



Программа курса «Системное программирование»

Системне програмування
System programming

Для групп стационара. Версия 3.1.1

Объем курса: 24 пары

Цель курса

Обучить слушателя разработке Windows-приложений с использованием механизмов системного программирования. Получить теоретические и практические знания об управлении памятью в Windows. Изучить организацию динамических модулей в Windows. Разработать приложения, использующие системные ловушки (hooks), многопоточность. Выяснить принципы синхронизации потоков в Windows. Выяснить принципы взаимодействия с системной базой данных Windows (реестр).

По окончании курса слушатель будет:

- взаимодействовать с унаследованным кодом;
- разбираться в основах архитектуры Microsoft Windows;
- использовать механизмы синхронизации;
- порождать процессы;
- создавать многопоточные приложения;
- уметь устанавливать системные ловушки;
- разрабатывать динамически подключаемые библиотеки.

По окончании данного курса студент сдает практическое задание и теоретический экзамен по материалам курса. Для допуска к экзамену должны быть сданы все домашние и практические задания. Практическое задание должно охватывать максимум материала из различных разделов курса.

Важно отметить, что в данном курсе вопросы системного программирования рассматриваются в разрезе Win API и .NET Framework.

Тематический план

Модуль 1. Использование унаследованного программного кода.	2 пары
Модуль 2. Процессы.	2 пары
Модуль 3. Многопоточность и асинхронность.	4 пары
Модуль 4. Синхронизация.	3 пары
Модуль 5. Параллельное программирование.	3 пары
Модуль 6. Управление памятью.	2 пары
Модуль 7. Использование реестра.	2 пары
Модуль 8. Динамически подключаемые библиотеки.	2 пары
Модуль 9. Хуки.	2 пары
Модуль 10. Экзамен.	2 пары

Модуль 1

Использование унаследованного программного кода

1. Почему необходимо использовать унаследованный программный код.
2. Пространство System.Runtime.InteropServices.
3. Взаимодействие с модулями Dll:
 - класс DllImportAttribute;
 - поле ExactSpelling;
 - поле EntryPoint;
 - поле CharSet;
 - поле CallingConvention;
 - поле SetLastError.
4. Примеры использования.

Модуль 2

Процессы

1. Основные сведения о процессах.
2. Функции манипулирования процессами.
3. Понятие дочернего процесса.
4. Манипулирование дочерним процессом.
5. Примеры создания процессов в Win API и .NET Framework.
6. Домен приложения.
7. Использование доменов приложения.

Модуль 3

Многопоточность и асинхронность

1. Что такое многопоточность?
2. Общие сведения о потоках:
 - что такое поток?
 - порождение потоков;
 - приостановка, возобновление, прекращение потока;
 - приоритеты потоков.
3. Потоки в .NET Framework:
 - пространство System.Threading;
 - класс Thread;

- потоки фоновые и первичные;
 - порождение потоков;
 - приостановка, возобновление, прекращение потока;
 - приоритеты потоков.
4. Потоки в Win API:
 - создание потока;
 - потоковая функция;
 - остановка потока;
 - передача параметров потоковой функции;
 - приостановка, возобновление, прекращение потока;
 - приоритеты потоков.
 5. Асинхронный вызов методов.
 6. Использование таймеров обратного вызова.
 7. Использование пула потоков.

Модуль 4

Синхронизация

1. Проблемы синхронизации.
2. Объекты синхронизации:
 - мьютексы;
 - семафоры;
 - события;
 - критические секции;
 - взаимоисключающий доступ.
3. Мьютексы в .NET Framework. Класс Mutex.
4. Семафоры в .NET Framework. Класс Semaphore.
5. События в .NET Framework.
 - класс ManualResetEvent;
 - класс AutoResetEvent.
6. Критическая секция в .NET Framework:
 - класс Monitor;
 - ключевое слово lock.
7. Использование Volatile.
8. Взаимоисключающий доступ в .NET Framework. Класс Interlocked.
9. Использование объектов синхронизации в Win API:
 - мьютексы;
 - семафоры;
 - события;
 - критические секции.

Модуль 5

Параллельное программирование

1. Что такое параллельное программирование?
2. Цели и задачи параллельного программирования.
3. Что такое Task Parallel Library?
4. Класс Task:
 - цели и задачи класса Task;
 - методы класса Task;
 - свойства класса Task;
 - примеры использования класса Task.
5. Класс Parallel:
 - цели и задачи класса Parallel;
 - методы класса Parallel;
 - примеры использования класса Parallel.
6. Структуры данных для параллельного программирования.
7. PLINQ:
 - что такое Parallel LINQ;
 - отличия LINQ от PLINQ;
 - класс ParallelEnumerable:
 - цели и задачи ParallelEnumerable;
 - методы класса ParallelEnumerable;
 - примеры использования класса ParallelEnumerable.
 - скорость выполнения в PLINQ:
 - факторы, влияющие на скорость выполнения;
 - практические примеры.
 - сохранение порядка;
 - параметры для слияния.

Модуль 6

Управление памятью

1. Управление памятью в Windows.
2. Структура виртуального адресного пространства процесса.
3. Понятие региона памяти, страницы памяти, страничного файла.
4. Трансляция виртуального адреса на физический.
5. Защита памяти.
6. Особенности управления памятью при разработке приложений платформы Microsoft .NET.

7. Стек потока.
8. Куча процесса по умолчанию (стандартная куча).

Модуль 7

Использование реестра

1. Основные сведения о реестре.
2. Работа с реестром с помощью Win API.
3. Работа с реестром с помощью платформы Microsoft.NET.

Модуль 8

Динамически подключаемые библиотеки

1. Принципы создания динамически подключаемых библиотек в Visual C++.
2. Неявное связывание.
3. Явное связывание.
4. Отложенная загрузка.
5. Анализ функции DllMain.
6. Разработка динамически подключаемых библиотек с использованием .NET Framework.

Модуль 9

Хуки

1. Создание и использование ловушек.
2. Функции Win API для работы с хуками.
3. Создание хуков в приложениях платформы Microsoft .NET.

Модуль 10

Экзамен