# Варианты заданий

Реализовать кольцевую очередь сообщений от родительского процесса дочернему процессу. Каждое сообщение представляет собой структуру, содержащую:

* + Номер сообщения,
  + текстовую строку сообщения, длиной до 30 символов,
  + время отправки сообщения.

Длина кольца- 10 структур. Размещение очереди – в отображаемой памяти. Запись и чтение сообщений –асинхронные (то есть с разной скоростью). Ввод текста сообщения - с клавиатуры. Вывод - в окно дочернего процесса.

Другими словами: два процесса обмениваются сообщениями. Родительский процесс запускает дочерний. В программе – родителе создаем сообщения (вводим текст). Созданное сообщение помещается в очередь. Дочерний процесс периодически просматривает очередь, если в ней есть непрочитанное сообщение, то отображает его в своем окне.

Реализовать очередь сообщений между двумя потоками одного процесса. Каждое сообщение представляет собой структуру, содержащую:

* Номер сообщения,
* текстовую строку сообщения, длиной до 30 символов,
* время отправки сообщения.

Размещение очереди – в памяти процесса (можно использовать подходящий контейнер STL). Запись и чтение сообщений – асинхронные, ввод текста сообщения - с клавиатуры, вывод сообщения вторичным потоком в текстовый файл.

Подробнее: в основном потоке программы пользователь создает сообщения (вводит с клавиатуры текст, нажимает кнопку «отправить») эти сообщения помещаются в очередь. Вторичный поток с заданной периодичностью просматривает очередь, как только в ней появляется необработанное сообщение, записывает его в файл и удаляет из очереди.

Предусмотреть управление вторичным потоком с формы: запуск, завершение работы, изменение периода просмотра очереди во время работы потока.

Реализовать программу с возможностью запуска ограниченного числа дочерних процессов. Статистика: количество разрешенных и запущенных копий динамически отображается на форме родительского приложения.

Очень похоже на лабораторную работу №1. Только надо настраивать количество разрешенных к запуску процессов, отслеживать и отображать на экране, сколько дочерних процессов в данный момент работает.

Реализовать периодический вывод баннера (текстового – сообщения) на экран в главном окне родительского процесса. Тексты сообщений предварительно заносятся в текстовый файл, который читается дочерним процессом. Выбор сообщения- по случайному закону, передача его текста родительскому процессу- через отображаемую память. Место вывода и вид сообщения - по выбору разработчика. Синхронизация доступа к памяти - мьютексы.

Подробнее: родительское приложение запускает дочерний процесс. Дочерний процесс периодически открывает тестовый файл и читает из него случайную строку (сообщение). Считанное сообщение помещает в отображаемую память. Родительский процесс получает сигнал, что сообщение в памяти обновилось, считывает его и отображает у себя в окне.

Смоделировать работу бензозаправочной колонки, имеющей два независимых заправочных шланга-пистолета с одинаковой маркой бензина. Программа должна отображать:

* количество заправляющихся автомашин;
* количество машин, ожидающих заправки;
* прогнозируемое время ожидания в очереди (через сколько времени начнет заправляться только что приехавшая на заправку машина).

Каждая автомашина моделируется отдельным потоком процесса.Объектсинхронизации – семафор.

Подробнее: основной поток имитирует заезд машин на заправку с некоторой частотой, либо по кнопке «добавить машину». Поведение каждой машины моделируется отдельным потоком. Каждая машина (отдельный поток), проходит следующий алгоритм работы:

* Если пистолеты заняты, стоит в очереди на заправку.
* Как только пистолет освобождается – в течение некоторого времени заправляется. Время заправки выбирается случайным числом – моделируется разный объем пустого места в баках машин.
* После заправки машина уезжает, освобождая пистолет.

Основной поток должен подсчитывать и отображать статистику (количество машин ожидающих в очереди, количество заправляющихся машин, среднее время ожидания в очереди).

Смоделировать работу продавцов-кассиров супермаркета.Число рабочих мест задается пользователем в диапазоне 1 – 6 кассиров.То есть в программе должна быть возможность «включать / выключать» кассы во время работы программы. Количество покупателей в зале отбора товара увеличивается генератором случайных чисел, либо нажатием кнопки «добавить покупателя». Работа каждой кассы моделируется отдельным потоком. У каждой кассы есть своя собственная очередь покупателей. Новый покупатель встает в очередь к той кассе, где она короче. Время, необходимое для обслуживания кассиром очередного покупателя, также генерируется случайным образом в разумном диапазоне. Программа (основной поток) должна динамически отображать статистику по каждой кассе:

* Состояние: не работает(«выключена»), занята обслуживанием, свободна
* количество покупателей в каждой кассовой очереди;

Написать программу, моделирующую работу кассы приема платежей. Количество работающих кассиров (1, 2 или 3) выбирается переключателем во время работы программы. Работа каждого кассира моделируется отдельным потоком. Очередь посетителей общая - одна на всех кассиров. Время обслуживания каждого клиента определяется генератором случайных чиселв диапазоне 30 – 200 сек. На главной форме визуально должны динамически отображаться:

* количество обслуживаемых в данный момент клиентов;
* количество клиентов в общей очереди;
* состояние каждого кассира (занят-свободен);
* если он занят – время, оставшееся до окончания обслуживания клиента (секунд).

ГЭС имеет 4 гидрогенератора и снабжает электроэнергией 3 объекта. Мощность, потребляемая каждым объектом, меняется в течение суток относительно некоторого среднего значения по синусоидальному закону. Мощности гидрогенераторов фиксированы:

Ргг1 = 100 МВт

Ргг2 = 100 МВт

Ргг3 = 120 МВт

Ргг4 = 120 МВт

Мощность, потребляемая объектами (МВт):

Р1= 100 + 20\*Sin(pi/2\*(h/6 - 1))

Р2= 180 + 50\*Sin(pi/2\*(h/6 - 1))

Р3=90+ 20\*Sin(pi/2\*(h/6 - 1))

Где h – текущий час в течение суток (0 - 23).

Написать программу, моделирующую работу энергосистемы. Временной масштаб моделирования 1 : 3600 (1 Сек соответствует 1 часу). Моделирование потребления электроэнергии каждым объектом выполняется в отдельном потоке. Первичный поток процесса получает данные от каждого объекта и при необходимости отключает (или включает) гидрогенератор 4, в зависимости от того, хватает ли суммарной мощности первых трех генераторов для энергоснабжения объектов. На главной форме визуально должны отображаться текущие мощности потребления всех объектов, а также состояние всех генераторов (включен -отключен).

Написать программу моделирования работы елочной гирлянды. Рисунок Новогодней елки подготовить самостоятельно или использовать готовую фотографию. Параметры для самостоятельного выбора:

* количество параллельных нитей гирлянды (не менее двух);
* количество лампочек в нити и их цвета;
* алгоритмы мигания лампочек каждой нити (не менее трех алгоритмов).

Работа каждой нити гирлянды моделируется отдельным потоком. Предусмотреть возможность включения / выключения каждой гирлянды во время работы программы (запуска и завершения потока), смены алгоритма мигания гирлянды во время ее работы.

Смоделировать работу светофора пешеходного перехода через дорогу.  
Светофор должен разрешать переход, если число пешеходов, желающих пересечь дорогу, не менее трех человек. После разрешения перехода все пешеходы переходят дорогу, движение транспорта возобновляется через 20 секунд и не может быть приостановлено повторно ранее, чем через минуту. Количество пешеходов моделируется по случайному закону: каждые 10 секунд к светофору подходят от 0 до 3 человек. Можно предусмотреть кнопку на форме, добавляющую внеочередного пешехода к количеству ожидающих на переходе. Если светофор зеленый, то пешеходы не накапливаются: «переходят» сразу.

Работа светофора моделируется в отдельном потоке. Основной поток программы должен отображать количество пешеходов в ожидании и текущее состояние светофора: красный, через N секунд загорится зеленый, гореть зеленому осталось N секунд.

Написать игровую программу «Угадай число». Вторичный поток программы генерирует случайное целое число в диапазоне 1 - 100 и предлагает угадать его. Играющий в поле ввода первичного потока вводит вариант числа, после чего вторичный поток анализирует его и либо сообщает об успешной попытке, либо предлагает ее повторить, сообщая при этом, больше или меньше задуманного введенное играющим число. Способ синхронизации потоков и защиты данных выбрать самостоятельно.

Реализовать программу, которая ежесекундно пытается запустить новый дочерний процесс, формируя при этом случайное целое число секунд (в диапазоне 3 -10), по истечении которого данный процесс должен закрыться. Запускаемый процесс представляет собой окно, в заголовке которого отображается порядковый номер процесса, а в самом окне - счетчик секунд, оставшихся до завершения процесса. Количество одновременно существующих дочерних процессов должно настраиваться в окне родительского процесса. Время отсчитывается родительским процессом: он уведомляет дочерние о том, что прошла очередная секунда.

Использовать передачу параметров командной строки и сообщения (MSG).

Подробнее:

Родительская программа запускает дочерние, передавая им в командной строке два параметра: номер и количество секунд жизни. При этом всем дочерним приложениям раз в секунду посылается сообщение о том, что прошла очередная секунда. Дочернее приложение получив сообщение уменьшает свое время жизни, и как только оно становится равным нулю – закрывается.

Родительское приложение отслеживает дочерние, и как только очередное приложение закрывается – запускает вместо него новое.

При включении стиральной машины в режим отжима, электродвигатель способен разогнать барабан от нуля до скорости 3000 оборотов в минуту за 10 секунд. При этом в машину встроен настраиваемый ограничитель максимального числа оборотов на уровне, который можно выбирать дискретно: 800,1000, 1500, 2000 об/мин.

Написать программу моделирования по линейному закону и отображения на графике текущей скорости вращения (числа оборотов в минуту) барабана стиральной машины.

Моделирование работы двигателя сделать в одном дополнительном потоке (разгоняет барабан). Еще один поток должен моделировать ограничитель, который будет отслеживать текущее значение скорости оборотов и ограничивать его на заданном уровне. Визуализация – основной поток. Объект синхронизации потоков выбрать самостоятельно.

Предусмотреть запуск / остановку потоков, смену уровня ограничения во время работы потоков.

В девятиэтажном жилом доме четыре подъезда, в каждом работает лифт. После пуска кабина лифта перемещается с этажа на этаж за 4 секунды. Написать программу визуального отображения состояния кабины каждого лифта: стоит, движется вверх или вниз, текущее местоположение. Управление каждым лифтом производится оператором в отдельном дочернем процессе. В окне дочернего процесса необходимо предусмотреть числовые поля как для номера этажа, на котором остановилась кабина, так и для номера этажа, куда она должна переместиться после нажатия кнопки «ПУСК». Способ кодирования и передачи данных родительскому процессу, а также способ синхронизации выбрать самостоятельно.

Подробнее: Главная программа запускает четыре дочерних программы – «лифта» и отображает их текущее состояние (номер этажа, стоит / едет вверх / едет вниз). Кнопки управления лифтом находятся в дочерней программе. Нужно придумать, как передавать информацию о состоянии каждого лифта в головную программу.

Смоделировать работу системы слежения за графиком движения автобусов одного маршрута. Движение каждого автобуса моделируется отдельным потоком. Длительность маршрута, количество остановок, расстояние между ними и временной масштаб модели выбрать самостоятельно. Скорость движения каждого автобуса между остановками считать постоянной, однако на каждом перегоне она должна изменяться случайным образом в разумном диапазоне. Программа (основной поток) должна отображать информацию по каждому автобусу и для любой остановки отображать время ожидания прихода автобуса. Первоначальное количество автобусов на маршруте и их начальное местоположение может быть любым. Предусмотреть возможность моделирования поломки транспортной единицы (снятия автобуса с линии) и вывод на линию дополнительного автобуса.

Разработать программу, демонстрирующую на графике фигуры Лиссажу. График строится по следующему закону:

x(t) = Ax \* sin( Wx\*t + Dx);

y(t) = Ay \* sin( Wy\*t + Dy);

(x,y) – координаты точки в момент времени t

Ax, Wx, Dx – параметры X канала

Ay, Wy, Dy – параметры Y канала

Непрерывное вычисление значений x(t) и y(t) необходимо сделать в отдельных дополнительных потоках. Основной поток, каждый момент времени, должен запрашивать у дополнительных потоков текущее значение вычисленных координат X и Y и ставить точку на графике. Предусмотреть на форме кнопки запуска / остановки, изменение параметров (A, W, D), для каждого из дочерних потоков независимо, непосредственно во время их работы.

# РЕКОМЕНДУЕМАЯЛИТЕРАТУРА

1. Таненбаум Э. Современные операционные системы. 2-е изд.: Пер. с англ. - СПб.:Питер, 2002. - 1040 с.

Назаров С.В. Операционные среды, системы и оболочки. Основы структурной ифункциональной организации. Учеб.пособие. - М: КУДИЦ-ПРЕСС, 2007, - 504 с.

РихтерД.Windowsдляпрофессионалов:создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows/ Пер. с англ. - 4-еизд. СПб: Питер;М.: Издательский отдел "Русская редакция ", 2004. - 749 с.

Румянцев П.В. Азбука программирования в Win 32 API. 2-е изд. - М.: Радио и связь,Горячая линия. - Телеком, 1999.

Румянцев П.В. Работа с файлами в Win 32 API. 2-е изд. - М.: Радио и связь, Горячая линия. - Телеком, 2002. - 216 с.