Московский государственный университет печати имени Ивана Фёдорова
Кафедра медиасистем и технологий

Анна Юрьевна Филиппович

ПРИНЦИПЫ ПОСТРОЕНИЯ КОМПЬЮТЕРНЫХ ШРИФТОВ

Лекции по дисциплине
"ИНТЕГРИРОВАННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ В МЕДИАИНДУСТРИИ"

Для студентов,
осваивающих образовательные программы
подготовки бакалавров и магистров по направлениям
230400 «Информационные технологии и системы»,
230100 «Информатика и вычислительная техника»

Москва, 2013

Система измерений типографских шрифтов

В основу измерений типографских шрифтов положены система Дидо, распространенная в Европе и принятая в России, и так называемая англоамериканская система или система Пика. И в том, и в другом случае основной единицей измерения является типографский пункт, равный в системе Дидо 0,376 мм, а в системе Пика — 0,352 мм.

Форматы шрифтов

В отношении компьютерного шрифта слово формат (format) используется в двух смыслах. Во-первых, формат определяется платформой, для которой шрифтовой файл создан. Например, два шрифтовых файла с одинаковыми данными для одних и тех же гарнитур могут иметь разные форматы в зависимости от того, предназначены они для платформ Apple Macintosh или Windows PC. Во-вторых, формат шрифтового файла отражает способ представления и организации собственно типографической информации. Рассмотрим три основных шрифтовых формата – *PostScript (PFM), TrueType (TTF) и OpenType (OTF)*.

Шрифты в формате **PostScript** основаны на языке описания страниц PostScript, и для их обработки и отображения требуется интерпретатор этого языка – RIP[1]. У принтеров с высоким разрешением и фотонаборных автоматов такой интерпретатор обычно встроен в устройство. Он представляет собой отдельный процессор, предназначенный для преобразования PostScript-кодов в управляющие коды устройства. Для устройств с низким разрешением, какими являются экран монитора и настольные офисные принтеры, PostScript-шрифты отображаются PostScript-интерпретатором, встроенным в операционную систему, или с помощью дополнительного приложения, которое называется Adobe Type Manager (ATM). PostScript-шрифты обычно снабжаются еще и комплектом растровых шрифтов для экранного отображения в системах без PostScript-интерпретатора.

В настоящее время PostScript-шрифты стали стандартом в издательской отрасли, поскольку они на нее ориентированы. Практически все устройства, используемые в издательствах снабжены растровыми процессорами (RIP). Естественно, такие процессоры лучше всего работают с PostScript-шрифтами.

Кроме этого PostScript-шрифты построены с помощью кривых 3-го порядка, так называемых кривых Безье (рис. 1). За счет этого достигается большая гладкость контуров и компактность шрифтовых файлов.

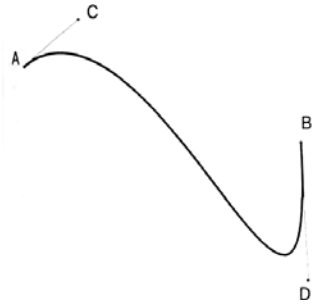

Рис. 1. Элементарная кривая в шрифтах PostScript (кривая Безье или кубическая парабола)

В течение нескольких лет в конце 1980-х годов в области компьютерного шрифта и наборных процессов PostScript-шрифт являлся первым и единственным стандартом цифровых шрифтовых форматов (font format). Однако, в последствии фирмами Apple Computer и Microsoft был создан новый формат – **TrueType**, который дал возможность

[1] RIP – raster image processor, растровый процессор.

обеим компаниям встроить отображения шрифта в свои операционные системы, не будучи ничем обязанными компании Adobe.

Хотя предполагалось, что шрифты TrueType совместимы с PostScript-интерпретаторами, на фотонаборных автоматах возникали проблемы с выводом шрифтов этого формата. По этой причине PostScript-шрифты остались форматом, который предпочитают профессиональные издатели. Эти проблемы не утратили своей остроты, хотя популярность шрифтов TrueType в ОС Windows и новые коммерческие взаимоотношения компаний Adobe и Microsoft привели к более устойчивой работе PostScript-устройств.

В формате TrueType нашли свое воплощение несколько улучшений по сравнению с PostScript-шрифтами. TrueType-шрифты обычно распространяется без создаваемых вручную экранных (растровых) вариантов. Экранное представление шрифта генерируется непосредственно из контура знака. Кроме этого формат TrueType допускает размещение более широкого комплекта знаков. В нем найдется место для альтернативных форм знаков и возможность контекстной замены знаков (contextual character switching). Это значит, что при определенных условиях один знак автоматически заменяется другим.

TrueType-шрифты в отличии от PostScript построены на кривых 2-го порядка. Каждый участок контура задается двумя точками – границами и направлением линии на каждой из границ (рис. 2).

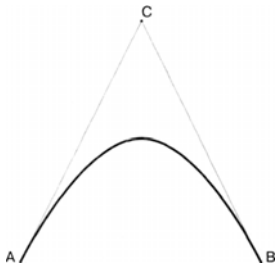

Рис. 2. Элементарная кривая в шрифтах TrueType (парабола второго порядка).

Т.о. для описания контура символа TrueType-шрифта требуется большее количество точек, по сравнению с PostScript (рис. 3), поэтому они объемнее. За счет большего числа степеней свободы PostScript-линия не имеет изломов в точках сопряжения фрагментов, тогда как для TrueType больший или меньший перелом линии в точке стыковки двух сегментов является почти неизбежным злом. Иначе говоря, символы PostScript-шрифта являются более гладкими, чем TrueType.

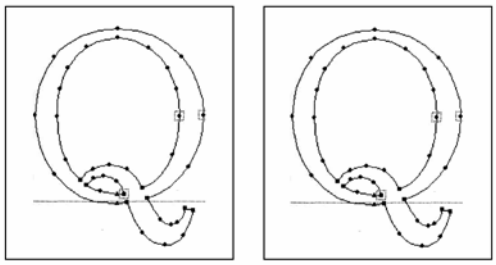

Рис. 3. Литеры шрифтов: а – PostScript-шрифт; б – TrueType шрифт.

Формат **OpenType** является гибридным, он создан компаниями Adobe и Microsoft и сглаживает различия двух форматов, позволяя им сосуществовать в одном шрифтовом файле. Он также дает возможность один и тот же шрифтовой файл использовать в обеих операционных системах Macintosh и Windows.

Проще говоря, шрифт формата OpenType – это шрифт TrueType с «кармашком» для PostScript-данных. Шрифт OpenType может содержать данные формата TrueType, данные формата PostScript или (теоретически) обоих форматов. Таким образом, существует

потенциальная возможность оптимальным образом объединить лучшие стороны обоих форматов. Операционная система сама сортирует данные шрифта OpenType и выбирает только те из них, которые ее устраивают.

Проблема шрифтовых файлов форматов OpenType и TrueType состоит в том, что будучи просто пользователем шрифта трудно узнать, что у них внутри. Формат PostScript-шрифтов обычно содержит только стандартный комплект знаков со стандартными параметрами. А формат TrueType, и в еще большей степени формат OpenType, предлагает широкий набор дополнительных параметров, которые могут включаться, а могут и не включаться в каждый конкретный шрифтовой файл. Например, формат OpenType может содержать от 256 до 65 536 знаков. И не существует способа узнать об этом, если только параметры шрифта не отражены в каком-либо сопроводительном документе.

Кодировка шрифта

Вся информация в компьютере, в том числе и текстовая, хранится в виде двоичных чисел (кодов).

Сейчас для кодировки шрифтов используется международный стандарт Unicode. Он расширяет кодовую схему, включая знаковые комплекты для нелатинских алфавитов. Большинство шрифтовых файлов стандарта Unicode двухбайтовые и могут содержать до 65000 знаков. Стандарт Unicode обеспечивает межплатформенную совместимость шрифтов, эту кодовую таблицу поддерживают Macintosh OS X и Windows NT 4, Windows 2000 и Windows XP.

У первых шрифтовых файлов имелись ячейки только для 256 знаков, и данный комплект знаков остается стандартом для большинства шрифтовых файлов. Иногда, даже если формат шрифта поддерживает 65000 знаков, он включает только 256 стандартных.

До Unicode единственным межплатформенным стандартом кодировки был американский стандартный код для обмена информацией (сокращенно ASCII или просто ASC-код), разработанный для телетайпа и других подобных систем связи. Код ASCII первоначально являлся семибитным и включал в себя символы с кодами от 32 до 128 (кодам от 0 до 31 соответствовали неотображаемые, служебные символы, типа – «звонок», 10 – «перевод строки», 13 – «возврат каретки»). Для отображения символов национальных алфавитов, символов псевдографики и некоторых математических символов таблица ASCII-кода была расширена до 16 бит, получившийся в результате код стали называть «расширенным ASCII-кодом».

До того как Macintosh и Windows стали поддерживать стандарт Unicode, они использовали разные кодовые схемы (таблицы). Таблицы совпадали в основной части комплекта ASCII, но различались в знаках, имеющих коды после 128, так называемые знаки старших разрядов (high-bit). Результатом стало то, что документы, кодировавшиеся на одной компьютерной платформе, на другой очень часто отображались некорректно.

И дело не только в том, что операционные системы до стандарта Unicode использовали разные таблицы кодирования, а в том, что они применяли разные подмножества комплекта Latin 1 в качестве своих стандартных комплектов знаков.

Комплект системы Macintosh (и кодовая таблица) называется MacRoman; а комплект системы для Windows (и кодовая таблица) называется Win ANSI. Хотя распространители шрифта могут продавать одноименные шрифтовые файлы для обеих платформ, пользователи системы Macintosh получат в шрифтовом файле одну группу знаков, а пользователи системы Windows – другую. Определенные знаки в комплекте MacRoman заимствованы из шрифтового файла Symbol. Когда вы работаете на компьютере Macintosh, то кажется, что эти знаки являются частью каждого шрифта.

Понятие о формате и шрифтовой машине

Любой цифровой шрифт, как это сразу становится понятно из названия, представляет собой описание входящих в него символов, метрических и других параметров, определяющих особенности шрифта в цифровой форме. Форматом представления цифрового шрифта называется способ (стандарт) представления цифровой информации, образующей шрифт. Обычно он представляет собой один или несколько файлов, с которыми можно поступать так же, как и с любыми другими файлами: копировать, удалять, переименовывать и т.д.

Шрифт, представленный в определенном формате, можно использовать в любых программных и аппаратных средствах, которые могут воспринимать закодированную в формате информацию. Таким образом, создание определенного формата представления шрифтов не является достаточным для их использования. Необходимо иметь еще два компонента: средства преобразования информации в заданный формат и средства воспроизведения шрифтов, представленных в этом формате. Если средства кодирования используются в основном производителями шрифтов, то средства воспроизведения необходимы в первую очередь пользователям цифровых шрифтов. Совокупность определенного формата представления шрифтов и средств воспроизведения шрифтов, заданных в этом формате, мы будем называть *шрифтовой машиной* (рис. 4).

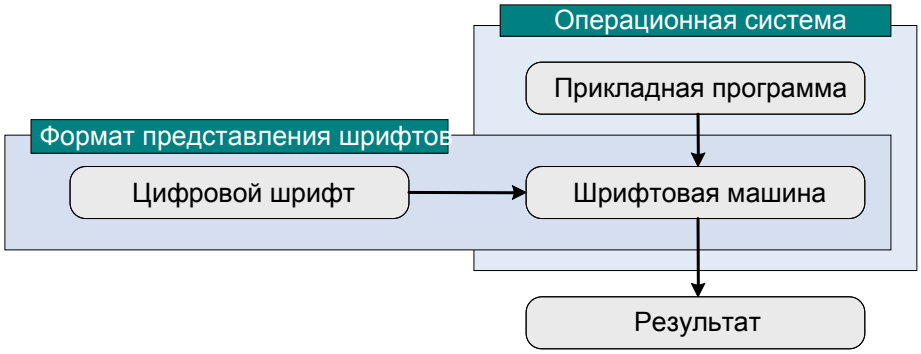

Рис. 4. Схема реализации шрифтовой машины

Очевидно, что без средств воспроизведения любой шрифтовой формат имеет только теоретический интерес, поскольку невозможно оценить качество воспроизведения шрифта и скорость работы. Поэтому бессмысленно говорить о преимуществах того или иного формата, оценивать можно только работу шрифтовой машины.

Структура шрифтового формата

Как и любой шрифт, имеющий определенный набор параметров, повторяющихся от шрифта к шрифту, любой шрифтовой формат имеет некоторые обязательные части. Перечислим их с краткими пояснениями.

Область заголовка. В этой части располагается следующая информация:

1. Информация о различных вариантах наименования шрифта (рабочее имя шрифта, имя гарнитуры и начертания, полное имя шрифта, имена и индексы, под которыми шрифт воспринимают прикладные программы).
2. Информация о создателях шрифта (знак принадлежности прав, ссылка на автора исходного рисунка шрифта, информация о торговой марке, история создания шрифта).
3. Регистрационная информация, предназначенная для автоматической классификации шрифта и обеспечения подстановки шрифтов. Обычно в этой области расположены описания насыщенности, угла наклона и

пропорциональности шрифта, а также код шрифта по одной из систем описания шрифтов.

4. Статистическая информация о шрифте (минимальный охватывающий прямоугольник[2], количество символов и др.).

Область описания метрических параметров. В этой части описываются все измерения символов. Обычно к ним относят информацию о ширине символов, минимальные охватывающие прямоугольники для всех символов, информацию о кернинге и трекинге шрифта. В некоторых форматах (например, в формате PostScript) информация о трекинге и кернинге сохраняется в отдельном файле.

Область описания общих элементов. Некоторые символы имеют одинаковые элементы. Для сокращения объема шрифтового файла и для того, чтобы гарантировать действительную одинаковость этих элементов, они отделяются от символов. Символы содержат только ссылки на такие элементы. То же самое относится и к некоторым средствам разметки, общим для нескольких символов.

Область описания системы кодирования. В этой области располагаются кодовые таблицы, относящиеся к шрифту.

Область описания разметки символов. В этой области находится информация о разметке символов, необходимая для их качественного воспроизведения.

Область описания символов. Это – основная часть шрифтового файла. В ней находится описание самих символов. Для формирования контуров символов могут использоваться различные математические и логические методы. Обычно метод описания контуров и определяет эффективность работы, а также особенности растеризации шрифтов определенного формата.

Управление растеризацией символов

Как уже говорилось, фундаментальной особенностью контурных шрифтов является отделение информации о форме символов от процесса их воспроизведения на растровом выводном устройстве. Если контуры символов шрифта можно описывать самыми разными способами, то задача воспроизведения, в конечном итоге, сводится к активизации некоторых точек (высвечиванию на экране дисплея или заполнении краской при печати на принтере).

Алгоритм растеризации

Итак, при воспроизведении каждого символа на растровом устройстве (например, на лазерном принтере) необходимо решить две задачи:

1. масштабировать (уменьшить или увеличить) контур символа до необходимого размера. Например, при печати текста 10 кеглем на лазерном принтере с разрешением 300 точек на дюйм (12 точек на миллиметр) необходимо, чтобы контур символа H имел примерно 28 точек в высоту;
2. активизировать все точки, попавшие во внутренние области этого контура, заполнить контур.

Проблемы растеризации

В ходе решения этих простых, на первый взгляд, задач возникает немало проблем, связанных с масштабированием и заполнением контуров. Перечислим некоторые из них.

Нарушение пропорций символа. При воспроизведении символов на устройствах с малой разрешающей способностью (300 точек на дюйм и меньше), особенно при выводе

[2] Минимальный охватывающий прямоугольник шрифта — это прямоугольник минимального размера, в который целиком помещаются все символы шрифта помещаются все символы шрифта (кегельная площадка).

текста небольшим кеглем (12 и меньше), сильно сказываются ошибки масштабирования. Масштабирование происходит в абсолютных координатах относительно некоторой произвольной точки и всегда приводит к получению целочисленного результата. При этом возникает проблема округления нецелых результатов. Например, если координаты некоторого элемента символа в системе координат описания контура равны (200; 100), то при уменьшении размера контура в 3 раза они трансформируются в (66. 666666; 33. 333333). Поскольку нам нужны целые значения, они превратятся в (67; 33), то есть значение горизонтальной координаты немного (на треть точки) увеличится, а горизонтальной – на столько же уменьшится. Если при этом специально не учитывать особенности формы символа, то он может сильно исказиться и даже стать нечитаемым. На рис. 34 приведен пример подобного масштабирования символа Н:

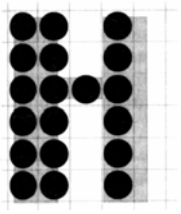

Рис. 5. Пример масштабирования символа Н

Нарушение симметричности некоторых символов. Прежде всего этот дефект относится к символам, обладающим симметрией, таким, как А, Ж, М, О, Т, Ф, Ш, и некоторым другим. Нарушение симметричности таких символов (например, возникновение разного расстояния между вертикальными штрихами буквы Ш) резко искажает их форму и затрудняет чтение текста.

Нарушение единства символов. Применяя некоторые приемы, мы можем избавиться от ошибок округления применительно к одному символу. Но при этом мы рискуем потерять единство символов в шрифте. Например, если в символе Н мы будем округлять толщину вертикальных штрихов в меньшую сторону, а в символе Ш – в большую, то некоторые слова станут трудно воспринимаемыми. Кроме того, при таком подходе нарушается ритмичность шрифта (характерный случай – разное округление расстояния между вертикальными штрихами в символах Ш и Щ).

Другой пример – масштабирование положения горизонтальных линий (например, средних линий символов в, е, ж, з, к) и величины оптических наплывов у округлых букв (таких, как а, б, е, з, о, с). В первом случае может возникнуть неприятный разнобой в некоторых словах, а во втором – искажение базовой линии текста и скачки букв в вертикальном направлении.

Смыкание штрихов. В некоторых случаях некачественного масштабирования штрихи и другие элементы символов смыкаются между собой. Наиболее часто это происходит с вертикальными штрихами в узких шрифтах. Ошибочное соединение штрихов (рис. 6) нарушает графему такой буквы, и человек теряет способность к ее распознаванию.

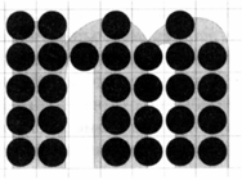

Рис. 6. Пример смыкания элементов символа.

Выпадение точек. Если не обращать внимания на прохождение линий при округлении координат опорных точек контура, то часто возникают ситуации, в которых программа заполнения масштабированного контура не может определить, какие именно растровые точки необходимо активизировать. Как правило, эта проблема возникает при заполнении тонких наклонных элементов (рис. 7).

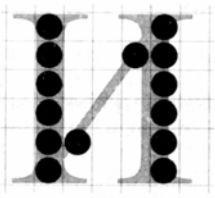

Рис. 7. Пример выпадения точек при заполнении контура.

Нарушение формы округлых букв. Этот дефект не так резко, как другие, влияет на удобство восприятия текста. Он «только» искажает форму символов, имеющих большие округлые элементы, например В, О, 3, Р, С, а, б и др. Вопрос о заполнении таких элементов можно решать разными способами, но лишь некоторые из них позволяют получить действительно качественное изображение буквы, а остальные приводят к подобным ошибкам, приведенным на рис. 8.

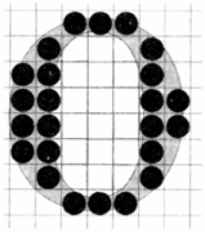

Рис. 8. Нарушение формы округлых букв.

Теперь, когда мы выявили некоторые проблемы, связанные с растеризацией символов, рассмотрим методы устранения этих проблем. Для этого прежде всего введем понятие разметки шрифта. *Разметкой* мы будем называть описание символов, их элементов и шрифта в целом, призванное улучшить качество растеризации символов. Иногда разметку называют хинтингом (от англ. *hint* – подсказка), но этот термин обычно относят к шрифтам в формате Type 1 (для TrueType шрифтов используют понятие инструкций), поэтому мы считаем необходимым ввести новый, более общий, термин.

Методы разметки символов

Существует два основных метода разметки символов контурных шрифтов: декларативный и программируемый. Первый применяется в формате Adobe Type 1, а второй – в TrueType шрифтах.

Декларативный метод разметки основан на описании особенностей символа при помощи их декларирования отдельно от описания контура (рис. 9). То есть описание символа при этом включает в себя две части: математическое описание контура символа и декларирование его особенностей:

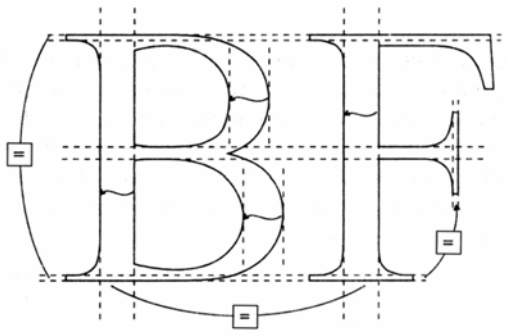

Рис. 9. Элементы разметки при декларативном методе.

Задачу связывания этих частей и построения правильных ассоциаций решает программа растеризации. Именно она анализирует форму символа, связывает ее с заданной разметкой и принимает решения об изменении контура в ходе его масштабирования и заполнения. Таким образом в шрифтовой машине, разметка символов в которой производится декларативным методом, основную часть работы по улучшению формы символов выполняет растеризатор. Обычно он представляет собой довольно сложную программу, содержащую множество высокоэффективных алгоритмов (ведь символы приходится воспроизводить очень быстро) и элементы искусственного интеллекта.

Огромное преимущество декларативной разметки – простота построения шрифтов. Так как производителей шрифтов намного больше, чем производителей растеризаторов, применение этого метода приводит к более быстрому появлению новых гарнитур.

Программируемый метод разметки основан на точном определении в шрифте всех действий, которые должен выполнять растеризатор. На долю растеризатора при этом остаются только интерпретация команд разметки и как можно более быстрое их выполнение. Растеризатор оказывается более простым, компактным и быстрым, но это происходит за счет резкого усложнения шрифтов и увеличения их в объеме. Программа разметки может быть очень сложной, имеющей циклы, условные переходы, описания переменных и массивов (рис. 10). Языки программирования разметки обычно имеют много команд модификации контуров символов, причем среди них есть как команды, работающие на этапе масштабирования контура, так и на этапе его заполнения.

В программируемом методе разметки используются не ассоциативные декларации, а точное указание взаимодействия между точками:

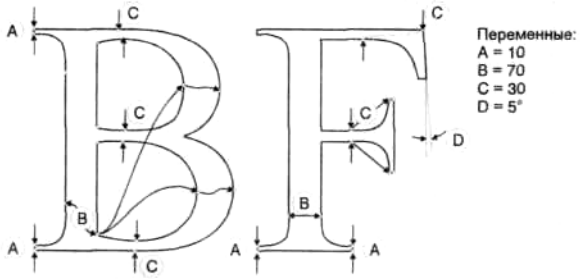

Рис. 10. Элементы разметки при программируемом методе.

Потенциально программируемая разметка может обеспечить намного лучшее качество, чем декларативная, но создание высококачественных шрифтов, использующих программы разметки, оказывается настолько трудоемким, что таких шрифтов появляется довольно мало. Обычно производители таких шрифтов (а это все TrueType-шрифты) применяют специальные системы, автоматически формирующие программы разметки символов и шрифта. Такой путь обычно приводит к невысокому качеству растеризации шрифтов, так что потенциальное преимущество программируемой разметки теряется.

Общая структура шрифта в формате Type 1 (PostScript)

Любой Type 1-шрифт состоит из двух основных частей: открытой и закрытой (зашифрованной):

Открытая часть:

- Обозначение шрифта
- Заголовок шрифта
- Кодовая таблица шрифта
- Уникальный идентификатор шрифта

Закрытая часть:

- Область глобальной разметки
- Область глобальных подпрограмм
- Область подпрограмм разметки и контурных подпрограмм
- Область описаний символов

Открытая часть. В открытой части Type 1-шрифта содержится информация, доступная для любого текстового редактора. Эта часть может быть изменена при условии, что закрытая часть останется нетронутой. В открытой части можно выделить 4 области.

Обозначение шрифта показывает, что файл является именно шрифтом:

%! PS-AdobeFont-1.0: TimeRoman 001.1

%%CreationDate: Wed Oct 20 17:08:26 1993

%%Creator: FontLab(c) for Windows v2.5

Заголовок шрифта, в котором хранится следующая информация:

Регистрационное имя рифта FontName;

Полное имя шрифта FullName;

Имя гарнитуры, в которую входит шрифт FamilyName;

Наименование версии рифта Version;

Информация о создателях шрифта и об авторских правах на шрифт Notice;

Информация о насыщенности шрифта Weight;

Угол наклона символов шрифта в градусах против часовой стрелки ItalicAngle;

Информация о том, является ли шрифт моноширинным IsFixedPitch;

Положение линии подчеркивания UnderlinePosition;

Толщина линии подчеркивания UnderlineThikness;

Вид шрифта PaintType: 0 – сплошной (заполняемый); 1 – контурный. Все Type 1-шрифты являются сплошными;

Тип шрифта Font Type: 0 – Type 0; 1 – Type 1; 3 – Type 3;

Стандартная матрица трансформирования символов FontMatrix. Ее более подробное описание приведено в разделе «Описание символов».

Минимальный прямоугольник, охватывающий все символы шрифта. FontBBox

Приведем пример типичного заголовка Type 1-шрифта:

```
/Fontlnfo 9 dict dup begin
/FullName (Times New Roman) readonly def
/FamilyName (Times) readonly def
/version (001. 1) readonly def
/Weight (Normal) readonly def
/Notice ((c) Copyright Monotype, 1990) readonly def
/ItalicAngle 0 def
/isFixed Pitch false def
/UnderlinePosition -100 def
/UnderlineThickness 50 def end readonly def
/FontName /TimesNewRoman def
/PaintType 0 def
/FontType 1 def
/FontMatrix [ 0. 001 0 0 0. 001 0 0 ] readonly def
/FontBBox { -63 -231 1148 882 } readonly def
```

Кодовая таблица шрифта определяет связь между именами и кодами символов. В Type 1-шрифтах все символы имеют уникальные имена, которые однозначно их идентифицируют. Кодовая таблица позволяет установить некоторое соответствие между кодами символов, с которыми работают программы, использующие шрифт, и именами символов. Поскольку кодовая таблица находится в открытой части шрифта, ее можно изменять, тем самым меняя кодировку, в которой работает шрифт. Кодовая таблица представляет собой набор пар вида: <код> <имя>. Код – это 8-разрядный код символа (от

0 до 255), а имя – это строка, не имеющая пробелов. В формате Type 1 в именах символов различаются прописные и строчные буквы.

Хотя кодовая таблица Type 1 -шрифтов позволяет использовать только 8-битные значения для кодов, то есть с ее помощью можно определить не более 256 разных символов, Type 1 -шрифт может содержать любое их количество. В кодовой таблице символы, не попадающие в 256-знаковую область никак не отражаются, но они присутствуют в шрифте под своими именами, отличающимися от других. Изменяя кодовую таблицу (напомним, что это можно делать, не затрагивая остальной шрифт), можно получить доступ ко всем символам.

Уникальный идентификатор шрифта – 24-разрядное число (от 0 до 16777215). Идентификатор должен определять один и только один шрифт. В случае использования двух шрифтов с одинаковыми идентификаторами возможно возникновение серьезных ошибок. Идентификаторы в диапазоне 4000000 – 4999999 могут использоваться для внутренних целей любой организации. Для других шрифтов (например, ориентированных на продажу) необходима регистрация идентификаторов в фирме Adobe.

Закрытая часть – это основная часть любого Type 1-шрифта, в которой содержатся описания символов и информация об их разметке. Закрытая часть шрифта определяется его создателями, шифруется при помощи особого алгоритма и не может быть изменена после загрузки шрифта в принтер. Вообще говоря, шифрование этой части потеряло всякий смысл после того, как в 1990 году был опубликован алгоритм дешифровки, но для обеспечения совместимости со старыми устройствами шрифты продолжают зашифровывать. Кроме того, шифрование закрытой части Type 1-шрифтов немного ограничивает возможности тех, кто нелегально пытается их изменить и выдать за свои. Теоретически, сам акт дешифровки может в некоторых случаях считаться нарушением авторских прав.

В закрытой части есть области, зашифрованные дважды, – это описания подпрограмм и символов. При этом для дополнительной экономии места применяется специальный метод кодирования числовых значений и команд.

Зашифрованная часть начинается после слова eexec и, так же, как и открытая, состоит из четырех областей.

Область глобальной разметки, в которой содержатся описания параметров шрифта, которые используются для улучшения качества растеризации. Вот краткое описание некоторых из них.

BlueValues – массив пар чисел (до 7 пар в возрастающем порядке), определяющих зоны выравнивания сверху (кроме первой пары, которая определяет зону выравнивания базовой линии снизу).

OtherBlues – массив пар чисел (до 5 пар в возрастающем порядке), определяющих зоны выравнивания снизу, например для нижних выносных элементов.

BlueShift – определяет величину оптического наплыва (в точках выводного устройства), начиная с которой отключается его подавление.

StdHW и StdVW определяют наиболее распространенные толщины горизонтальных и вертикальных штрихов. В том случае, когда после масштабирования контура толщины штрихов мало отличаются от стандартных значений, используются эти значения, что улучшает внешний вид символов и скрадывает ошибки построения контуров.

Приведем пример описания этих значений в шрифте.

```
/BlueValues [ -16 0 488 504 712 728 752 752 ] ND
/OtherBlues [ -224 -221 ] ND
/BlueShift 7 def
/StdHW [ 48 ] ND
```

Область глобальных подпрограмм содержит несколько подпрограмм, написанных на языке PostScript. Обычно они используются для реализации наиболее сложных методов разметки.

Область подпрограмм разметки и контурных подпрограмм. Язык описания Type 1-шрифтов, как и PostScript, имеет встроенные возможности для структурной организации программы, реализованные в виде команд вызова глобальных (PostScript) и локальных (написанных на языке Type 1) подпрограмм. Локальные подпрограммы обычно применяются для организации сложной разметки символов, например для смены хинтов, и для описания повторяющихся элементов символов.

Область описания символов – основная область PostScript Type 1-шрифта, определяющая изображения всех символов шрифта. Описание каждого символа включает его имя, ширину левого поля, ширину символа (расстояние от линии левого поля до линии правого поля), описания разметки и контура.