Силезский университет

Технологический факультет

Институт прикладной информатики

МАГИСТРСКАЯ ДИССЕРТАЦИЯ

Тема:

Эмуляция принтера и плоттера в графическом режиме

Выполнил: Руководитель:
Рышард Чекай, проф. доктор хаб. в . Дариус Бадура
Павел Шимик Консультант:
магистр Иренеуш Гоцинак

магистр Ирена Гощинах

СОСНА 1996

Вступление

Помимо печати простых текстов, современные принтеры позволяют использовать разнообразные масштабируемые шрифты и даже размещать на странице графические изображения. Плоттеры, как и принтеры, управляются специальными последовательностями, которые во многих случаях можно рассматривать как языки управления печатью. Поток управляющих кодов вместе с данными для печати можно отправить на принтер или сохранить в файле на диске компьютера. Такой файл, содержащий данные для принтера или плоттера, называется файлом печати. Этот файл впоследствии можно отправить на устройство, подключенное к другой компьютерной станции. Файл печати содержит управляющие последовательности, характерные только для определенного типа устройства, которые впоследствии будут его правильно интерпретировать.

Целью данной работы является разработка компьютерной программы, эмулирующей работу выбранного принтера и плоттера, что позволит графически представить файл печати на экране компьютера.

При достижении этой цели были сделаны следующие предположения:

- имеется файл печати выбранных принтеров и плоттеров в соответствующем им формате
- известна модель принтера или плоттера, для которого был создан файл печати
- файл печати может содержать простые текстовые данные и графическое изображение
- разработанная программа максимально точно представит изображение, содержащееся в файле
- программа позволит распечатать данные на принтере, отличном от того, для которого был подготовлен файл печати
- для сборки программы рекомендуется использовать графическую среду Windows 3.xx и пакет средств программирования Borland C++ .

Технические аспекты работы принтера и плоттера

1. Классификация принтеров

1.1. Матричные принтеры (принтеры ударного действия)

В матричном принтере изображение создается печатающей головкой, оснащенной серией штифтов (обычно 9, 24 или 48), которые ударяют по ленте, пропитанной чернилами, когда печатающая головка движется вперед и назад перед бумагой. Иглы расположены в один или несколько вертикальных рядов.

Другая, менее известная технология, называемая Comb Matrix или Shuttle Matrix, использует горизонтальный ряд игл (оптимально по одной игле для каждого горизонтального символа), поскольку один ряд точек может лучше воспроизвести символ. В случаях, когда игл не столько, сколько символов в горизонтальных строках, гребенка слегка перемещается вперед и назад, чтобы напечатать необходимые точки. Благодаря этой технологии бумага может продвигаться практически непрерывно, поэтому ее часто используют в высокоскоростных матричных принтерах. Эта технология намного эффективнее, чем печать отдельных символов.

Ударные матричные принтеры универсальны, т.е. игольчатую матрицу можно настроить для печати различных шрифтов и даже графики.

Матричные принтеры можно использовать для печати копий вместе с оригиналом, поскольку изображения создаются путем нанесения чернильной ленты. В этих принтерах используется непрерывная бумага, хотя большинство моделей также могут использовать отдельные листы и этикетки.

Из-за особенностей метода печати матричные принтеры значительно громче других моделей (даже в так называемом тихом режиме).

Скорость печати, обычно указываемая в cps (символах в секунду), варьируется в зависимости от используемых шрифтов (Draft/LQ) и различных значений разрешения cpi (точек на дюйм) [16], [15], [8].

1.2. Принтеры и пишущие машинки Daisywheel

В лепестковых принтерах отпечаток создается с помощью барабана, на котором выдавлены все буквы и символы. Головка движется вперед и назад по бумаге, и электромагнит начинает работать, когда барабан устанавливается в положение, соответствующее требуемой букве.

Принтеры Daisywheel были популярны, поскольку качество их печати было лучше, чем у более ранних моделей, таких как девятипиксельные матричные принтеры.

Их основными недостатками являются: низкая скорость печати, ограниченное количество символов на барабане и отсутствие возможности создания графики.

Принтеры Daisywheel — это разновидность принтеров ударного действия. Это позволяет распечатать копию одновременно с оригиналом. Печать возможна как на рулонной бумаге, так и на отдельных листах и этикетках.

Скорость печати обычно указывается в символах в секунду (cps) [8].

1.3. Струйные принтеры

Струйные принтеры работают по принципу матричных принтеров, которые создают отпечаток путем размещения точек на бумаге. В струйных принтерах капли чернил распыляются непосредственно на бумагу. Гораздо более плотный рисунок точек, чем в матричных принтерах, позволяет получать более качественные отпечатки.

В новых моделях струйных принтеров чернила не выбрасываются электростатическим способом или с помощью миниатюрных насосов, а преобразуются в пар при нагревании и под давлением расширяющегося пара осаждаются на бумаге.

Струйные модели не являются принтерами ударного действия, поэтому копии печатаются отдельно. Возможна печать на порезанной бумаге, как на копировальном аппарате. Для улучшения качества печати некоторым моделям требуется бумага особого качества, хотя для большинства принтеров достаточно стандартной бумаги.

Некоторые принтеры могут использовать непрерывную бумагу и даже печатать этикетки, при условии, что они изготовлены из бумаги, хорошо впитывающей чернила.

Струйные принтеры работают тише и намного быстрее, чем ударные принтеры.

Параметр скорости печати указывается в cps (символах в секунду), а иногда, для различных шрифтов, в ppm (страницах в минуту).

рысь.И HP DeskJet 600C

Хотя чернила этих принтеров водорастворимы, они сильно различаются по смачиваемости. Это может быть важно, если отпечаток будет использоваться для презентаций в условиях высокой влажности.

Некоторые типы чернил не являются водостойкими. К таким чернилам относятся: все чернила Canon , черные чернила DEC 520ic и чернила TI Micromarc . Однако черные чернила HP DeskJet (рис. 1) почти так же водостойки, как и лазерная печать. Черные и цветные чернила Epson , по-видимому, являются наиболее водостойкими : буквы растекаются, но не размазываются при воздействии влаги.

К сожалению, цветные (струйные) отпечатки не так долговечны, как фотографии. Те, которые устойчивы к свету, часто разрушаются под воздействием влаги, а другие быстро выцветают на ярком свету. Долговечность печати можно увеличить с помощью ламинирования. К сожалению, современные цветные струйные принтеры не используются достаточно долго, чтобы проверить долговечность новейших чернил [8], [10], [11].

1.4. Лазерные принтеры и светодиодные принтеры

Лазерные и светодиодные модели не являются принтерами ударного действия, поэтому копии необходимо печатать отдельно. Обычно они используют отдельные листы бумаги. Они могут печатать этикетки, выдерживающие высокую температуру во время горения тонера.

Для создания отпечатка на лазерном принтере специальный барабан заряжается электростатически, а рисунок страницы создается путем разряда барабана лазерным лучом или светодиодными линиями в соответствующих точках. Тонер прилипает к разряженным точкам. Эти точки являются зеркальным отображением того, где носитель будет отображаться на окончательной распечатке.

Бумага проходит через коронирующий провод, который придает ей электрический заряд, противоположный заряду точек на барабане, затем проходит через цилиндр, на поверхности которого удерживается тонер. Наконец, он перемещается под нагревательный элемент, который сплавляет тонер с бумагой.

Большинство современных лазерных принтеров не имеют коронирующего

провода, и бумага электризуется с помощью провода. Лазерные и светодиодные принтеры создают очень четкие и точные отпечатки с разрешением более 300 точек на дюйм (достигая даже 1200 точек на дюйм). Некоторые принтеры используют различные технологии, такие как технология улучшения разрешения, для управления расстоянием между точками и сглаживания контуров.

рис. Б HP LaserJet 4V

Лазерные и светодиодные принтеры работают относительно тихо. Производительность печати лазерных и светодиодных моделей измеряется в стр./мин (страницах в минуту) в режиме копирования (т. е. изображение страницы, уже находящееся в памяти принтера, копируется на бумагу как можно быстрее) [8], [10], [11].

1.5. Цветные принтеры

Цветные принтеры используют несколько методов для создания цветной распечатки.

Независимо от метода, используемого для создания цветной печати, существует три различных способа указания цветов:

RGB (красный-зеленый-синий): стандартный метод, используемый в сочетании с такими устройствами, как мониторы и телевизоры. Это аддитивный процесс смешивания цветов (белый цвет получается путем смешивания всех трех цветов одинаковой интенсивности).

CMYK (голубой-пурпурный-желтый-черный): метод, используемый в основном в печати в коммерческих целях. Это субтрактивный процесс смешивания цветов (черный цвет получается путем смешивания всех трех цветов одинаковой интенсивности или путем добавления дополнительного черного цвета, который дает более насыщенный и темный черный цвет).

Дизеринг: Метод, при котором цвета на самом деле не смешиваются, а путем размещения точек разных цветов в различных затененных областях изображение выглядит так, будто в нем используется более четырех цветов. Этот метод похож на метод создания оттенков серого путем установки большего или меньшего количества черных точек.

Цветные матричные принтеры используют цветную ленту. Струйные принтеры

содержат картриджи с цветными чернилами. Некоторые из них создают черный цвет путем смешивания цветов (HP DeskJet 600 C Рис.1), другие имеют отдельный картридж с черными чернилами (HP DeskJet 660 C).

В термопринтерах используется пленка с воском четырех цветов. Принтер подает бумагу четыре раза, по одному разу для каждого цвета. При переносе каждого цвета на бумагу создается полноцветное изображение.

Некоторые производители недавно представили цветные лазерные принтеры с четырьмя отдельными контейнерами для тонера, по одному для каждого основного цвета.

Цветные лазерные принтеры обычно используют четыре цвета тонера: три основных субтрактивных цвета (голубой, пурпурный и желтый) и черный.

Видимый спектр непрерывен, поэтому вы можете лишь приблизительно воспроизвести видимые цвета, смешивая основные, субтрактивные цвета (смешивание голубого, пурпурного и желтого тонеров на самом деле дает своего рода серый, а не насыщенный черный цвет). Добавление черного тонера позволяет получить больше цветов и более яркие оттенки черного. Во всех цветных лазерных принтерах цветовая гамма создается путем смешивания тонеров одним из двух основных способов:

1) В принтерах с непрерывным тонированием можно изменять количество тонера для каждого цвета в каждой точке. Эти принтеры довольно дороги, но они также обеспечивают почти фотографическое качество репродукций. Для 32-битного цветного принтера с непрерывным тонированием каждый пиксель может иметь 4 294 967 296 различных комбинаций тонера. Нет необходимости печатать их все, когда, например, сочетание насыщенного черного с C, M и Y выглядит одинаково, и пользователь только теряет тонер. Более того, переход от 24-битной палитры RGB к 32-битной палитре CMYK обычно неочевиден, поэтому пользователь фактически имеет доступ к 16 777 216 цветам.

Не все цвета, которые могут быть представлены на экране в палитре RGB , получаются путем смешивания четырех тонеров. Некоторые из них будут выходить за рамки шкалы и могут быть получены путем замены их на печатный цвет, который воспринимается аналогично. Таким же образом можно будет печатать цвета, которые невозможно воспроизвести на экране RGB .

2) В принтерах с градуированным тонированием количество тонера на цвет в каждом пикселе изменить невозможно. Они относительно дешевле, но качество печати у них значительно ниже, особенно при воспроизведении реальных изображений. Каждый из четырех цветов может присутствовать или отсутствовать в

изображений. Каждый из четырех цветов может присутствовать или отсутствовать в каждом пикселе, поэтому каждая точка может иметь только 16 различных комбинаций тонера. Более того, черный цвет, смешанный с любым другим цветом, будет выглядеть как черный, поэтому 8 из этих комбинаций будут выглядеть одинаково, и каждый пиксель может фактически выглядеть как 9 разных цветов. Цвета, которые не могут быть представлены напрямую, имитируются путем сглаживания, что заставляет человеческий глаз воспринимать промежуточные цвета так же, как полутоновые газетные фотографии. Печать невозможна в 16 777 216 цветах, представленных 24-битным RGB. Можно напечатать только 9 различных цветов, поэтому для отображения 24-битных цветов необходимо использовать процесс дизеринга. Дизеринг позволяет представлять промежуточные цвета способом, сопоставимым с непрерывным тоном. Принтеры с градуированным тонированием могут снабжаться 24-битными данными, что не влияет на возможность смешивать различные количества каждого тонера для получения одинакового цветового диапазона в каждой точке [8].

1.6. Другие типы принтеров

Существуют и другие технологии печати, например, технология построчной печати, применяемая, в частности, в мэйнфреймовых системах. Печать осуществляется с помощью вращающегося цилиндра или цепи, содержащей все символы.

При использовании поворотного цилиндра все столбцы содержат полный набор символов, цилиндр вращается в направлении подачи бумаги, и бумага зажимается между цилиндрами, когда печатаемый символ находится в правильном положении. Один оборот цилиндра печатает целую строку текста.

При использовании вращающейся цепочки набор символов повторяется на цепочке несколько раз. Разработчик принтера определяет количество повторений каждого символа в зависимости от того, как часто он встречается в тексте. В американском английском последовательность выглядит так: ETAON.... Если символы используются часто, то печать всего текста выполняется быстрее. цепь вращается перпендикулярно направлению движения бумаги, и бумага зажимается между лентой и цепью, когда печатаемый символ находится в правильном положении. Вариантом цепного принтера является ленточный принтер. Качество печати аналогично качеству печати цепного принтера, но с меньшим энергопотреблением и меньшим уровнем шума. В стандартном ленточном принтере используется высококачественная цепная передача. На принтерах IBM 3203 и 1403 каждый штрих печатает три символа из строки. Использование цепочки слов является преднамеренным: три символа на

модуле следуют друг за другом во время вращения цепочки.

Рис. С IBM 3203

В ленточном принтере символы располагаются отдельно на пальцах (похоже на символы на ромашке), размещенных на резиновой ленте. Как и в случае с цепным принтером, не все символы повторяются с одинаковой частотой. В отличие от цилиндрических и цепных принтеров, ремень защелкивается между полосками бумаги.

Скорость печати для этих принтеров обычно указывается в lps (строках в секунду) или даже pps (страницах в секунду). Для увеличения скорости печати рекомендуются ленточные принтеры, цепные принтеры и ротационные цилиндрические принтеры (до 70 строк в секунду).

Существуют также термоматричные принтеры, которые позволяют печатать символы и графику (например, штрихкоды) путем нагревания бумаги (точек): нагретые области становятся черными.

В области создания печатных изданий используются и другие экзотические технологии, такие как бумага с алюминиевым покрытием, используемая компанией Sinclair , и технология искрового разряда. Однако они очень редки и в основном устарели [8].

2. Классификация плоттеров в

2.1. Плоттеры для печатных плат

Плоттеры — это устройства с относительно большими габаритами, близкими к максимальному формату чертежа. Например, плоттер формата A1 или A0 выглядел как огромный стол, на котором электростатически были закреплены листы кальки. Маркеры перемещались по стальным рельсам на доске, и рисование даже самого маленького элемента занимало очень много времени, поскольку маркерам приходилось перемещаться по рельсам взад и вперед [11].

2.2. Перьевые плоттеры (новые)

Перьевые плоттеры намного лучше (рис. 4). Эти устройства занимают гораздо меньше места, а лист кальки крепится всего в двух точках.

рисунок с. Д Перьевой плоттер

Принцип работы проще всего представить так: рамка с натянутой на нее калькой движется по роликам в вертикальной плоскости, а каретка с перьями движется в горизонтальной плоскости. Габариты перьевого плоттера пришлось подгонять исключительно под ширину формата. Это были гораздо более компактные, тихие и быстрые устройства.

Скорость работы постоянно увеличивалась за счет программной оптимизации движения каретки пера.

В перьевых графопостроителях не удалось преодолеть следующие недостатки: возможности работы с цветом ограничивались заменой перьев, а заполнение полей сводилось к их штриховке. Рисование растровых изображений было возможно только теоретически. Качество рисунка зависит от ряда факторов, таких как качество чернил и пера, а также скорость движения пера. В случае сложных, многослойных чертежей больших форматов возникали неточности порядка миллиметров, вызванные накоплением механических отклонений подачи пера.

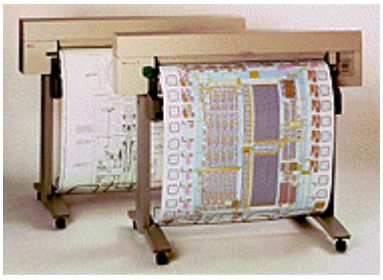
Развитие перьевых плоттеров (лаборатории Houston Instruments) в конечном итоге привело к появлению плоттера со сканирующей головкой. Установка такой головки превращает плоттер в широкоформатное сканирующее устройство [11].

2.3. Струйные плоттеры

Растровые струйные плоттеры разрабатывались параллельно с перьевыми плоттерами. Это устройство работает по принципу распыления капель чернил на кальку, подобно струйному принтеру. Струйные плоттеры гораздо точнее и проще в использовании, а результаты их работы более наглядны.

В настоящее время многие компании выводят на рынок широкоформатные растровые плоттеры. Новейшим продуктом на этом рынке является Hewlett-Packard DesignJet 750C . Это цветной плоттер формата A0 с чрезвычайно высоким

разрешением 600 точек на дюйм в монохромном режиме. Как и большинство современных изделий Hewlett-Packard, это устройство не требует никакого обслуживания. После установки рулона бумаги и картриджей с чернилами на свои места плоттер можно включить. DesignJet (рис. 5) уже сам заботится о том, чтобы все сопла работали исправно, чтобы бумага была правильно позиционирована, проверяет параметры носителя и адаптируется к заданному формату.



чертеж E DesignJet 350 C

Первоначальным вкладом компании Hewlett-Packard в разработку концепции растрового плоттера стало использование так называемого пигментные чернила. До сих пор главным аргументом против струйных плоттеров было явление высыхания сопел. Компания HP позаботилась о том, чтобы устройство самостоятельно проверяло наличие засоров в соплах и пыталось устранить неисправность самостоятельно . Только использование пигментных чернил даст радикальный эффект и будет способствовать улучшению качества печати. Эти чернила, в отличие от других, представляют собой суспензию твердого вещества. Картридж с такими чернилами уже не подлежит заправке, т. н. рефилами .

Эффект использования пигментных чернил заключается в повышении четкости нарисованной линии или поверхности. Слой вещества, нанесенный на бумагу, защищает ее от влаги и повышает долговечность и стойкость рисунка.

Мелкозернистая структура чернил позволяет достичь разрешения около 600 точек на дюйм.

Еще одним интересным решением, примененным в модели 750C , является компенсация люфта в зубчатых передачах каретки. Он устраняет эффект неровных вертикальных линий, который в случае некоторых других устройств, особенно при рисовании крупных элементов тонкими линиями, создает менее эстетичный визуальный эффект.

DesignJet 750C — чрезвычайно тихое и быстрое устройство. Хотя он не может конкурировать с перьевыми плоттерами в плане простых чертежей, он превосходит их в плане сложных: время создания чертежа формата A1 составляет около 4 минут.

Большим удобством является возможность оснащения плоттера так называемым Карты HP JetDirect . Это сетевая карта, благодаря которой плоттер виден в сети как устройство, доступное всем сотрудникам и не требует настройки дополнительного сервера. 8 МБ оперативной памяти , входящие в стандартную комплектацию

серверы с MS операционной системой, выводящие в стандартную комплектацию

DesignJet, позволяют создавать очень большие чертежи (размером даже в несколько мегабайт).

Драйверы спроектированы таким образом, что печать происходит в непрерывном режиме в памяти компьютера, параллельно с обработкой векторов в растровые изображения. Дополнительная память необходима только при расширении системы функциями PostScript [11].

3. Языки управления принтером и плоттером

Современные принтеры способны печатать очень сложные рисунки, порой конкурирующие с фотографиями или документами, полученными из типографии. Способ отправки потока данных с принтера сам по себе является форматом, в котором изображение сохраняется в файле. Существует два основных типа форматов файлов печати для принтеров: расширенные текстовые форматы и языки описания страниц.

3.1. Расширенные текстовые форматы

Такие форматы включают графическую информацию в обычный поток текстовых данных. Печатается обычный текст, а управляющие последовательности вводят нетекстовые элементы. Широко используемым форматом является язык управления принтером (PCL). Он стал стандартом для лазерных принтеров малых и средних объемов печати [6], [15], [16], [8].

3.1.1. IBM Пропринтер

Язык принтера IBM ProPrinter , изначально использовавшийся для принтеров IBM (IBM Graphics Printer 5152, IBM ProPrinter XL 4201/4202, IBM ProPrinter X24/XL24 4207/4208), в настоящее время также используется многими матричными принтерами, такими как Epson ESC/P и ESC/P2 . Функциональность различается в зависимости от версии, используемой или выбранной при настройке принтера (примерами таких версий являются XL, XL24 или AGM).

Язык управления IBM ProPrinter в принтерах разных производителей может немного отличаться по функциональности. Обычно отличия заключаются в дополнительных функциях, не входящих в стандартную комплектацию.

3.1.2. Epson ESC/P, ESC/P2

ESC/P (стандартный код Epson).

ESC/P2 (стандартный код Epson, уровень 2).

Язык принтера ESC/P был создан компанией Epson для использования в первых матричных принтерах. Сегодня он также используется в струйных и лазерных принтерах Epson, а также во многих других матричных принтерах, представленных на рынке. ESC/P2 — это усовершенствованная версия ESC/P , в ней есть, например, новые функции для масштабирования шрифтов, печати растровой графики и т. д.

Принтеры ESC/P или ESC/P2 разных производителей могут немного отличаться по своим функциональным возможностям. Обычно эти отличия представляют собой дополнительные функции, отсутствующие в оригинальной версии Epson .

Информацию об используемых версиях ESC/P и ESC/P2 можно найти в документации к принтерам отдельных производителей. Некоторые пункты представляют собой краткий список доступных функций с примерами на BASIC . Там же вы найдете таблицы с наборами символов и таблицы с шириной шрифтов. Справочное руководство ESC/P2 компании Epson содержит список кодов ESC/P и ESC/P2 , а также полное описание различий в каждой команде принтера. Последняя версия датирована августом 1992 года.

3.2. Языки описания страниц

Другой подход к управлению принтером заключается в определении совершенно нового языка описания печати. В прошлом использовалось несколько таких языков, но в настоящее время стандартом стал PostScript [8], [6].

3.2.1. Постскриптум

PostScript — язык описания страниц, разработанный Adobe Systems Inc. с начала 80-х годов. Компания Adobe была основана в 1982 году доктором Джоном Э. Уорноком и доктором Чарльзом М. Гешке. В нем содержатся инструкции по описанию информационной страницы. Поскольку ему требовалось больше памяти, чем большинству языков описания страниц, он стал первым широкодоступным продуктом, способным управлять большим количеством шрифтов и графики.

Первая версия, опубликованная в 1985 году, называется Level I, текущее усовершенствование называется Level II (не путать с PostScript версии 47.0 или 2011.110, а также с номером в начале любого вывода PostScript — например, %!PS

2011.110, а также с номером в начале любого вывода PostScript , например %!PS-Adobe-3.0). Уровень PostScript и расширение версии с интерпретатором расширяют возможные операции.

В обращении находится несколько клонов PostScript , но из-за стоимости лицензии интерпретатора Adobe ; Самый известный — GhostScript . Другие, встроенные непосредственно в лазерные принтеры или адаптированные с помощью дополнительного картриджа, включают: Phoenix Page, BrotherScript , Page Styler, True Image, Turbo PS, PDL и KPDL . Говорят, что они полностью совместимы с PostScript, но эта совместимость иногда нарушается при загрузке шрифта, при манипулировании шрифтом (например, при вставке метрической таблицы или новых символов) и во время других операций. Однако при использовании клонов не возникает проблем с печатью простого текста или графики.

Для написания программы PostScript используется набор символов PostScript , а не символы печатного шрифта PostScript.

Adobe рекомендует использовать в PostScript только печатаемое подмножество символов ASCII : пробелы, табуляции, а также символы CR и LF . PostScript не запрещает использование символов, выходящих за рамки этого набора, но их использование может вызвать проблемы при передаче (например, передача 8-битных символов по 7-битной последовательной линии на принтер). Для представления 8-битных символов вне строки используйте формулу \ddd .

Файл принтера на самом деле представляет собой компьютерную программу PostScript , которая рисует изображение. PostScript используется для выполнения операций, тесно связанных с генерацией изображений.

Программы PostScript рисуют графику двумя основными способами. Более простой способ — нарисовать пиксельную карту. Один из основных операторов PostScript считывает последовательность пикселей и отображает их в прямоугольной области на странице. Такой подход подходит для прикрепления к документам изображений, отсканированных или снятых с экрана монитора, а также любых других изображений, которые уже существуют в растровом формате.

Альтернативным способом использования PostScript является создание рисунка в векторном формате или формате метафайла. Существуют операторы для рисования линий, окружностей, кривых, прямоугольников и т. д., и из этих элементов можно строить графику.

Вы также можете объединить эти два подхода. Например, операторы линии, прямоугольника и кривой можно использовать для определения области, которая затем будет использоваться в качестве маски обрезки для управления отображением

затем будет использоваться в качестве маски обрезки для управления отображением растрового изображения [8], [11].

3.2.2. HP PCL и PJL

Язык PCL был создан компанией Hewlett-Packard для нужд ее принтеров (лазерных и струйных). Версии языка PCL нумеруются от 1 до текущей версии 5e.

Краткая история PCL (на основе Технического справочного руководства по языку печати HP):

PCL 1 — Функциональность печати и пространства является основой функций, предоставляемых для простой и удобной однопользовательской рабочей станции.

Функциональность PCL 2 - EDP (электронная обработка данных) является улучшением по сравнению с PCL 1 . Добавлены функции общего назначения и многопользовательская система печати.

PCL 3 — функциональность офисного текстового процессора является улучшением по сравнению с PCL 2. Также были добавлены функции для улучшения качества печати и обработки офисных документов (принтеры семейства HP DeskJet) .

PCL 4 — форматирование страниц является улучшением по сравнению с PCL 3 . Добавлены новые возможности печати страниц (принтеры: HP LaserJet, HP LaserJet IIP (PCL 4,5))

PCL 5 — функциональность Office Publishing представляет собой улучшение по сравнению с PCL 4 . Новые возможности публикации включают масштабирование шрифтов и графики HP-GL/2 (принтеры HP LaserJet III, HP LaserJet 4 (PCL 5e)).

Версии PCL различаются по функциональности (например, типы шрифтов, растровые шрифты, масштабируемые шрифты (Intellifonts, True Type), методы сжатия графики, поддержка графики принтером HP LaserJet III).

PCL — наиболее распространенный язык печати на современном рынке лазерных принтеров. Большинство производителей лазерных принтеров используют для своих принтеров PCL 4 или PCL 5.

Язык заданий принтера (PJL) также был создан компанией HP для предоставления метода изменения уровня задания и считывания параметров состояния между принтером и хост-компьютером. PJL можно использовать в начале печати для установки некоторых параметров, таких как язык принтера (PCL, PostScript или другие), разрешение (300 или 600 точек на дюйм), количество копий и т. д.

В настоящее время PJL используется в следующих принтерах HP : LaserJet IIISi ,

семейство LaserJet 4, PrintJet XL 300 и DesignJet.

PJL также используется в принтерах серии LaserJet 5 [11], [7], [10].

3.3. Другие языки управления принтером

На рынке существует множество других уникальных языков управления принтером. Поэтому следующий список не является полным (мы лишь упомянули их, а не описали подробно). Порядок, в котором перечислены языки, не имеет ничего общего с их важностью на рынке.

Расширенная функция печати (AFP): используется в мэйнфреймах IBM для страничных принтеров. Это функция представления набора архитектуры содержимого смешанных объектно-документных документов (MO:DCA), который является частью архитектуры системных приложений IBM .

На самом деле вы не печатаете с помощью MO:DCA , вы используете IPDS (Intelligent Printer Data Stream).

Информация включает в себя PTOCA (архитектура содержимого печатных текстовых объектов), GOCA (архитектура содержимого графических объектов), IOCA (архитектура содержимого изображений) и другие.

IPDS — это язык печати IBM SAA . Он работает с различными растровыми шрифтами, с простой базовой графикой и с растровыми изображениями. Благодаря простоте представленной модели ее можно использовать для управления и эксплуатации высокоскоростных лазерных принтеров.

Diablo 630: Первоначально использовался с розеточными принтерами и пишущими машинками. Он допускает только последовательности табуляции, интервалы между строками и символами, выбор атрибутов (жирный, двойной штрих, подчеркивание), горизонтальные перемещения в обоих направлениях, пропорциональные интервалы, а также автоматическое центрирование и выравнивание и т. д. Этот язык иногда используется другими производителями в качестве основы для их специальных эмуляций.

CaPSL : (язык системы печати Canon) — предыдущий стандартный язык для лазерных принтеров Canon . Другое название, которое встречается, — LIPS (система лазерной визуализации). У CaPSL есть своя история : Canon производит принтерные двигатели для HP , но не имеет лицензии на использование HP PCL в своих собственных принтерах. Отсюда возникла необходимость найти собственный язык принтера. Лазеры Canon традиционно включают эмуляции CaPSL, IBM ProPrinter, ESC/P и PostScript , но не PCL). Эта

часть контракта между Canon и HP , по-видимому, истекла, поскольку теперь Canon предлагает принтеры с PCL 4 и PCL 5.

LIPS поддерживает Diablo 630 (заводская настройка для командного режима), режим ISO (для печати текста и растровой графики) и режим VDM (для векторной графики и печати символов).

RENO : Это стандартный язык управления для принтеров Agfa (P400, P3400 и т. д.). RENO — это тип языка описания страниц. Его функциональность огромна: помимо печати текста с использованием различных масштабов шрифта, вы можете рисовать линии, заполнять значки (Windows) узорами, использовать программные выражения (if-then-else, repeat-until, set, use and print variables, push и pop), вы можете загружать и печатать собственные символы и переносить данные в оперативную память принтера или на жесткий диск или дискету, если таковая подключена.

Prescribe : Это язык описания страниц, созданный Kyocera . Его преимуществом является тот факт, что его можно встроить в другую текущую эмуляцию принтера на устройствах Kyocera . Принтеры Kyocera поддерживают HP PCL , клон HP-GL под названием KC-GL , Epson ESC/P (режим LQ-850), IBM ProPrinter X24E, Diablo 630 , универсальную эмуляцию линейного принтера, и в качестве опции KPDL , клон PostScript.

DEC: DEC имеет свои собственные уникальные языки для лазерных принтеров (LN03, LN06).

ANSI : Дэн Макгоуэн из Mannesmann Tally утверждает, что: Принтеры Mannesmann Tally , имеющие сертификат ANSI, послужили основой для нотации ANSI 3.64 . Это довольно свободная спецификация, охватывающая общие периферийные функции. Большинство принтеров, произведенных в США, имеют все команды ANSI 3.64, относящиеся к функциям принтера. Принтеры, произведенные в Германии, представляют собой серию принтеров с летающей последовательной головкой. У них есть MTPL (Mannesmann Tally Printer Language), который основан на ANSI 3.64, но содержит дополнительные команды [8].

3.4. Языки управления плоттером

Название HP-GL (Hewlett -Packard Graphics Languages) относится к языку, который изначально использовался для управления плоттерами Hewlett -Packard . Датой создания этого языка принято считать 1976 год — год появления на рынке

вышеупомянутых плоттеров. По мере совершенствования этих устройств язык HP-GL обогатился новыми командами, и его вторая версия теперь обозначается как HP-GL/2.

Язык HP-GL также поддерживает векторные шрифты, что на практике означает возможности управления устройствами, аналогичные тем, которые предоставляет PostScript . Большая часть языка HP-GL/2 была включена в пятый уровень языка PCL , который содержит специальные команды, переключающие управление с классического PCL на HP-GL и наоборот.

Синтаксис языка HP-GL/2 прост. Все инструкции представляют собой двухбуквенные сокращения их названий. После ярлыка идут параметры. Они могут быть обязательными или необязательными. В качестве разделителя между параметрами можно использовать пробел или запятую. Предпочтительным разделителем является запятая. Каждая инструкция заканчивается терминатором. Терминаторами могут быть: точка с запятой, первый символ следующей инструкции или пробел [10], [7], [6].

Системы перевода

1. Преобразование между типами файлов

Поскольку существует множество форматов языка управления принтером, само собой разумеется, что часто возникает необходимость конвертировать один формат файла печати в другой. В зависимости от типа исходного и целевого форматов это может оказаться простой задачей или даже невыполнимой. Изображения, сохраненные в файле печати, могут иметь форму . Это может быть растровое изображение, векторное изображение или даже простой текст ASCII . Учитывая формат языка, понятный принтеру, может возникнуть необходимость преобразовать формат изображения, сохраненный в файле печати [6].

1.1. Растровое изображение в растровое изображение

Преобразование одного типа растрового изображения в другой обычно не вызывает затруднений. Как только мы углубимся в детали кодирования файла, мы будем иметь дело с пикселями, которые почти всегда одинаковы, поэтому преобразование будет простым.

Если необходимо преобразовать изображение из более описательного формата в менее описательный, например, из цветного в черно-белый или оттенки серого, то существуют известные методы преобразования с сохранением наилучшего возможного качества изображения.

1.2. Вектор в векторный формат

Основная проблема при конвертации между векторными форматами — это учет несколько различной семантики отдельных форматов, а также, в некоторой степени, системы координат. В простейшем случае команды преобразуются один в один, например, команда нарисовать линию в такую же команду в выходном файле. Проблемы возникают, когда два формата не имеют соответствующих команд. Если исходный формат имеет команду для рисования эллипса, а целевой формат — нет, то необходимо использовать один из доступных методов перевода. Вы можете аппроксимировать эллипс окружностью или многоугольником, составленным из

коротких сегментов. Если формат назначения допускает раздельное масштабирование осей x и y, хорошим методом может быть установка на них разных единиц и рисование окружности, которая будет масштабирована в эллипс.

1.3. Векторный формат в растровый

Обработка векторного изображения в растровое называется растеризацией. Он включает в себя нахождение набора пикселей, соответствующих каждому вектору исходного изображения. Базовый алгоритм растеризации известен с 1963 года, когда он был опубликован Брезенхэмом ; Круги и дуги можно рисовать тем же методом.

Этот тип преобразования решает те же проблемы, что и отрисовка векторного изображения на растровом экране или лазерном принтере, поэтому часто можно адаптировать код отображения для преобразования в растровое изображение.

1.4. Растровый формат в векторный

Преобразование растрового изображения в векторное намного сложнее, чем любое из предыдущих преобразований. В настоящее время существуют удовлетворительные алгоритмы обнаружения границ. Их можно использовать для поиска линий в растровом изображении, но только в самых простых случаях. Проблема поиска линий в отсканированном изображении (например, в отправленном по факсу документе) остается нерешенной.

2. Конвертация программ

2.1. Замена PostScript другими стандартами

Самая известная программа для интерпретации PostScript — GhostScript . Вызвав GS с опцией -help, вы можете получить информацию о доступных устройствах. GhostScript может эмулировать следующие системы: epson, epsonc, necp6, laserjet, ljetplus, ejet2p, ljet3, paintjet , bj10e, djet500, djet500c, pjetxl, lbp8 (эти названия используются в программе). Также возможно добавлять новые драйверы в программу в [8].

2.2. Изменение других стандартов на PostScript

В случае текста ASCII переход от стандарта к PostScript не представляет

сложности. Для этой цели существует несколько общедоступных инструментов. Самая простая из них — программа a2ps.

Для текста, отличного от ASCII (ISO 8859-1 или PC 437), или последовательностей управления, специфичных для принтера, преобразование может быть затруднено. Для преобразования HP PCL в PostScript используйте утилиту lj2ps , которая очень полезна в случае непропорциональных шрифтов (например, для HP LaserJet II). Однако проблема усложняется при конвертации графики. Существует также конвертер для HPGL (hpgl2ps), но он встречается редко. Чтобы изменить Epson на PostScript, используйте фильтр epson2ps [8].

Приложение Эмулятор

Программа EmuLator предназначена для сред Windows, начиная с версии 3.1. Выбор данной системы был обусловлен прежде всего простотой ее использования и возможностью общения с пользователем посредством насыщенного графического интерфейса. Простота использования программы стала возможной благодаря сложности программирования с использованием API (Application Programming Interface) — набора функций, которые программисты должны использовать при написании приложений для Windows. Windows API содержит более 600 функций. Помимо вышеупомянутых неудобств, есть некоторые преимущества программирования для Windows [13]:

Windows обеспечивает независимость от оборудования. Одна и та же программа может отображать информацию на разных мониторах (EGA, VGA и т. д.) и печатать на разных принтерах: от матричных до лазерных.

С точки зрения программиста, Windows предоставляет множество готовых элементов пользовательского интерфейса, таких как экранные кнопки, меню, диалоговые окна, списки и поля редактирования.

Windows включает в себя расширенный интерфейс графических устройств (GDI) для отображения текста и графики. В частности, этот интерфейс позволяет вам рисовать в собственной системе координат.

В качестве языка программирования был выбран C++ для новейшего на момент создания программы компилятора Borland 4.52 [4]. Этот инструмент позволяет реализовать объектно-ориентированный подход к проектируемому приложению [2] и позволяет использовать библиотеки классов OWL и CLASSLIB [1].

Основная цель библиотеки классов OWL (Object Windows Library) — предоставить программистам полную среду для приложений Windows. Был использован OWL 2.5 компании Borland , что позволило быстро создать пользовательский интерфейс с привлекательными графическими элементами управления.

Библиотека контейнеров CLASSLIB компании Borland предоставляет выходные объекты для хранения и обработки данных.

1. Описание программы

Программа предназначена для решения задачи интерпретации графического изображения, содержащегося в печатном файле, на экране компьютера. Большое количество языков управления принтерами и плоттерами и различные формы их записи подсказывают программисту интерпретировать выбранный язык принтера. Такой подход можно наблюдать, например, в программе GhostScript , интерпретирующей язык PostScript . Эта работа включает в себя написание программы, которая будет интерпретировать различные языки управления.

Предполагается, что файл печати содержит различные команды, вызывающие графические действия, такие как: печать текста или рисование графики, а также последовательности, изменяющие параметры принтера. Результаты можно увидеть на распечатанном листе бумаги, а также представить на экране компьютера. Отдельные языки управления вызывают одно и то же действие, используя разные комбинации кодов. Для имитации работы принтера или плоттера программа содержит набор основных операций, выполняемых этими устройствами, и позволяет выполнять эти операции.

Существенной проблемой было использование программой любого языка принтера таким образом, чтобы это было удобно для пользователя и в то же время позволяло добавлять новые языки управления в любое время. Эмулятор открывает драйвер принтера, представляющий собой текстовый файл в указанном формате, и проверяет его на наличие ошибок записи. Этот файл также содержит начальные настройки для данного принтера или плоттера. Пользователь может самостоятельно создать драйвер для каждого принтера, используя некоторые операции, содержащиеся в программе, без необходимости перекомпиляции программы. Если данный принтер использует специфические механизмы, не включенные в обсуждаемую программу, опытный программист может быстро расширить программу новыми функциями.

Приложение использует механизмы MDI (Multiple Document Interface), которые позволяют просматривать несколько документов одновременно. Программа позволяет интерпретировать распечатки только одного типа принтера одновременно.

Окончательный результат можно просмотреть на экране в масштабе 1:1 (с учетом погрешностей экрана монитора), в полностраничном предварительном просмотре или распечатать на любом принтере. Используя аппаратную независимость контекста устройства GDI , программа позволяет преобразовать известный формат принтера или плоттера в формат любого устройства, доступного в Windows.

1.1. Строительство

Приложение создано на основе архитектуры MVC (Model-View-Controller), широко используемой в языке Smalltalk-80, с учетом специфических возможностей библиотеки OWL 2.5 . Архитектура MVC делит приложение на три уровня:

1. Модель означает прикладной уровень, в котором размещены все объекты, зависящие от приложения. В EmuLator все это объекты, представляющие слои данных и операции, выполняемые над ними. В частности, это может быть программируемый словарь, позволяющий переводить код в файле печати в графические операции, и класс, обрабатывающий данные файла печати в графическое изображение, используемое на уровне представления.
2. Вид — это так называемый уровень представления, который отвечает за представление данных. Уровень представления считывает соответствующую информацию с уровня приложения и отображает ее на экране. Этот уровень также имеет дело с окнами графического пользовательского интерфейса.
3. Контроллер - это так называемый уровень управления, который обеспечивает передачу информации от устройств ввода (клавиатура, мышь) к двум другим уровням.

Эта модель претерпевает незначительные изменения в ходе реализации и подчиняется новым методам, найденным в OWL 2.5, таким как Document/View , в зависимости от решаемой задачи.

1.1.1. Уровень управления

1.1.1.1. Основной класс применения

class emuApp: public TApplication — важнейший класс приложения, реализацию которого можно найти в файлах emuapp.cpp и emuapp.h , где *также определен* вспомогательный класс который занимается обработкой документов, загруженных методом *перетаскивания* . Класс *emuApp* является базовым для всех объектов в приложении. В *функции InitMainWindow* создаются объекты *классов emuMDIFrame и emuMDIClient ,* графические элементы программы (такие как иконки и меню) загружаются ресурсами, строка состояния (*TStatusBar*) и панель инструментов (*TControlBar) вставляются в рамку emuMDIFrame* , устанавливаются атрибуты главного окна и включаются пространственные элементы диалога с помощью *функции EnableCtl3d(true)* . Помимо обработки стандартных сообщений Windows, *класс emuApp* отвечает за следующие задачи для рассматриваемого приложения:

класс emuApp отвечает за следующие задачи для рассматриваемого приложения:

поддержка справочной системы - загрузка файла справки в обычном или контекстном режиме;

создание нового документа и представления - при открытии нового документа и создании представления пользователь информируется, если файл драйвера не загружен; при загрузке нового файла драйвера все окна закрываются;

передача управляющего сообщения для представления документа файла печати (*emuView*) - класс *emuApp* передает запрос текущему представлению *emuView* для перехода на следующую или предыдущую страницу, сообщая о ее текущем номере на панели инструментов;

обновление информации в строке состояния - класс *emuApp* информирует о текущем загруженном драйвере принтера или его отсутствии, а при выборе команды из панели инструментов или меню отображается подсказка.

1.1.1.2. Классы управления приложениями MDI

класс emuMDIFrame: public TDecoratedMDIFrame (emmdifrm.cpp, emmdifrm.h) — содержит видимые элементы управления и обрабатывает все управляющие сообщения, предназначенные для *дочерних окон emuMDIChild* , направляя их в *emuMDIClient*.

класс emuMDIClient: public TMDIClient (mmdiclnt.cpp, mmdiclnt.h) — отвечает за управление *объектами окна emuMDIChild* в ответ на запросы из меню «Окно».

класс emuMDIChild: public TMDIChild (mmdichld.cpp, mmdichld.h) — определяет базовое поведение окна MDI , содержащего указатель на *объект TWindow* , который является видимым элементом представления документа.

1.1.1.3. Класс контроля документов в

класс emuDocManager: public TDocManager (emdcmngr.cpp, emdcmngr.h) - объект, управляющий списком текущих зарегистрированных шаблонов. *Макрос DEFINE_DOC_TEMPLATE_CLASS* создает шаблон, связывающий объекты документа с представлениями, на основе которых *TDocManager* обрабатывает стандартные *команды меню Файл* , передавая их соответствующим документам. Он также отвечает за отображение элементов пользовательского интерфейса для выбора и просмотра файлов.

1.1.2. Уровень

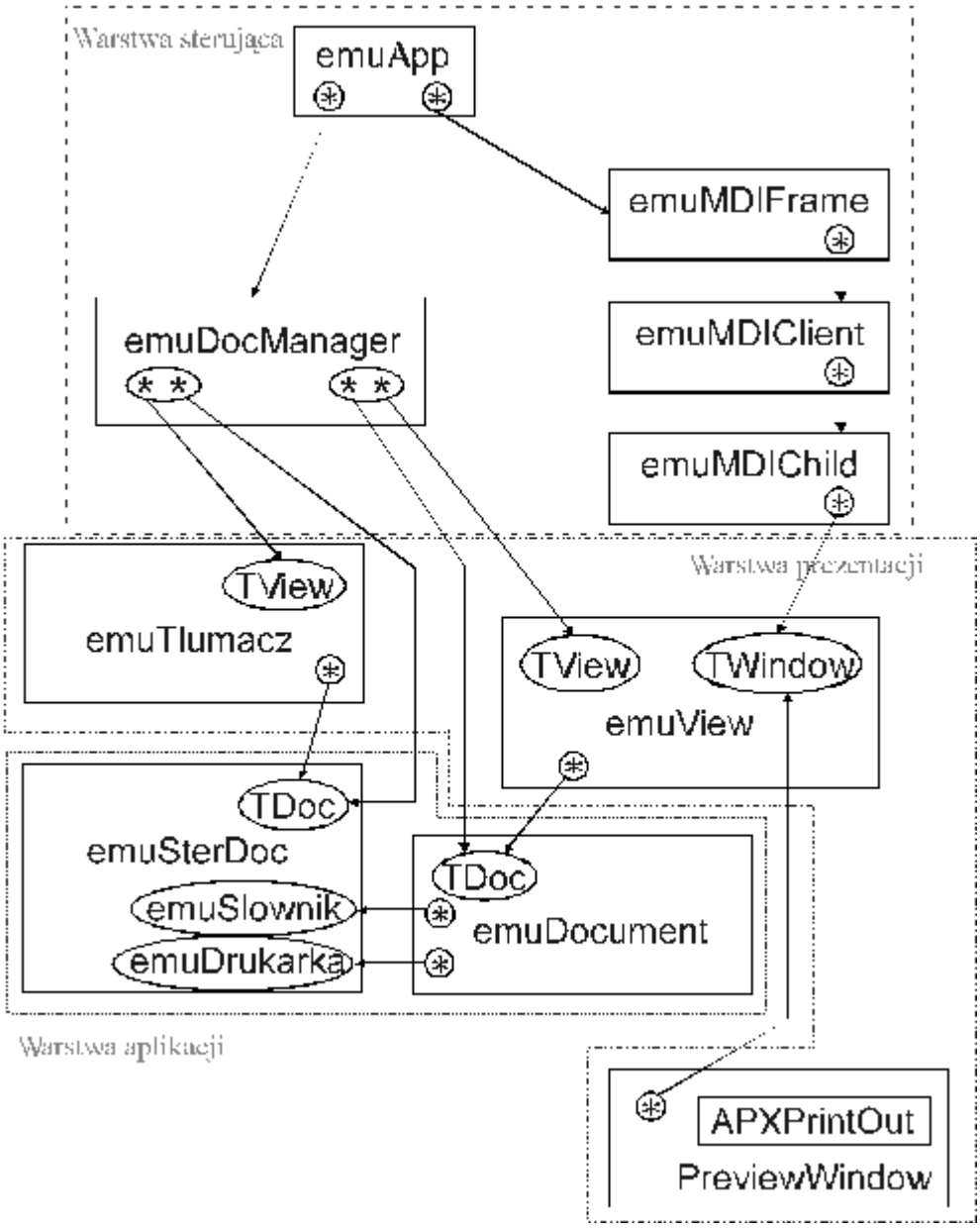


Рис. F Иерархия основных классов приложений

представления

1.1.2.1. *Класс представления данных драйвера принтера*

класс emuTluczacz: public TView (emutlmcz.cpp, emutlmcz.h) — объект, выступающий посредником при доступе к данным, содержащимся в документе. *emuTluacz* позволяет получить указатель на класс словаря (*emuSłownik*), созданный в объекте документа драйвера принтера (*emuSterDoc*). Являясь единственным представлением в приложении, *emuTlumacz* не имеет связанного окна и

обрабатывается особым образом в основном объекте приложения. Для каждого

объекта документа файла печати (*emuDocument*) *emuApp* устанавливает указатель на *emuTranslator* (в один момент времени может существовать только один объект *emuTranslator*).

1.1.2.2. *Файл классов представления данных в печати в*

класс emuView: public TWindowView (emuview.cpp, emuview.h) — объект представления документа файла печати, наследующий свойства двух классов OWL . Являясь потомком *TView,* он поддерживает доступ к *данным emuDocument* . Наследование от *TWindow* — графическое представление документа в виде видимого окна, указатель на которое передается объекту *emuMDIChild* . В ответ на запрос на отображение следующей страницы *emuView* передает *объекту emuDocument* контекст отображения для рисования изображения печатной страницы. Для того чтобы каждая реакция представления на *сообщение WM_PAINT* не приводила к запуску всего механизма трансляции, интерпретации и создания рисунка из файла печати, используется контекст метафайла, хранящийся в виде объекта класса *TMetaFilePict* .

класс APXPrintOut: public TPrintout (apxprint.cpp, apxprint.h) — представляет физический документ, отправленный на принтер. Объект этого класса отвечает за отрисовку страницы на физическом принтере или в окне предварительного просмотра печати.

класс PreviewWindow: public TDecoratedFrame (apxprev.cpp, apxprev.h) — создает рамку предварительного просмотра печати и поддерживает отображение одной или двух страниц одновременно в *объекте TLayoutWindow* . Он также отвечает за печать на физическом принтере в результате выбора *команды «Печать»* приложения .

1.1.2.3. Другие классы

класс emuEditView : public TEditView (emuedtvw.cpp, emuedtvw.h) - класс представления, связанный с любым типом файла (непосредственно *TFileDocument*); позволяет выполнять базовые операции редактирования, печати текста и поиска.

класс TBmpViewWindow (emuabout.cpp, emuabout.h) — используется для представления начальной заставки программы в виде растрового изображения.

класс emuAboutDlg: public TDialog (emabtdlg.cpp, emabtdlg.h) — с помощью *вспомогательного класса ProjectRCVersion* , который предоставляет информацию о проекте, отображает диалоговое окно о программе.

1.1.3. Уровень приложений

1.1.3.1. Классы документов драйвера принтера

класс emuSterDoc: public TFileDocument (emustrdc.cpp, emustrdc.h) — представляет объект данных документа и предоставляет способ их интерпретации в представлении. Содержит ряд методов для манипулирования физическим файлом на диске, включая поддержку потоков. Каждый документ может быть связан с несколькими представлениями, поэтому он поддерживает связь с ними, создавая список текущих представлений и отправляя сообщения о любых изменениях в них. Документы и представления имеют список свойств (*Свойства*), которые приложение использует для принятия решения о том, как обрабатывать данные.Среди свойств *emuSterDoc есть указатели на объекты emuDictionary* и emuPrinter *,* которые он создает при открытии документа драйвера принтера .

class emuDictionary: public ContainerType (emuslwnk.cpp, emuslwnk.h) — этот объект является производным от класса словаря контейнера, созданного с помощью *макроса typedef TDictionaryAsHashTable<AssociationType> ContainerType;* представляет собой набор ассоциаций, а также методы их добавления и поиска. Ассоциация (*typedef TIIAssociation<emuMyClass, GraphicalObject> AssociationType;*) представляет собой пару указателей на объекты в *emuMyClass* и *GraphicalObject* . Объект класса *emuDocument* , найдя управляющий код, выполняет графическую операцию, производную от *GraphicalObject* . Когда *emuSterDoc* создает словарь , динамически создаются и добавляются только те графические объекты, которые необходимы для интерпретации данным устройством.

класс emuMyClass (emumycla.cpp, emumycla.h) — отдельный код принтера или плоттера, а также оператор сравнения для его поиска.

класс GraphicalObject (emugrobj.cpp, emugrobj.h) - базовый класс для графических операций в контексте устройства, из которого виртуально выводятся объекты, соответствующие отдельным управляющим кодам. Объект этого класса не выполняет никаких операций, но в диагностических целях выводит имя операции, которая должна быть выполнена.

class GraphObjHPGL: public virtual GraphicalObject (emughpgl.cpp, emughpgl.h) - базовый класс для графических операций, специфичных для плоттера. Классы для конкретных графических операций являются производными от него: *class HPGL_0xXXXX: public virtual GraphObjHPGL* , где *0xXXXX* - уникальный внутренний номер приложения для данной операции.

класс GraphObjPCL: публичный виртуальный GraphicalObject (emugopcl.cpp, emugopcl.h) — базовый класс для графических операций, специфичных для струйных и лазерных принтеров. Он является источником классов для определенных графических операций: *класс PCL_0xXXXX: публичный виртуальный GraphObjPCL* , где *0xXXXX* — уникальный внутренний номер приложения для данной операции.

класс GraphObjIBMPro: публичный виртуальный GraphicalObject (файл emugopro.cpp, emugopro.h) — базовый класс для графических операций, специфичных для матричных принтеров. Он является источником классов для определенных графических операций: *класс IBMPro_0xXXXX: публичный виртуальный GraphObjIBMPro* , где *0xXXXX* — уникальный внутренний номер приложения для данной операции.

класс emuPrinter (emudrkrk.cpp, emudrkrk.h) — класс, содержащий начальные настройки эмулируемого устройства, также используется при создании графического изображения.

1.1.3.2. Классы документов для печати файлов

класс emuDocument: public TFileDocument (emudcmnt.cpp, emudcmnt.h) — поддерживает операции, связанные с физическим файлом печати. По запросу *emuView* анализирует файл, ищет коды управления в *emuDictionary* и выполняет графические операции с виртуальной *функцией Draw()* объекта, производного от *GraphicalObject* .

класс emuStrona (emustron.cpp, emustron.h) — класс, поддерживающий создание рисунка определенной страницы. *emuDocument* использует массив страниц *(typedef TIArrayAsVector<emuPage> emuPages;*) для запоминания позиции потока файла печати и настроек принтера (*emuPrinter*) для каждой страницы.

1.2. Разработка

Программа предоставляет прекрасную основу для расширения новыми функциями и модулями. Используя технику OWL Document/View , можно создавать другие типы представлений и документов для существующих или новых объектов. Также можно легко расширить модули, интерпретирующие файлы печати. При расширении программы вы найдете полезные комментарии в исходных файлах и вспомогательных классах, которые не были встроены в финальную версию программы.

1.2.1. Добавление объекта в представление в

Чтобы добавить объект представления, просто определите новый класс, производный от *TView* , который использует методы, содержащиеся в выбранном объекте документа. Если в представлении презентации будут использоваться специальные формы данных, их необходимо определить в объекте документа. Объект представления может быть связан с документом, имеющим несколько различных способов представления. Чтобы программа поддерживала их, необходимо определить соответствующие шаблоны для *emuDocManager* , например:

```
DEFINE_DOC_TEMPLATE_CLASS(emuSterDoc, TSterListView, DocType7);
DocType7 __dvt7("Пользователь-просмотрщик", "*.emu", 0, "EMU",
dtAutoDelete);
```

где: *emuSterDoc* — существующий класс документа, а *TSterListView* — класс представления контроллера. Представление может выглядеть как окно, отображаемое на экране, или в любой другой форме интерпретации данных, содержащихся в документе.

1.2.2. Добавление объекта в документ в

Каждый объект документа отвечает за загрузку данных и их корректную доставку в представления. Если документ имеет несколько представлений и одно из них допускает изменение данных, необходимо также обрабатывать сообщения, передающие информацию об этом факте в другие представления. Связь между представлениями и документами может осуществляться посредством сообщений, списков свойств (*Prosperities*) или прямых ссылок-указателей. Дополнительную информацию о взаимодействии документов с представлениями см. в документации компилятора Borland C++ 4.52 .

1.2.3. Расширение графических функций

В программе EmuLator при интерпретации файла печати используется следующая функция:

```
int emuDocument::Draw( TDC& strDC, emuPage* страница )
```

где: *strDC* — контекст устройства, в котором должно быть создано изображение; *page* — это указатель на объект, содержащий текущие настройки страницы и копию настроек принтера (*emuPrinter*) с ранее созданной страницы или (в случае первой страницы) настройки устройства, считанные *emuSterDoc* . Если в потоке документа обнаружен управляющий код, объект emuDictionary *возвращает* указатель на

ассоциацию, из которой получается указатель на соответствующий графический объект. Операция выполняется путем вызова виртуальной функции объекта, производного от *GraphicalObject* :

```
если (найдено)
{
если (объект = найден->Значение())
объект -> Draw(strDC, поток, страница);
```

Функция *void GraphicalObject::Draw(TDC& dc, TInStream* is, emuStrona* str)* для выполнения задачи вызывается со следующими параметрами:

TDC& dc - ссылка на контекст устройства, на котором должна быть выполнена графическая операция;

*TInStream** — указатель на поток файла печати, из которого можно извлечь дополнительные данные для выполнения операции;

emuPage str* - указатель на объект, в котором можно узнать текущие настройки эмулируемого устройства (*emuPrinter *prints*) и, в случае ошибки или конца страницы, следует установить одно из значений *переменной status* :

```
перечисление {
Ошибка=0,
Конечная страница,
КонецФайла,
Следующий,
};
```

Чтобы расширить программу новыми графическими функциями, вам нужно создать объект, полученный виртуально, напрямую или косвенно (например, как объекты *GraphObjHPGL*) из класса *GraphicalObject* , который выполняет операцию с использованием некоторых из приведенных выше параметров. Затем вам нужно выбрать уникальный внутренний номер приложения для графической операции (из диапазона 0x0000 до 0xFFFF) и поместить операцию добавления объекта в emuslwnk.cpp:

```
инт
emuDictionary::AddItem(константная строка& mc, константа беззнакового
длинного целого числа & mv, константа строка& mvs)
{
целочисленный результат=0;
переключатель (мв) {
случай 0xXXXX: {
```

```
случай 0xXXXXX: {
Тип Ассоциации ассоц( новый emuMyClass(mc),
новый ГрафическийОбъект(mvs) );
результат=Добавить(ассоц.);
перерыв;
}
```

Эта функция вызывается *emuSterDoc* при чтении файла драйвера устройства со следующими параметрами:

mc - управляющий код эмулируемого устройства;

mv - внутренний код приложения для графических операций;

mvs - имя операции, используемое в диагностических целях, берется из файла драйвера.

1.2.4. Расширение свойств эмулируемых устройств

Если для графической операции требуются параметры, которые не могут быть найдены в определении *emuPrinter* (файл emudrkrk.h), их следует добавить в этот класс. Инициализация значения происходит в *bool emuSterDoc::GetPrinter(TInStream* is),* куда следует добавить процедуру чтения параметров, например:

```
если (key.contains("РАЗМЕР СТРАНИЦЫ")) // Размер страницы
{
строка += ПропуститьВсе(*is);
длинный x = GetLong(*is);
строка++;
если (!is->good()) перейти к ошибке;
строка += ПропуститьВсе(*is);
длинный y = GetLong(*is);
строка++;
если (!is->good()) перейти к ошибке;
принтер->SetRStr(x*(метрика ? мм_пакет : каль_пакет),
y*(метрическая система ? мм_точки : дюйм_точки));
}
еще
```

key — это ключевое слово, встречающееся в файле драйвера принтера и имеющее следующее обозначение:

```
!Размер страницы
```

Затем из текстового потока считываются все параметры * для них выбирается соответствующая функция. Функция *SkipAll(*is)* пропускает все символы в потоке,

пока не встретит символ ! в начале строки. Переменная *row* указывает номер строки, в которой произошла ошибка. Конструкция *x*(metric ? mm_pkt : cal_pkt)* позволяет задавать параметры в предопределенных единицах (в настоящее время мм или дюйм).

1.2.5. Вспомогательные классы

При запуске проекта вы можете использовать диагностические классы, включенные в исходный код. Для этого нужно включить включение кода для отладчика в параметрах проекта и изменить комментарии в файле emuapp.cpp, активировав шаблоны для этих представлений. Это следующие классы:

класс TDumpView: public TListBox, public TView (dumpview.cpp, dumpview.h) — позволяет связываться с любым типом документа и отображать именно его содержимое с шестнадцатеричными кодами отдельных байтов в файле;

класс TInfoView : public TWindowView (infoview.cpp, infoview.h) - позволяет просматривать список свойств (*Prosperities*) любого документа;

class TSterListView : public TListBox, public TView (strlisvw.cpp, strlisvw.h) — представление, специально созданное для нужд *класса emuSterDoc* document , отображающее содержимое словаря, инициализированного файлом драйвера принтера.

Поддержка программы

Инструкция по применению

Перед использованием программы EmuLator ее необходимо установить на жесткий диск вашего компьютера. Установочный пакет находится на дискетах HD 3,5". Процесс установки типичен для среды Windows .

После установки программы EmuLator в окне диспетчера программ появляется собственная группа под названием Printer and Plotter EmuLator . Основная иконка приложения называется EmuLator .

После запуска приложения на экране появляется окно приложения под названием EmuLator (рис. 7).

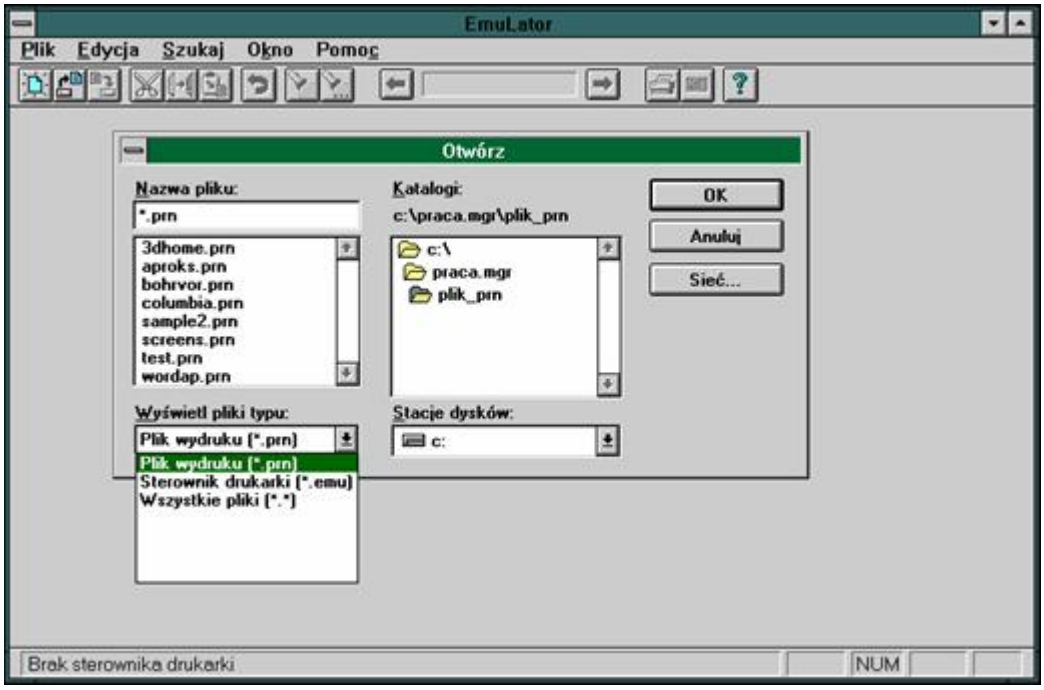
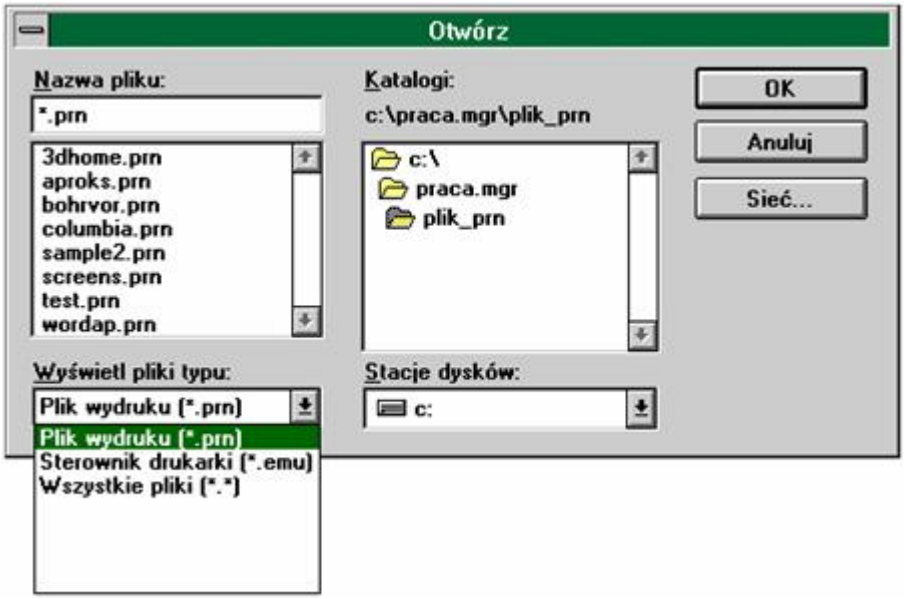


Рис. G. Главное окно приложения

Операции по работе программы обычно ограничиваются несколькими основными:



Открытие документа

Рис. H Окно открыто

Открытию документа, содержащего распечатку принтера, должна

предшествовать загрузка драйвера устройства, для которого создан файл. Драйвер загружается путем выбора пункта *«Открыть»* в меню *«Файл»* или щелчка по соответствующему значку на панели инструментов. После раскрытия диалогового окна «Открыть» *установите* тип файла « *Драйвер принтера»* с расширением *. emu и в диалоговом окне *«Имя файла* » введите имя драйвера, соответствующего вашему принтеру. Имя устройства, для которого загружен драйвер, появится в строке состояния в нижней части окна (рис. 9).

Если код команды некорректно сохранен в контроллере, то при попытке его загрузки программа выведет информацию о номере строки с неверно введенным кодом.

Рис. I Строка состояния с загруженным драйвером.

Программа готова открыть файл печати. Процедура аналогична описанной выше, с той разницей, что в качестве типа открываемого файла следует выбрать *Файлы печати* с расширением *. прн . Если вы попытаетесь повторно открыть уже загруженный документ, программа сообщит об этом пользователю и не позволит вам сделать это. Однако можно добавлять и другие представления к тому же документу; *Добавьте* опцию «Вид» из меню *«Окно»* . Содержимое файла печати будет отображено в окне MDI вместе с заголовком открытого документа. Следующая (предыдущая) страница документа (если таковая имеется) отображается при выборе пункта *Страница* в меню *Правка* или кнопки со стрелкой на панели инструментов (рис. 10).

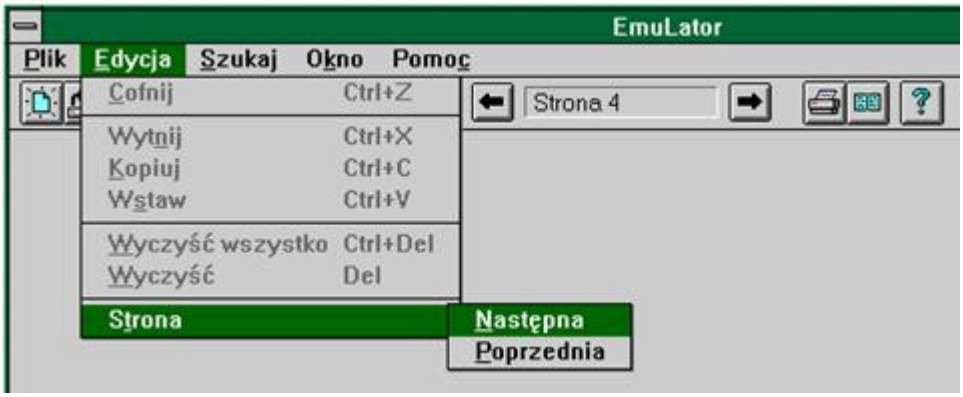


Рис. J Просмотр страниц документа

Закрытие документа

Чтобы закрыть открытый документ, выберите *пункт Закрыть* в меню *Файл* . При смене устройства, на котором отображается документ, все представления (документы) автоматически закрываются.

Предварительный просмотр печати

Чтобы просмотреть документ так, как он будет выглядеть на странице после печати, выберите опцию *«Предварительный просмотр» в меню «Файл»* или соответствующий значок на панели инструментов. Изображение окна нового документа показывает одну или две страницы файла печати. Для просмотра следующих (предыдущих) страниц щелкните значки стрелок на панели инструментов (рис. 11).

Распечатать документ

После просмотра документа вы можете распечатать его на устройстве, подключенном к рабочей станции. Печать выполняется путем выбора пункта *«Печать»* в меню *«Файл»* или путем нажатия на соответствующий значок на панели инструментов. Печать будет выполнена с использованием текущих настроек принтера, которые можно изменить в пункте *«Настройки принтера»* в меню *«Файл»* .

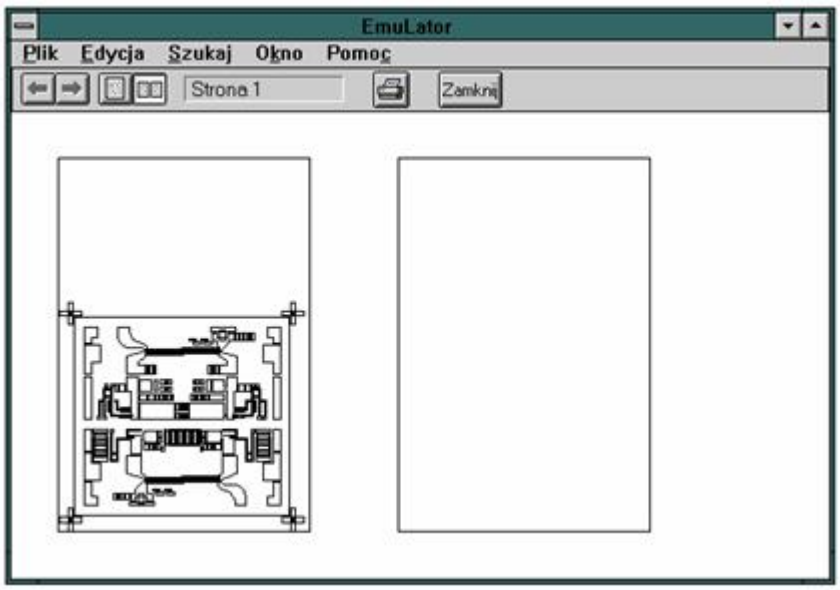


Рис. K Предварительный просмотр печати

Редактирование драйвера принтера

Драйверы принтера можно модифицировать в соответствии с используемым устройством. Их необходимо загрузить в диалоговом окне *«Открыть»* , выбрав тип файла *«Все файлы (*.*)»*. Операции редактирования доступны через пункты меню *«Правка», «Поиск», «Окно»* . Драйвер не может быть загружен во время внесения изменений.

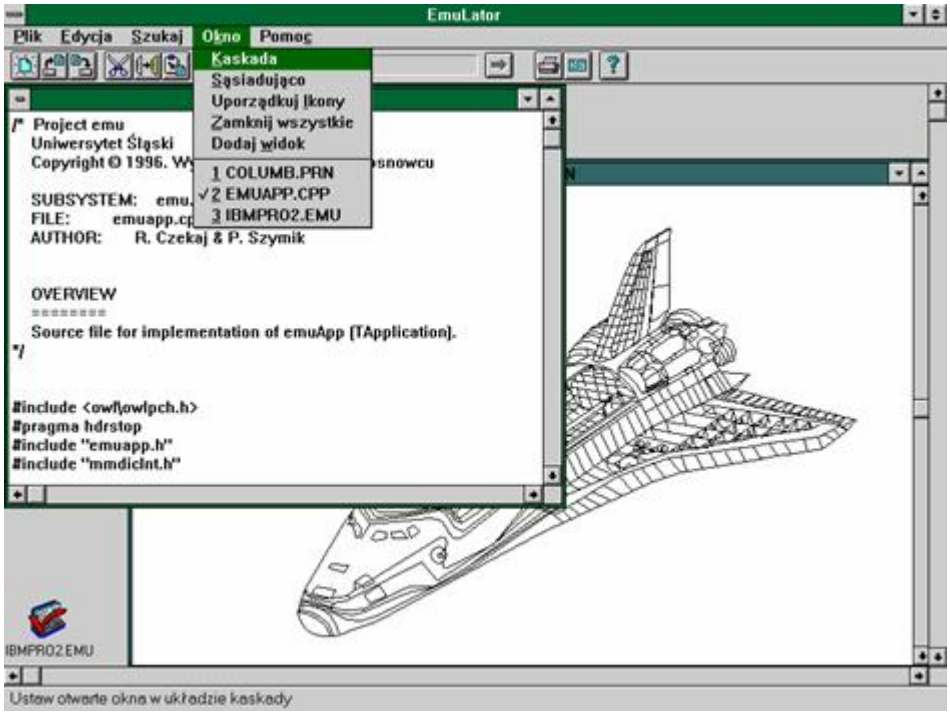


Рис. L Редактирование текстовых файлов

Получение помощи

Справочную систему приложения EmuLator можно вызвать из меню *«Справка»* . Он позволяет вам получить помощь по ключевому слову или теме поиска. При нажатии Shift+F1 указатель мыши меняет форму, позволяя вам указать на элемент управления, о котором вам нужна информация.

Описание драйвера

Формат файла драйвера принтера был создан для приложения EmuLator с использованием простого соглашения об обозначениях. В отличие от других программ, EmuLator использует драйверы, которые не являются компилируемыми программами. Они сохраняются в текстовом формате, не требующем никаких дополнительных операций. Программа имеет встроенную проверку корректности записи и при обнаружении неверного значения пользователю сообщается, в какой

строке оно произошло. Количество кодов управления принтером, которые можно сохранить, в принципе неограниченно, как и количество возможных начальных настроек принтера или плоттера.

Каждая строка, начинающаяся с символа w: ! , # , $, % , воспринимается модулем чтения кода драйвера как комментарий. Каждая строка начинается с символа ! считывается модулем инициализации настройки принтера, и каждая строка должна начинаться с символа комментария или содержать допустимые данные. С момента встречи символа комментария программа игнорирует последующие символы в потоке до тех пор, пока не встретится символ новой строки.

Первая строка драйвера обязательна, вторая, без комментариев, — имя эмулируемого устройства. Об этом сообщается в строке статуса программы.

```
Файл драйвера EMU
$ Файл драйвера плоттера:
Плоттер HP-GL 2
$ за программу EmuLator
```

Описание команды копируется обработчиком потока как целая строка.

Следующий фрагмент содержит данные для одного кода управления принтером.

```
# Пример описания данного кода плоттера:
2 2 байта кода для чтения
0x2b 0xff код управления устройством
описание команды кодовое слово описание
0xffff Код программы EmuLator для операции (шестнадцатеричный код от 0x0
до
# 0xffff)
```

Числа можно вводить произвольно в соответствии с соглашениями языка Си:

```
22 - ДЕК
0x22 - ШЕСТНАДЦАТЕРИЧНЫЙ
022 - ОКТ
```

разделенные пробелом, например 0x56 0x53

Заказы делятся на группы. Разделение введено только с целью повышения наглядности записи контроллера.

```
#########################################################################
# HP-GL2 ЯЗЫКОВЫЕ КОДЫ КОМАНДА #
# ГРУППОВАЯ КОМАНДА В "0A" #
#########################################################################
```

Фактическое описание кода команды приведено в следующих строках файла драйвера. Первый параметр указывает количество байтов в коде драйвера, которые необходимо прочитать. Следующая строка содержит фактические коды управления устройством. После строки описания следует уникальный номер графической операции приложения EmuLator . В приведенном ниже случае это однобайтовый код возврата каретки 0D, расположенный в группе 0A и обозначенный номером 11.

возврата каретки 0D, расположенных в группе под обозначенным номером 11.

```
1
0x0D
возврат каретки
0x0A11
```

Описание одного кода управления принтером должно быть отделено от следующего символом комментария, например:

```
1
0x0A
подача линии LF
0x0A01
#
2
0x56 0x53
выбор скорости рисования
0x131Б
```

Список всех доступных графических операций с соответствующими им номерами находится в каталоге драйвера принтера, в файле *emu_prop. текст* .

После описания всех необходимых кодов управления принтером или плоттером следует описание начальных настроек эмулируемого устройства. Первый элемент одного блока, описывающего одну из настроек, является ключевым словом, последующие — его значениями.

```
#!!!!!!!!!!!!!!!!!!! НАЧАЛЬНЫЕ НАСТРОЙКИ ПРИНТЕРА!!!!!!!!!!!!!!!
# Первая строка – тип настроек, вторая – значение
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# Тип устройства
!Устройство
#!0 # неизвестный тип
#!1 # матричный принтер
#!2 # струйный принтер
#!3 # лазерный принтер
!10 # плоттер
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# Единица, используемая далее
# мм
# дюйм дюйм
!Единица
!мм
#
```

Строки, которые не закомментированы, инициализируют свои значения значениями по умолчанию. Они изменяются, когда в файле печати появляется управляющий код, изменяющий заданное значение.

Краткое содержание

Приложение EmuLator было создано для универсального решения проблемы интерпретации файлов в распечатке. Он включен в инсталляционную версию на дискетах, прилагаемых к описательной части работы. Программа содержит набор базовых графических функций, позволяющих интерпретировать файлы, распечатанные на матричном принтере IBM ProPrinter и плоттере HPGL.

Файл печати может содержать простые текстовые данные и графическое изображение. Программа анализирует файл печати и, найдя управляющий код, вызывает соответствующую графическую функцию, отображая эффект операции на мониторе компьютера. Поиск управляющих кодов в файле включает сравнение символов, извлеченных из потока файла печати, с известными кодами, хранящимися в текстовом файле контроллера. После распознавания управляющего кода следующие байты данных обрабатываются как аргументы вызова графической операции. Нераспознанная последовательность символов обрабатывается как простой текст ASCII, отображаемый на экране. Файлы, сохраненные в текстовом формате, также представлены таким же образом.

Программу можно легко адаптировать для эмуляции других типов устройств, отредактировав текстовые файлы в драйвере принтера в рамках встроенных графических функций. Можно быстро создавать дополнительные объекты, выполняющие другие функции принтера или плоттера. Однако это подразумевает полную перекомпиляцию программы.

В описанном виде программа может быть адаптирована для чтения любого неизвестного формата файла печати, а также других файлов, содержащих любые данные, которые могут быть отображены на экране монитора (например, векторная и растровая графика, текстовый редактор и т. д.).

Надо признать, что универсальность созданной программы не оказала положительного влияния на скорость анализа. Поиск контрольных кодов начинается с самого длинного слова в словаре. Данный метод позволяет унифицировать поддержку различных форматов управления, т.е. возможно чтение как языков управления в текстовом формате (например, PostScript) , так и управляющих последовательностей, начинающихся с символа ESC (например, PCL и ESC/P). Однако для файлов, содержащих в основном символы ASCII, скорость анализа снижается по мере увеличения длины контрольного слова. Программу можно

модифицировать, заменив класс контейнера Borland (от которого произошел *emuSłownik*) на более быструю структуру данных с более эффективным методом поиска. Другим решением является использование разных методов анализа файлов для каждого формата. Однако это требует от программиста написания отдельного модуля для каждого из них, аналогично традиционным скомпилированным драйверам и фильтрам импорта.

Программу можно улучшить, обогатив ее механизмами обмена данными с использованием буфера обмена или даже OLE 2. Добавляя соответствующие функции к классам, производным от *GraphicalObject* , можно даже управлять и улучшать отдельные объекты, например, масштабирование для растровых изображений, любые преобразования для векторных рисунков или изменение типа или размера шрифта.

EmuLator позволяет распечатать просмотренные файлы после их интерпретации. Данные можно распечатать на любом принтере, доступном в среде Windows. Благодаря аппаратной независимости контекста устройства GDI программа может выполнять неявные преобразования любого известного ей формата таким образом, что это позволяет ей передавать результаты на любое устройство, доступное Windows.

Принимая во внимание вышеизложенные соображения и наблюдения, полученные в ходе разработки и эксплуатации приложения EmuLator, разработчики полагают, что цель и предпосылки данной работы были полностью реализованы.

Литература

[1] Баркакати Н.: *Графика и анимация в Windows.* Варшава, Intersoftland 1994, (перевод с английского).

[2] Бартечко К.: *Объектно-ориентированное программирование; Практическое введение в объектно-ориентированное программирование на C++.* Варшава, LUPUS 1993

[3] Дродевич П.: *Программирование для Windows в Ч.* Варшава, Lynx-SFT 1994.

[4] Фэйсон Т.: *Borland C++ 4.5 объектно-ориентированное программирование.* Варшава, Издательство READ ME 1996, (перевод с английского).

[5] Кляйн М.: *Руководство по библиотекам DLL и управлению памятью.* Варшава, Intersoftland 1994, (перевод с английского).

[6] Левин Дж.: *Графическое файловое программирование на C/C++.* Варшава, Переводчик 1994, (перевод с английского).

[7] Марциняк А.: *Язык PCL.* Познань, НАКОМ 1992.

[8] Маккой ВС: *Резюме - Домашняя страница часто задаваемых вопросов о принтерах*. Usenet, comp.periphs.printers Список часто задаваемых вопросов (FAQ), 06.08.1996

[9] Осяк С.: *PostScript шаг за шагом.* Варшава, Издательское агентство M&M 1991.

[10] Смит Н.Э.: *Лазерные принтеры.* Варшава, ZNI MIKOM 1995, (перевод с английского).

[11] Совски Р.: *Звезда заговорщика.* CADmania № 5 (11), ноябрь 1995 г.

[12] Вацлавек Р.: *Программная поддержка лазерных принтеров.* Варшава, Издательство компьютеров HELP, 1992.

[13] Вацлавек Р.: *Окна из кухни.* Варшава, Издательство компьютеров HELP, 1993.

[14] Залевский А.: *Программирование на языках C и C++ с использованием пакета Borland C++.* Познань, НАКОМ 1995.

[15] *Матричный принтер D-100MPC.* Бони, завод точных и механических станков Mera-Bonie, 1991.

[16] *Руководство пользователя принтера LC-20.* Варшава, Интерсофтленд 1991.

Абстрактный

В статье представлены технические аспекты работы современных принтеров и плоттеров, а также методы управления этими устройствами с использованием языков управления печатью. Описанная компьютерная программа является результатом анализа отдельных форматов на языках управления.

Приложение EmuLator позволяет эмулировать работу выбранного принтера и плоттера, а также обеспечивает графическое представление файла распечатки на экране компьютера. Программа разработана с использованием графической среды Windows 3.11 и инструментария программирования Borland C++ версии 4.52.

Программа EmuLator содержит набор базовых графических функций, позволяющих интерпретировать файлы, распечатанные с матричного принтера IBM ProPrinter и плоттера HPGL . В рамках этих функций каждый пользователь может легко подготовить приложение для анализа файлов печати для других устройств, отредактировав текстовые файлы в драйвере принтера.

Подробная документация и комментарии в исходных файлах программы позволяют расширять программу новыми графическими функциями для других устройств. Объектно-ориентированные свойства языка C++ позволяют расширять приложение функциями, которые не были предусмотрены его создателями.

Программа EmuLator вместе со справочной системой доступна на дискетах в инсталляционной версии.

Оглавление