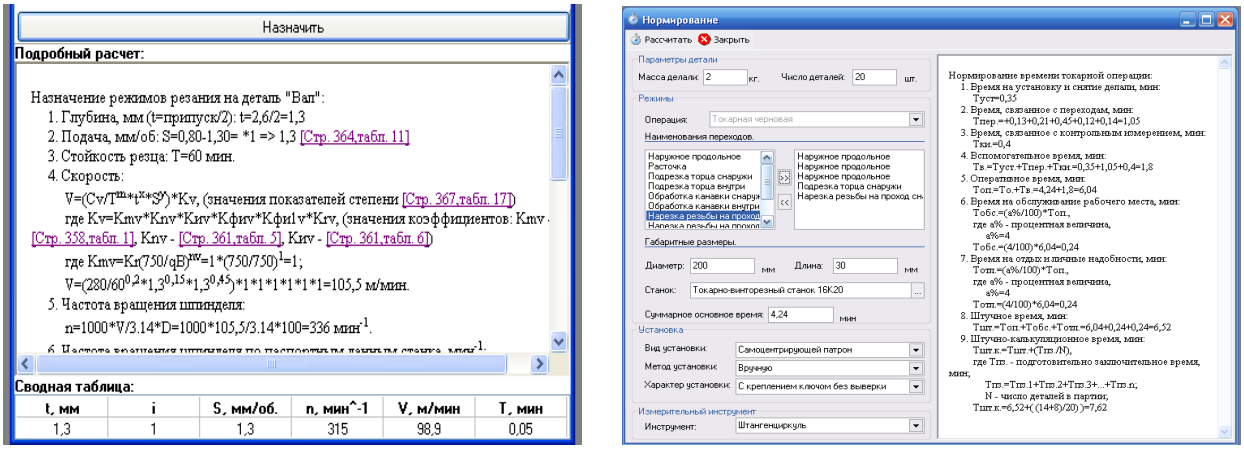

Рис 2. Пример расчета режимов резания и норм времени

Надо отметить, что эти виды вычислений традиционным методом самые объемные, однако применение прикладной программы «Моделирование процесса технологической обработки тел вращения» значительно облегчает работу как по времени, так и по оформлению. Экспортирование расчетов в «Microsoft Word» дает возможность отображения ссылок на использованные справочники (с постраничной нумерацией).

В завершение надо отметить, что использование данного программного продукта позволяет:

– визуализировать работу за счет использования геометрических моделей и приложений по оборудованию, режущему инструменту и оснастке;

– экономить время при работе со справочниками;

– сократить время на оформление курсовых, расчетно-графических или дипломных работ.

КОМПЛЕКСНАЯ ЗАЩИТА ИНФОРМАЦИИ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

Е. В. Москалёв, А. А. Елисеев
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

Появившиеся в начале 80-х годов персональные ЭВМ прочно вошли во все сферы человеческой деятельности. Вместе с ними у эксплуатирующих ЭВМ организаций и ведомств возникли и многочисленные проблемы. Одна из них – защита информации.

Проблема защиты информации представляет собой совокупность тесно связанных проблем в областях права, организации управления, разработки технических средств, программирования и математики. Одна из центральных за-

дач проектирования систем защиты состоит в рациональном распределении имеющихся ресурсов. Характерная особенность использования ЭВМ заключается в том, что доступ к ним имеют многие пользователи. В связи с таким режимом работы возникает целый набор взаимосвязанных вопросов по защите информации, хранящейся в ЭВМ. В коммерческих и военных областях одной из основных является проблема защиты информации. Так, можно выделить следующие объективные причины, определяющие важность проблемы защиты информации:

- высокие темпы роста парка ЭВМ, находящихся в эксплуатации;
- широкое применение ЭВМ в самых различных сферах;
- высокая степень концентрации информации в ЭВМ;
- совершенствование доступа пользователей к ресурсам ЭВМ;
- усложнение вычислительного процесса в ЭВМ.

Усложнение методов и средств организации машинной обработки информации приводят к тому, что информация становится все более уязвимой. Этому способствуют такие факторы, как постоянно возрастающие объемы обрабатываемых данных, накопление и хранение данных в ограниченных местах, постоянное расширение круга пользователей, имеющих доступ как к ресурсам ЭВМ, так и к программам и данным, хранящимся в них, усложнение режимов эксплуатации вычислительных систем.

Одним из самых распространённых видов преступлений является несанкционированный доступ к информации. Рассмотрим, что представляет собой НСД и какие его методы являются самыми распространёнными в наше время. Документ Гостехкомиссии «Защита от НСД. Термины и определения» трактует термин следующим образом: «Несанкционированный доступ к информации – доступ к информации, нарушающий правила разграничения доступа с использованием штатных средств, предоставляемых средствами вычислительной техники или автоматизированными системами».

Основные причины несанкционированного доступа: ошибки конфигурации, слабая защищённость средств авторизации, ошибки в программном обеспечении, злоупотребление служебными полномочиями, использование клавиатурных шпионов, вирусов и троянов на компьютерах сотрудников.

С прогрессивным развитием технологий обработки информации большое распространение получили следующие методы несанкционированного доступа к информации:

– подключение к линиям связи и вживление в компьютерную систему с использованием интервалов времени в действиях законного пользователя;

– несанкционированное использование компьютерной системы в своих личных целях или блокирование системы для отказа в обслуживании другим пользователям. Для реализации используются программные средства, которые могут захватывать определенные ресурсы системы;

– повторное использование удаленных объектов системы. Примером является удаление файлов операционной системой. ОС сообщает, что определённый файл удален, однако информация, которая содержится в данном файле, не

обязательно удалена. Она никуда не исчезает до момента записи в это место другой информации. Этот процесс называется работой с компьютерным «мусором»;

– нарушитель подключается к линиям связи и имитирует работу системы с целью осуществления незаконных манипуляций. Например, он может имитировать сеанс связи и получить данные под видом легального пользователя;

– захватчик анализирует поведение пользователей в системе;

– программы для анализа средств защиты от НСД и их обхода.

Развитие технологий связи и электронной почты выделило злоупотребление, которое в литературе получило название «pinging». Суть данного злоупотребления заключается в том, что, используя стандартные или специально разработанные программные средства, злоумышленник может вывести из строя электронный адрес, атакуя его большим количеством сообщений.

Существует множество путей реализации воздействий, которые считаются опасными для информационной системы. По характеру возникновения их можно разделить на преднамеренные и непреднамеренные.

Непреднамеренные угрозы – это случайные действия, выраженные в неадекватной поддержке механизмов защиты или ошибками в управлении. А преднамеренные – это несанкционированное и незаконное получение информации и несанкционированная манипуляция данными или ресурсами.

Также угрозы можно разделить на программные и непрограммные. К программным относят те, которые реализованы в виде отдельного программного модуля или модуля в составе программного обеспечения. К непрограммным относят злоупотребления, в основе которых лежит использование технических средств для подготовки и реализации компьютерных преступлений.

Преследуя различные цели, киберпреступники используют богатый набор программных средств, которые разделяют на тактические и стратегические. К тактическим относят средства, которые преследуют достижение цели – используются для подготовки и реализации стратегических средств, которые связаны с большими финансовыми потерями для ИС. К группе стратегических относятся средства, реализация которых обеспечивает возможность получения контроля над технологическими операциями функционирования ИС.

Потенциальными программными злоупотреблениями можно считать программные средства, которые обладают следующими функциональными возможностями:

– искажение произвольным образом, блокирование каналов связи;

– скрытие следов присутствия в программной среде ЭВМ;

– уничтожение кодов программ в оперативной памяти;

– сохранение информации из оперативной памяти в областях внешней памяти.

Разобрав понятие НСД, можно перейти к системам защиты, но для начала нужно рассмотреть проблемы создания систем защиты информации. Всего две взаимодополняющие задачи: разработка системы защиты информации и оценка системы защиты информации.

Вторая задача решается путем анализа ее технических характеристик и, далее, сертификацией средств защиты информации и аттестацией системы защиты информации в процессе ее внедрения.

Методы и средства обеспечения безопасности информации:

Препятствия – физические преграждения пути злоумышленнику к защищаемой информации.

Управление доступом – регулирование использования всех ресурсов компьютерной информационной системы. Управление доступом включает следующие функции защиты:

– идентификацию пользователей;

– опознание объекта по идентификатору;

– проверку полномочий;

– регистрацию обращений к защищаемым ресурсам;

– регистрацию при попытках несанкционированных действий.

- Маскировка – криптографическое закрытие информации.
- Принуждение – метод защиты, при котором пользователи системы вынуждены соблюдать правила работы с защищаемой информацией под угрозой материальной, административной или уголовной ответственности.

Такие методы обеспечения безопасности основаны на применении технических, программных, организационных, законодательных и морально-этических средств защиты. К основным средствам защиты можно отнести технические средства, которые реализуются в виде электрических, электромеханических и электронных устройств, которые, в свою очередь, разделяются на аппаратные и физические.

Программные средства представляют собой программное обеспечение, специально предназначенное для выполнения функций защиты информации: механизм шифрования, механизм цифровой подписи, механизмы контроля доступа, архивация, защита при вводе и выводе информации.

Организационные средства защиты представляют собой организационно-технические и организационно-правовые мероприятия, осуществляемые в процессе создания и эксплуатации вычислительной техники, аппаратуры телекоммуникаций для обеспечения защиты информации. Организационные мероприятия охватывают все структурные элементы аппаратуры на всех этапах их жизненного цикла.

Законодательные средства защиты определяются законодательными актами страны, которыми регламентируются правила пользования, обработки и передачи информации и устанавливаются меры ответственности за нарушение этих правил.

В настоящее время наиболее острую проблему безопасности составляют вирусы, которые успешно внедрились в повседневную компьютерную жизнь. Компьютерный вирус – это небольшая по размерам программа, которая может «заражать» другие программы, а также выполнять различные незаконные действия на компьютере. Основным средством защиты от вирусов служит использование антивирусных программ.

В профилактических целях для защиты от вирусов необходимо:

– работа с носителями информации, защищенными от записи;

– разделение новых программ и установленных ранее программ;

– проверка новых программ на наличие вирусов;

– хранение программ на жестком диске в архивированном виде.

На данный момент с развитием технологий существует большое количество угроз, направленных на несанкционированный доступ к информации, на ее искажение, удаление. Следовательно, при организации защиты информации система защиты должна быть надежной, эффективной и управляемой. Границы безопасности должны быть разумными, а затраты – на том уровне, который обеспечит поддержание системы в работоспособном состоянии на протяжении длительного периода времени.

Эффективность защиты информации достигается способностью ее адекватно реагировать на все попытки несанкционированного доступа к информации, а мероприятия по защите информации от несанкционированного доступа должны носить комплексный характер.

ТЕХНОЛОГИИ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ СИСТЕМ В ПРОМЫШЛЕННОСТИ

О. В. Пергунова

Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

Основное содержание технологии проектирования составляют технологические инструкции, состоящие из описания последовательности технологических операций, условий, в зависимости от которых выполняется та или иная операция, описаний самих операций.

Выделяют следующие общие требования, которым должны удовлетворять технологии проектирования, разработки и сопровождения информационных систем:

– поддерживать полный жизненный цикл информационной системы;

– обеспечивать гарантированное достижение целей разработки системы с заданным качеством и в установленное время;

– обеспечивать возможность разделения крупных проектов на ряд подсистем – делить композицию проекта на составные части, разрабатываемые группами исполнителей ограниченной численности, с последующей интеграцией составных частей;

– технология должна обеспечивать возможность ведения работ по проектированию отдельных подсистем небольшими группами (3-7 человек);

– обеспечивать минимальное время получения работоспособной системы;

– предусматривать возможность управления конфигурацией проекта, ведения версий проекта и его составляющих, возможность автоматического выпуска проектной документации и синхронизацию ее версий с версиями проекта;

– обеспечивать независимость выполняемых проектных решений от средств реализации системы – системы управления базами данных, операционной системы, языка и системы программирования.

Технологии характеризуются в двух измерениях: вертикальном (представляющем процессы) и горизонтальном (представляющем стадии).

Процесс – совокупность взаимосвязанных действий, преобразующих некоторые входные данные в выходные. Процессы состоят из набора действий, а каждое действие – из набора задач. Вертикальное измерение отражает статические аспекты процессов и оперирует такими понятиями, как рабочие процессы, действия, задачи, результаты деятельности и исполнители.

Стадия – часть действий по созданию программного обеспечения, ограниченная некоторыми временными рамками и заканчивающаяся выпуском конкретного продукта, определяемого заданными для данной стадии требованиями. Стадии состоят из этапов, которые обычно имеют итерационный характер. Иногда стадии объединяют в более крупные временные рамки, называемые фазами. Горизонтальное измерение представляет собой время, отражает динамические аспекты процессов и оперирует такими понятиями, как фазы, стадии, этапы, итерации и контрольные точки.

Специфика комбинаций стадий и процессов, ориентированная на разные классы программного обеспечения и на особенности коллектива разработчиков, определяет технологический подход.

Существуют несколько технологических подходов.

Подходы со слабой формализацией не используют явных технологий, и их можно применять только для очень маленьких проектов, как правило завершающихся созданием демонстрационного прототипа. К подходам со слабой формализацией относятся так называемые ранние технологические подходы, например подход «кодирование и исправление».

Строгие (классические, жесткие, предсказуемые) подходы применяют для средних, крупномасштабных и гигантских проектов с относительно ясными требованиями к системе и более-менее фиксированным объемом работ. Одно из основных требований к таким проектам – как можно большая предсказуемость. В эту группу входят следующие подходы:

1) Каскадные технологические подходы:

а) классический каскадный подход – переход к следующему процессу осуществляется только после того, как завершена работа с текущим процессом. Возвраты к уже пройденным процессам не предусмотрены;

б) каскадно-возвратный подход – разрешены возвраты к предыдущим стадиям и пересмотр или уточнение ранее принятых решений;

в) каскадно-итерационный подход – предусматривает последовательные итерации каждого процесса до тех пор, пока не будет достигнут желаемый результат;

г) каскадный подход с перекрывающимися процессами – предполагает наличие специализированных команд, позволяющих сократить передаваемую

документацию. Следующий процесс начинается до завершения текущего. Несколько процессов могут выполняться параллельно;

д) каскадный подход с подпроцессами – очень близок подходу с перекрывающимися процессами. Особенность в том, что проект может быть разделен на подпроекты, которые могут разрабатываться индивидуально.

2) Спиральная модель использует понятие прототипа, то есть программы, реализующей частичную функциональность создаваемого программного продукта. Особенность модели – в разработке итерациями, причем каждый следующий итерационный прототип будет обладать большей функциональностью.

3) Каркасные подходы. Рациональный унифицированный процесс вобрал в себя лучшее из технологических подходов каскадной группы. Включает в себя следующие фазы: начало (определение целей проекта), исследование (разработка плана и архитектуры проекта), построение (постепенное создание системы), внедрение (поставка системы конечным пользователям). Особенности – итеративность, контроль качества, возможность выявить ошибки на ранних стадиях, предпочтение отдается моделям, а не бумажным документам, конфигурирование, настройка, масштабирование.

4) Формальные подходы предусматривают особые формальные требования к процессу создания программного обеспечения. Например, для генетических подходов требуются формальности, связанные с происхождением программы и дисциплиной ее создания.

5) Синтезирующее программирование предполагает синтез программы по ее спецификации. Документ на языке спецификаций является базисом для последующей реализации.

6) Сборочное (расширяемое) программирование предполагает, что программа собирается путем переиспользования уже известных фрагментов.

7) Подходы на основе формальных преобразований:

а) формальное синтезирующее программирование использует математическую спецификацию – совокупность логических формул;

б) формальное сборочное программирование использует спецификацию как композицию уже известных фрагментов;

в) формальное конкретизирующее программирование использует смешанные вычисления и конкретизацию по аннотациям.

8) Гибкие (адаптивные, легкие) подходы применяют для небольших или средних проектов в случае неясных или изменяющихся требований к системе. Команда разработчиков должна быть ответственной и квалифицированной, заказчики должны быть согласны принимать участие в разработке.

9) Ранние технологические подходы быстрой разработки:

а) эволюционное прототипирование – первый прототип включает создание развитого пользовательского интерфейса;

б) итеративная разработка – первый прототип уже должен включать завершенное ядро системы;

в) постадийная разработка – должна решить недостаток первых двух подходов – невозможность определения сроков завершения проектов.

10) Адаптивные подходы:

а) экстремальное программирование (eXtreme Programming или XP). Тщательное предварительное проектирование ПО заменяется, с одной стороны, постоянным присутствием в команде заказчика, готового ответить на любой вопрос и оценить любой прототип, а с другой стороны, регулярными переработками кода. Основой проектной документации считается тщательно прокомментированный код. Большое внимание в методологии уделяется тестированию. Как правило, для каждого нового метода сначала пишется тест, а потом уже разрабатывается собственно код метода до тех пор, пока тест не начнет выполняться успешно. Эти тесты сохраняются в наборах, которые автоматически выполняются после любого изменения кода;

б) адаптивная разработка. В основе лежат три стадии – обдумывание, сотрудничество и обучение. Результаты планирования в данном подходе всегда не предсказуемы. В отличие от обычного планирования, отклонения в котором ведут к ошибкам, здесь отклонения ведут к правильным решениям. Обязательства и планы программистов и заказчиков пересматриваются в течение всего процесса разработки.

11) Подходы исследовательского программирования:

а) компьютерный дарвинизм основан на принципе восходящей разработки, когда система строится вокруг ключевых компонентов и программ, которые создаются на ранних стадиях разработки. Представляет собой метод проб и ошибок, основанный на интенсивном тестировании, причем на любом этапе система должна работать;

б) фрагментарное программирование состоит в том, что сначала создается шаблон программ с работающими кусочками (фрагментами). Далее выполняется постепенное приближение к конечной цели. Применяется в том случае, когда не могут быть точно сформулированы требования к большей части задачи.

ОБЛАЧНЫЕ ТЕХНОЛОГИИ ХРАНЕНИЯ ДАННЫХ

Д. Н. Саргсян
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

Последние достижения в области технологий, включая невероятное распространение Интернета и веб-приложений, создали убедительные аргументы в бизнесе для многих организаций, достаточно серьезно рассмотреть вопрос о переносе своих ИТ-операций от внутренних серверов в *облачное хранение данных***,** что касается и использования веб-приложений.

Облачное хранилище данных – самая популярная облачная технология, которая заключается в размещении пользователем данных удаленно, а именно – в облаке, на удаленном сервере, принадлежащему третьей стороне – облачному провайдеру, с возможностью доступа к информации при помощи различных устройств, имеющих возможность доступа к Интернету.

Каждый пользователь знаком с проблемой транспортировки файлов с рабочего места на домашний персональный компьютер.

Например, пользователь работает с файлом электронной таблицы на рабочем месте и не успевает доделать свою работу, результат которой нужно будет показать руководству уже утром следующего дня. Какие стандартные действия можно предпринять? Можно скопировать этот файл на флеш-накопитель.

Гораздо быстрее и комфортнее будет воспользоваться облачным хранилищем, к которому пользователь будет иметь доступ не только с одного-двух рабочих мест, а с любого устройства, будь то планшет или смартфон. Облако позволит избежать проблем параллельной работы пользователей с одним информационным пространством и даст возможность с комфортом проделать коллективную работу.

Да, облачное хранилище гораздо безопаснее, чем флеш-накопитель. Дело в том, что облачные технологии – прерогатива больших компаний, например, таких как Amazon, Apple, Dropbox, Google, Microsoft, Яндекс. Эти компании к вопросу безопасности данных относятся с особой щепетильностью, и пока не замечено ни одного случая утечки важных данных.

Удаленное хранилище – это лучшее решение проблем для студентов. Все, кто учился или еще учится, довольно часто сталкиваются с проблемой обмена файлами в учебных целях.

Например, студенты часто делятся друг с другом лабораторными работами, практическими работами и домашним заданием, уже не говоря про коллективную учебу во время написания курсовых и дипломных работ и проектов. Каким образом тут могут пригодиться облачные хранилища данных? Все очень просто, студенты, как правило, обмениваются файлами или при помощи внешних носителей, или средствами электронной почты. При использовании флеш-накопителей велика вероятность заражения вирусами сначала накопителя, а затем и персонального компьютера.

При использовании электронной почты возникают проблемы, связанные с нюансами отправки писем, начиная с неправильно написанного электронного адреса, заканчивая техническими проблемами почтовых серверов, не говоря уже про банальную потерю письма в контакт-листе.

Облачное хранилище данных решит подобные проблемы в удобной для тебя форме, предоставляя коллективный доступ к файлу и к его редактированию в облаке. Хранилище данных предназначено для хранения файлов разного типа и большого размера.

К достоинствам облачного хранения данных относятся следующие моменты:

– облачные хранилища могут быть бесплатны, как правило, с предоставлением виртуального места объемом до 5-7 Гб;

– доступ к информации в облаке осуществим с любого устройства, поддерживающего интернет-подключение, из любой страны мира;

– возможен коллективный доступ к чтению и редактированию файлов;

– абсолютная защищенность от вирусов в данной облачной технологии;

– отсутствие возможности потери данных.

У облачного хранения данных есть свои недостатки:

– необходимо стабильное интернет-соединение;

– для использования большого объёма на виртуальном сервере необходимо оплачивать обслуживание;

– необходимо решать вопросы безопасного хранения данных для корпоративных информационных систем.

Таким образом, не следует всецело полагаться на облачные хранилища, они не безупречны с точки зрения защищенности данных, и выложенная в открытый доступ информация может попасть в чужие руки. Поэтому необходимо всегда держать резервные копии на локальных хранилищах, а важные документы отправлять в облачное хранение в зашифрованных архивах.

СУПЕРКОМПЬЮТЕРЫ

Д. А. Сороколетов
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

С момента появления первых компьютеров одной из основных проблем, стоящих перед разработчиками, была производительность вычислительной системы. За время развития компьютерной индустрии производительность процессора стремительно возрастала, однако появление все более изощренного программного обеспечения, рост числа пользователей и расширение сферы приложения вычислительных систем предъявляют новые требования к мощности используемой техники, что и привело к появлению суперкомпьютеров.

Суперкомпьютер – специализированная вычислительная машина, значительно превосходящая по своим техническим параметрам и скорости вычислений большинство существующих в мире компьютеров. Как правило, современные суперкомпьютеры представляют собой большое число высокопроизводительных серверных компьютеров, соединённых друг с другом локальной высокоскоростной магистралью для достижения максимальной производительности в рамках подхода распараллеливания вычислительной задачи.

Чаще всего авторство термина приписывается Джорджу Майклу и Сиднею Фернбачу, в конце 60-х годов XX века работавшим в Ливерморской национальной лаборатории, и компании CDC. В общеупотребительный лексикон термин «суперкомпьютер» вошёл благодаря распространённости компьютерных систем Сеймура Крэя, таких как CDC 6600, CDC 7600, Cray-1, Cray-2, Cray-3 и Cray-4. Сеймур Крэй разрабатывал вычислительные машины, которые, по сути, становились основными вычислительными средствами правительственных, промышленных и академических научно-технических проектов США с середины 60-х годов до 1996 года. Сам Крэй никогда не называл свои детища суперкомпьютерами, предпочитая использовать вместо этого обычное название «компьютер».

А зачем вообще нужны суперкомпьютеры? На этот вопрос дает ответ представитель компании «Крей рисерч» Вито Бонджорно, который отмечает, что раздвижение границ человеческого знания всегда опиралось на два краеугольных камня, которые не могут существовать друг без друга, – теорию и опыт. Однако теперь ученые сталкиваются с тем, что многие испытания стали практически невозможными – в некоторых случаях из-за своих масштабов, в других – дороговизны или опасности для здоровья и жизни людей. Тут-то и приходят на помощь мощные компьютеры. Позволяя экспериментировать с электронными моделями реальной действительности, они становятся «третьей опорой» современной науки и производства. Суперкомпьютеры используются во всех сферах, где для решения задачи применяется численное моделирование; там, где требуется огромный объём сложных вычислений, обработка большого количества данных в реальном времени или решение задачи может быть найдено простым перебором множества значений множества исходных параметров.

Первым суперкомпьютером стал Cray-1 – спроектированный Сеймуром Крэйем и созданный компанией Cray Research Inc. в 1976 году. Пиковая производительность машины – 133 Мфлопса.

Суперкомпьютер «Ломоносов» – первый гибридный суперкомпьютер в России. В нём используется 3 вида вычислительных узлов и процессоры с различной архитектурой. В качестве основных узлов, обеспечивающих свыше 90% производительности системы, используется blade-платформа T-Blade2. Предполагается использовать суперкомпьютер для решения ресурсоёмких вычислительных задач в рамках фундаментальных научных исследований, а также для проведения научной работы в области разработки алгоритмов и программного обеспечения для мощных вычислительных систем.

На данный момент самым мощным суперкомпьютером в мире является Tianhe-2 (КНР). Суперкомпьютер Tianhe-2, спроектированный компанией Inspur совместно с Оборонным научно-техническим университетом Народно-освободительной армии Китайской Народной Республики, был запущен в 2013 году. Его производительность – 33862.7 TFlop/s.

В заключение хотелось бы сказать, что еще 10-15 лет назад суперкомпьютеры были чем-то вроде элитарного штучного инструмента, доступного, в основном, ученым из засекреченных ядерных центров и криптоаналитикам спецслужб. Однако развитие аппаратных и программных средств сверхвысокой производительности позволило освоить промышленный выпуск этих машин, а число их пользователей в настоящее время достигает десятков тысяч.

ОРГАНИЗАЦИЯ ЗАПИСИ СИСТЕМЫ ДОМЕННЫХ ИМЁН

Е. Н. Стародубцев
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

DNS-сервер, (name server) – приложение, предназначенное для ответов на DNS– запросы по соответствующему протоколу. Также DNS-сервером могут называть хост, на котором запущено приложение. DNS была разработана Полом Мокапетрисом в 1983 году. DNS-серверы управляются американской некоммерческой организацией IANA.

Запись NS (name server) указывает на DNS-сервер для данного домена. Также для доменов ниже второго уровня можно добавлять DNS на партнерские сервера, например:

example.ru IN NS ns.example.ru.

Работать это будет при условии, что на серверах example.ru будет заведена зона для этих доменов.

Запись MX (mail exchange), или почтовый обменник, указывает сервер обмена почтой для данного домена, например:

example.microsoft.com. MX 10 mailserver1.example.microsoft.com

Цифра, указанная перед «mailserver1.example.microsoft.com», – это величина приоритета, меньшая цифра означает больший приоритет. Записи MX используются системой электронной почты для более эффективной маршрутизации почты. С помощью записей MX производится посылка почтовых сообщений не напрямую адресату, а на почтовый сервер на узле получателя. В приведенном выше примере почта будет приходить на сервер mailserver1.example.microsoft.com в первую очередь (приоритет 10 < 20), если этот сервер откажет – на mail.example.org.

Запись A (address record) – запись адреса связывает хост с адресом IP.

Основное назначение адресной записи – установить соответствие между доменным именем машины и IP-адресом. Собственно, это главная задача всей системы доменных имен. По этой причине адресная запись описания ресурса является одной из ключевых записей описания зоны. Например: host1.example.microsoft.com. IN A 127.0.0.1

Запись AAAA (address record для IPv6) – запись адреса связывает хост с адресом IPv6.

Запись AAAA является полным эквивалентом записи типа A, которая описывалась выше, с одним исключением: IPv6 адрес имеет другой вид. Например:

example.org IN A 2a03:4900:0:3::96:156

Запись CNAME (canonical name record), или каноническая запись имени, используется для перенаправления на другое имя.

CNAME обозначает каноническое имя или синоним существующего имени хоста, который должен иметь запись A.

Пример:

truename.example.microsoft.com

Обратите внимание: мы не заводим CNAME для доменов второго уровня. Только для поддоменов.

Запись TXT

Text (текст). Свободное текстовое поле, иногда заполняется специфичными для сайта дополнительными данными. Запись TXT используется для добавления произвольного текста к DNS-записям машины. Например, следующие записи определяют организацию:

IN TXT «University of CO, Boulder Campus, CS Dept»

IN TXT WP-PH://directory.colorado.edu/105

IN TXT WP-SMTP-EXPN-Finger://ns.cs.colorado.edu

В рамках TXT-записей используются так называемые SPF-записи (Sender Policy Framework), не дающие спаммерам рассылать письма от имени доменов, которые им не принадлежат. SPF позволяет владельцу домена указать в TXT-записи DNS-сервера специальным образом сформированную строку, указывающую список серверов, способных отправлять email сообщения от имени этого домена. Агенты передачи почты, получающие почтовые сообщения, могут запрашивать SPF-информацию с помощью простого DNS-запроса, верифицируя таким образом сервер отправителя.

Пример SPF данных в TXT-записи DNS:

example.org. IN TXT «v=spf1 a mx –all»

v= определяет используемую версию SPF. Далее следует перечисление механизмов верификации: в данном случае «a» и «mx» разрешает отправку писем для всех записей A и MX домена example.org. Строка завершается «-all» – указанием того, что сообщения, не прошедшие верификацию с использованием перечисленных механизмов, следует игнорировать.

Запись PTR (Pointer) – запись-указатель «обратной зоны».

Задача поиска доменного имени по IP-адресу является обратной к прямой задаче – поиску IP-адреса по доменному имени. Как было сказано выше, прямая задача решается в DNS при помощи записей типа A (Address). Обратная же задача решается при помощи записей-указателей типа PTR (Pointer), которые совместно с записями SOA и NS составляют описание так называемой «обратной» зоны. Решением «обратной» задачи занимается специальный домен, структура которого совпадает со структурой IP-адресов. Называется этот домен IN-ADDR.ARPA. Отметим, что PTR-записи мы не прописываем, ввиду невозможности данной операции на виртуальном хостинге.

SRV-записи (Server selection) местоположения серверов.

Указание на местоположение серверов для определенных сервисов, например, Jabber, Active Directory. Например:

_ldap._tcp._msdcs SRV 0 0 389 dc1.example.microsoft.com

SRV 10 0 389 dc2.example.microsoft.com

Запись IN-ADDR.ARPA.

В апреле 2000 года между DARPA (бывшей ARPA) и ICAAN было заключено соглашение о том, что домен верхнего уровня (TLD) ARPA будет использоваться для целей поддержки инфраструктуры Интернет. Кроме того, само слово «ARPA» следует расшифровывать как «Address and Routing Parameter Area Domain (ARPA)» и не следует его ассоциировать с сетью ARPANET. Основное назначение домена ARPA – обеспечивать отображение численных величин, определяемых протоколами межсетевого обмена, в пространство имен. Делегирование поддоменов в домене ARPA возложено на IAB (Internet Architecture Board). В настоящее время в ARPA выделено три поддомена:

in-addr.arpa для отображения IP-адресов IPv4 в пространство доменных имен;

ip6.arpa для отображения IP-адресов IPv6 в пространство доменных имен;

e164.arpa для отображения телефонных номеров формата E.164.

Сама зона ARPA поддерживается корневыми серверами и серверами TLD зон, хотя в соответствии с рекомендациями RFC 2870 для ARPA желательно выделение отдельных серверов. Имена в домене IN-ADDR.ARPA образуют иерархию цифр, которые соответствуют IP-адресам. Правда, записываются эти имена в обратном порядке относительно написания IP-адреса. Например, машина vega-gw.vega.ru, которая имеет адрес 194.226.43.1, должна быть описана в домене in-addr.arpa как 1.43.226.194.in-addr.arpa, то есть адрес записывается в обратном порядке. Так как речь идет о доменной адресации, то разбиение сети, на подсети в данном случае значения не имеет. Имена обрабатываются точно так же, как и обычные доменные имена.

BIND (Berkeley Internet Name Domain) – открытая и наиболее распространённая реализация DNS-сервера, обеспечивающая выполнение преобразования DNS-имени в IP-адрес и наоборот. BIND поддерживается организацией Internet Systems Consortium. 10 из 13 корневых серверов DNS работают на BIND, оставшиеся 3 работают на NSD. BIND был создан студентами в начале 1980-х на грант, выданный DARPA и впервые был выпущен в BSD 4.3. Ранние версии BIND хранили информацию только в текстовых файлах зон. Начиная с версии 9.4, в качестве хранилища можно использовать LDAP, Berkeley DB, PostgreSQL, MySQL и ODBC.

КОМПЬЮТЕРНЫЕ ВИРУСЫ И АНТИВИРУСНАЯ ЗАЩИТА

Н. В. Сулим

Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

Существует большое количество различных компьютерных вирусов. Одни из них могут просто заставлять двигаться курсор мыши, другие могут украсть ваши личные данные и даже повредить работу всей операционной системы. Рассмотрим основные виды компьютерных вирусов.

Компьютерный вирус – программа, скрытно работающая в системе, с целью нанесения вреда компьютеру. Вирус способен самостоятельно создавать и распространять свои копии.

К наиболее распространённым видам компьютерных вирусов относятся черви, троянские программы, программы-шпионы, зомби, программы-блокировщики и другие.

Червь – программа, которая делает копии самой себя. Её вред заключается в захламлении компьютера, из-за чего он начинает работать медленнее. Отличительной особенностью червя является то, что он не может стать частью другой безвредной программы.

Троянская программа (троянский конь, троян) маскируется в других безвредных программах. До того момента, как пользователь не запустит эту самую безвредную программу, троян не несет никакой опасности. Троянская программа может нанести различный ущерб для компьютера. В основном трояны используются для кражи, изменения или удаления данных. Отличительной особенностью трояна является то, что он не может самостоятельно размножаться.

Программы-шпионы собирают информацию о действиях и поведении пользователя. В основном их интересует информация (адреса, пароли).

Зомби позволяют злоумышленнику управлять компьютером пользователя. Компьютеры-зомби могут быть объединены в сеть и использоваться для массовой атаки на сайты или рассылки спама. Пользователь может не догадываться, что его компьютер зомбирован и используется злоумышленником.

Программы-блокировщики (баннеры) – это программы, которые блокируют пользователю доступ к операционной системе. При загрузке компьютера появляется окно, в котором пользователя обвиняют в скачивании нелицензионного контента или нарушении авторских прав. И под угрозой полного удаления всех данных с компьютера требуют отослать смс на номер телефона или просто пополнить его счет. Естественно, после того как пользователь выполнит эти требования, баннер никуда не исчезнет.

Вредоносная программа (Malware) – это любое программное обеспечение, созданное для получения несанкционированного доступа к компьютеру и его данным, с целью хищения информации или нанесения вреда. Термин «вредоносная программа» можно считать общим для всех типов компьютерных вирусов, червей, троянских программ и так далее.

Все представленные виды компьютерных вирусов могут в той или иной степени, нанести вред компьютеру и материальный ущерб организации.

Защита от вирусов на данный момент является едва ли не самой актуальной задачей в компьютерной индустрии. Ущерб от компьютерных вирусов может быть весьма значительным, и, чтобы не пришлось считать убытки, лучшим решением будет заблаговременно установить антивирус для комплексной защиты компьютера.

Антивирусная защита компьютера, как правило, включает в себя целый набор средств для предотвращения вредоносных действий различного характера. На рынке существует множество программ антивирусной защиты, которые

различаются ценой, скоростью работы, качеством антивирусных баз и другими параметрами. Среди них можно выделить следующие: Kaspersky Internet Security, ESET NOD32, Norton Antivirus, Dr. Web, Avast.

Персональная версия программы Avast предоставляет бесплатную антивирусную защиту компьютера. Несмотря на то, что по сравнению с коммерческими версиями этот антивирус имеет некоторые ограничения, он обеспечивает достаточный уровень антивирусной защиты компьютера и его можно рекомендовать к использованию, если нет желания платить деньги за коммерческую версию.

Для обеспечения более надёжной работы компьютера необходимо приобрести лицензионную копию антивирусной программы. В этом случае чаще всего специалисты рекомендуют антивирусную защиту Касперского. Она давно хорошо зарекомендовала себя на рынке и на сегодняшний день наиболее востребована. Стоимость лицензии антивируса Касперского на два компьютера составляет 1600 рублей за первый год и 990 за каждый последующий (версия Internet Security). Если человек не является активным пользователем интернета, то ему подойдёт стандартная версия антивируса Касперского, которая стоит дешевле, но обеспечивает хороший уровень защиты от заражения компьютера через внешние носители информации.

Существуют программы антивирусной защиты, не требующие установки. Например, Kaspersky Live CD. Её можно просто записать на диск и запускать прямо оттуда без какой-либо установки. Это удобно и доступно.

При всем многообразии средств антивирусной защиты компьютера все они имеют один существенный недостаток. Антивирусы могут эффективно находить и лечить лишь те объекты заражения, которые есть в их базе. Проблема частично решается тем, что антивирусные базы пополняются и обновляются едва ли не каждый час, но никто не может дать полной гарантии, что тот вирус, который достался именно вам, есть в этой базе.

СРЕДСТВА И ЯЗЫКИ ОПИСАНИЯ АЛГОРИТМОВ

Е. А. Христенкова
Орский гуманитарно-технологический институт (филиал) ОГУ, г. Орск

Под алгоритмом понимают набор правил, определяющих процесс преобразования исходных данных задачи в искомый результат.

Анализ примеров различных алгоритмов показывает, что запись алгоритма распадается на отдельные указания исполнителю выполнить некоторое законченное действие. Каждое такое указание называется командой. Команды алгоритма выполняются одна за другой. После каждого шага исполнения алгоритма точно известно, какая команда должна выполняться следующей. Совокупность команд, которые могут быть выполнены исполнителем, называется системой команд исполнителя.

В процессе разработки алгоритма могут использоваться различные способы его описания, отличающиеся по простоте, наглядности, компактности и другим показателям. В практике программирования наибольшее распространение получили:

1) словесная запись алгоритмов;
2) блок-схемы алгоритмов;
3) псевдокод;
4) структурограммы.

Приведём пример записи алгоритма на естественном языке, то есть на языке человеческого общения. Требуется вычислить сумму двух чисел. Обозначим эти числа A и B. Тогда алгоритм можно записать следующим образом:

1. Ввести число A.
2. Ввести число B.
3. Выполнить суммирование C := A + B.
4. Вывести число C.

Выбор и разработка алгоритма и численного метода решения задачи имеют важнейшее значение для успешной работы над программой. Тщательно проработанный алгоритм решения задачи – необходимое условие эффективной работы по составлению алгоритму.

Иногда используют полуформальный язык с ограниченным словарём (часто на основе английского языка), промежуточный между естественным языком и языком программирования. Такой язык называется псевдокодом. Запись алгоритма на псевдокоде называется структурным планом. Главная цель использования псевдокода – обеспечить понимание алгоритма человеком, сделать описание более воспринимаемым, чем исходный код на языке программирования. Псевдокод широко используется в учебниках и научно-технических публикациях, а также на начальных стадиях разработки компьютерных программ. Продемонстрируем псевдокод для решения задачи нахождения суммы двух чисел: A и B.

```
АЛГОРИТМ <Вычисление суммы>
Начало
Ввод: A
Ввод: B
Вычислить: C=A+B
Вывод: C
Конец
```

Блок-схема – распространенный тип схем, описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности. Блок-схемы можно рассматривать как графическую альтернативу псевдокоду. Для разработки структуры программы удобнее пользоваться записью алгоритма в виде блок-схемы. Для изображения основных алгоритмических структур и блоков на блок-схемах используют специальные графические символы.

Представим в виде блок-схемы алгоритм вычисления суммы двух произвольных чисел (рис. 1).

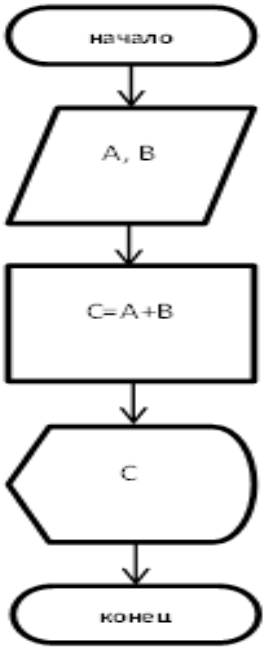

Рис. 1

Структурограммы изображают последовательность действий не с помощью линий перехода от блока к блоку, а в виде вложенных друг в друга фигур. Каждый блок структурограммы имеет прямоугольную форму и может быть вложен в любой внутренний прямоугольник другого.

Пример структурограммы представлен на рисунке 2.

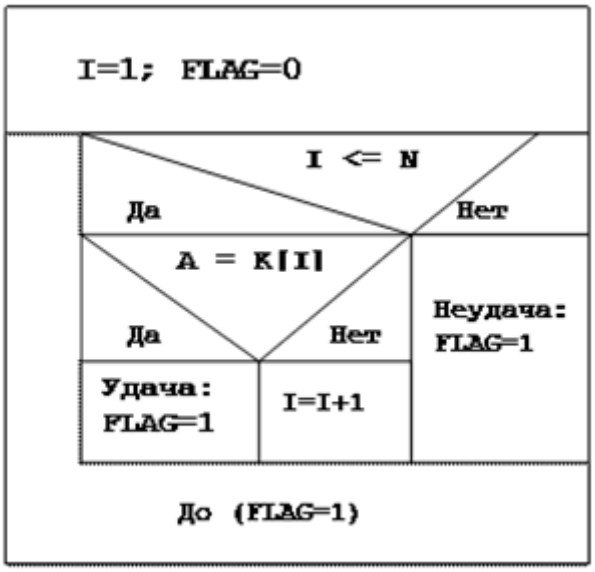

Рис. 2

Научное издание

ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ РАЗРАБОТКИ, ВНЕДРЕНИЯ И ЭКСПЛУАТАЦИИ ПРОГРАММНЫХ СРЕДСТВ

Материалы
III Всероссийской научно-практической конференции

Ответственный редактор
В. С. Янё

Ведущий редактор
Е. В. Кондаева

Редактор
Г. А. Чумак

Подписано в печать 31.08.2016 г.
Формат 60×84 1/16. Усл. печ. л. 3,8.
Тираж 42 экз. Заказ 34/1480.

Издательство Орского гуманитарно-технологического института (филиала) федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет»

462403, г. Орск Оренбургской обл., пр. Мира, 15А