

Билет 8.

1. Операционная среда ЭВМ и САПР. Системное программирование. Типы операционных систем.

Операционная система – это набор программ, контролирующих работу прикладных программ и системных приложений и исполняющих роль интерфейса между пользователями, программистами, прикладными программами, системными приложениями и аппаратным обеспечением компьютера.

Образно можно сказать, что аппаратура компьютера предоставляет "сырую" вычислительную мощность, а задача операционной системы заключается в том, чтобы сделать использование этой вычислительной мощности доступным и по возможности удобным для пользователя. Программист может не знать детали управления конкретными ресурсами компьютера и должен обращаться к операционной системе с соответствующими вызовами, чтобы получить от нее необходимые сервисы и функции. Этот набор сервисов и функций и представляет собой операционную среду, в которой выполняются прикладные программы.

Таким образом, **операционная среда** – это программная среда, образуемая операционной системой, определяющая интерфейс прикладного программирования (API) как множество системных функций и сервисов (системных вызовов), которые предоставляются прикладным программам. Операционная среда может включать несколько интерфейсов прикладного программирования. Кроме основной операционной среды, называемой естественной, могут быть организованы путем эмуляции (моделирования) дополнительные программные среды, позволяющие выполнять приложения, которые рассчитаны на другие операционные системы и даже другие компьютеры.

Еще одно важное понятие, связанное с операционной системой, относится к реализации пользовательских интерфейсов. Как правило, любая операционная система обеспечивает удобную работу пользователя за счет средств пользовательского интерфейса. Эти средства могут быть неотъемлемой частью операционной среды (например, графический интерфейс Windows или текстовый интерфейс командной строки MS DOS), а могут быть реализованы отдельной системной программой – оболочкой операционной системы (например, Norton Commander для MS DOS). В общем случае под **оболочкой операционной системы** понимается часть операционной среды, определяющая интерфейс пользователя, его реализацию (текстовый, графический и т.п.), командные и сервисные возможности пользователя по управлению прикладными программами и компьютером.

Большое разнообразие операционных систем подразделяется на четыре типа, в зависимости от типов компьютеров, которыми они управляют, и приложений, которые они поддерживают. ОС делятся на следующие **категории**.

- **Операционные системы реального времени** (Real-time operating system, RTOS) – используются для управления машинами, научной аппаратурой и промышленными системами. Очень важной частью RTOS является управление ресурсами компьютера таким образом, чтобы определенная операция выполнялась в течение абсолютно одинакового периода времени каждый раз, когда она должна осуществляться. В сложной машине более быстрое перемещение детали, чем это необходимо, лишь по той причине, что это позволяют ресурсы системы, может

привести к катастрофическим результатам, так же, как и невозможность осуществления перемещения этой детали вследствие занятости системы.

- **Однопользовательские однозадачные** – как видно из названия, такие системы предназначены для управления компьютером таким образом, чтобы в любой заданный момент времени один пользователь мог эффективно выполнять одну задачу либо действие. Хорошим примером однопользовательской однозадачной ОС является Palm OS для карманных компьютеров Palm.

- **Однопользовательские многозадачные** – такие ОС большинство пользователей в настоящее время применяют в своих настольных компьютерах и ноутбуках. Windows от Microsoft и MacOS от Apple – примеры операционных систем, позволяющих одному пользователю одновременно выполнять несколько программ.

- **Многопользовательская система** позволяет многим разным людям одновременно пользоваться ресурсами одного компьютера. Операционная система должна сбалансировать требования различных пользователей, а также обеспечить использование каждой задействованной ими программой достаточных и разделенных ресурсов, чтобы проблема, возникшая у одного пользователя, не распространилась на все сообщество пользователей. Примерами многопользовательских операционных систем являются Unix, VMS и операционные системы для больших компьютеров, такие как MVS.

Системное программирование (или программирование систем) - это вид программирования, который заключается в работе с системным программным обеспечением. Главным отличием системного программирования по сравнению с прикладным программированием является то, что прикладное программное обеспечение предназначено выпускать программы для пользователей (например, текстовые процессоры), тогда как системное программирование предназначено выпускать программы, обслуживающие аппаратное обеспечение (например, дефрагментация диска), что обуславливает значительную зависимость такого типа ПО от аппаратной части. Также для системного программирования характерно:

- программист должен учитывать аппаратное обеспечение и другие особенности системы, на которой предполагается запуск программы, и использовать эти особенности (например, применяя оптимизированные алгоритмы для определенной архитектуры)

- обычно используются низкоуровневые языки программирования или диалекты которые:

- могут работать в ресурсно-ограниченной среде
- максимально рациональные и имеют минимальные задержки по времени выполнения
- имеют малые библиотеки периода выполнения (RTL), или вообще их не имеют
- позволит прямой доступ к памяти и управляющей логики
- позволяют программисту писать части программы на ассемблере

- наладивание может быть сложным, если невозможно запустить программу в режиме отладки из-за ограничений в ресурсах. Выполнение программы в эмулируемой среде может снять эту проблему.

2. Автоматизация функционально-логического этапа проектирования цифровых узлов и устройств.

На современном уровне развития технологии проектирования достигнута значительная степень автоматизации функционально-логического этапа проектирования цифровых узлов и устройств. Если раньше при проектировании главным инструментом инженера был паяльник, то теперь ему на смену пришли современные пакеты прикладных программ. Одной из таких помощниц является программа MicroCap. Вместо использования реальных электрических схем (а в этом случае велики затраты и вероятность выхода схемы из строя из-за ошибок инженера) используются их математические модели. Это упрощает, ускоряет и облагораживает процесс проектирования.

Micro-Cap предоставляет практически все виды анализа аналоговых и цифровых схем, а также:

- средства синтеза пассивных и активных аналоговых фильтров;
- интерфейс с программами разработки печатных плат;
- режим анимации при анализе;
- редактор компонентов Component Editor.

Другие достоинства Micro-Cap:

- возможность задания функциональных зависимостей параметров схемы;
- многостраничный графический редактор принципиальных схем, поддерживающий иерархические структуры;
- поведенческое моделирование аналоговых и цифровых компонентов, возможность описания цифровых компонентов с помощью логических выражений, что позволяет моделировать динамические системы, заданные не только принципиальными, но и функциональными схемами;
- графики результатов выводятся в процессе моделирования или после его окончания по выбору пользователя, имеются сервисные возможности обработки графиков;
- многовариантный анализ при вариации параметров и статистический анализ по методу Монте-Карло;
- имеется специальная программа для расчета параметров математических моделей аналоговых компонентов по справочным или экспериментальным данным.

3. Графические форматы. BMP, GIF и JPEG.

BMP

BMP (от англ. Bitmap Picture) — формат хранения растровых изображений, разработанный компанией Microsoft. С форматом BMP работает огромное количество программ, так как его поддержка интегрирована в операционные системы Windows и OS/2.

В данном формате можно хранить только однослойные растры. На каждый пиксель в разных файлах может приходиться разное количество бит (глубина цвета). Microsoft предлагает битности 1, 2, 4, 8, 16, 24, 32, 48 и 64. В битностях 8 и ниже он указывается индексом из таблицы цветов (палитры), а при больших — непосредственным значением. Цвет же в любом случае можно задать только в

цветовой модели RGB (как при непосредственном указании в пикселе, так и в таблице цветов).

В большинстве случаев пиксели хранятся в виде относительно простого двумерного массива. Для битностей 4 и 8 доступно RLE-кодирование, которое может уменьшить их размер.

GIF

GIF - популярный формат графических изображений.

Область применения

Изображение в формате GIF хранится построчно, поддерживается только формат с индексированной палитрой цветов. Стандарт разрабатывался только для поддержки 256-цветовой палитры. Один из цветов в палитре может быть объявлен «прозрачным».

Анимированные изображения

Формат GIF поддерживает анимационные изображения. Они представляют собой последовательность из нескольких статичных кадров, а также информацию о том, сколько времени каждый кадр должен быть показан на экране

Сжатие

GIF использует формат сжатия LZW. Таким образом, хорошо сжимаются изображения, строки которых имеют повторяющиеся участки. В особенности изображения, в которых много пикселей одного цвета по горизонтали.

Чересстрочный GIF

Формат GIF допускает чересстрочное хранение данных. При этом строки разбиваются на группы, и меняется порядок хранения строк в файле. При загрузке изображение проявляется постепенно, в несколько проходов. Благодаря этому, имея только часть файла, можно увидеть изображение целиком, но с меньшим разрешением.

Альтернатива

Существует формат APNG, созданный в 2004 году, использующий 24-битные цвета и 8-битную полупрозрачность, работающий в браузерах Mozilla Firefox и Opera начиная с 2007 года. Некоторые программы и расширения также поддерживают APNG.

JPEG

JPEG—практически является стандартом де-факто для полноцветных изображений. Оперирует областями 8x8, на которых яркость и цвет меняются сравнительно плавно. Вследствие этого, при разложении матрицы такой области в двойной ряд по косинусам значимыми оказываются только первые коэффициенты. Таким образом, сжатие в JPEG осуществляется за счет плавности изменения цветов в изображении.

Алгоритм кодирования:

1. Переводим изображение из RGB в YUV (Y – яркость, V – Хроматический красный, U– Хроматический синий)
2. Разбиваем изображение на матрицы 8x8.
3. Применяем ДКП к каждой рабочей матрице. При этом мы получаем матрицу, в которой коэффициенты в левом верхнем углу соответствуют низкочастотной составляющей изображения, а в правом нижнем — высокочастотной.

4. Производим квантование. В принципе, это просто целочисленное деление рабочей матрицы на матрицу квантования поэлементно. Для каждой компоненты (Y, U и V), в общем случае, задается своя матрица квантования $q[u,v]$ (далее МК). На этом шаге осуществляется управление степенью сжатия. Задавая МК с большими коэффициентами, мы получим больше нулей и, следовательно, большую степень сжатия. В стандарт JPEG включены рекомендованные МК, построенные опытным путем.

5. Переводим матрицу 8x8 в 64-элементный вектор при помощи "зигзаг"-сканирования. Таким образом, в начале вектора мы получаем коэффициенты матрицы, соответствующие низким частотам, а в конце — высоким.

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$	$a_{0,6}$	$a_{0,7}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$a_{1,6}$	$a_{1,7}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{3,0}$			
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$				
$a_{4,0}$	$a_{4,1}$	$a_{4,2}$					
$a_{5,0}$	$a_{5,1}$						
$a_{6,0}$	$a_{6,1}$						
$a_{7,0}$	$a_{7,1}$						

6. Свертываем вектор с помощью алгоритма группового кодирования (аналог RLE). При этом получаем пары типа (пропустить, число), где "пропустить" является счетчиком пропускаемых нулей, а "число" — значение, которое необходимо поставить в следующую ячейку.

7. Свертываем получившиеся пары кодированием по Хаффману с фиксированной таблицей.