



## Программа курса «Системное программирование»

Системне програмування  
System programming

**Для групп полустационара. Версия 3.1.0**

Объем курса: 5 уроков

### **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:**

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

Оценка по данному курсу выставляется на основании всех сданных домашних заданий.

## Тематический план

- Урок 1.** Процессы, многопоточность
- Урок 2.** Асинхронность, пулы потоков, синхронизация
- Урок 3.** Многопоточность и примитивы синхронизации в WinAPI
- Урок 4.** Параллельное программирование
- Урок 5.** небезопасный код, управление памятью, использование реестра, создание dll модулей

## Урок 1

### Процессы, многопоточность

1. Основные сведения о процессах.
2. Функции манипулирования процессами.
3. Понятие дочернего процесса.
4. Манипулирование дочерним процессом.
5. Домен приложения.
6. Использование доменов приложения.
7. Что такое многопоточность?
8. Потоки:
  - пространство System.Threading;
  - класс Thread;
  - потоки фоновые и первичные;
  - порождение потоков;
  - приостановка, возобновление, прекращение потока;
  - приоритеты потоков.
9. Практические примеры использования.

## Урок 2

### Асинхронность, пулы потоков, синхронизация

1. Асинхронный вызов методов.
2. Использование таймеров обратного вызова.
3. Использование пула потоков.
4. Проблемы синхронизации.
5. Мьютексы. Класс Mutex.
6. Семафоры. Класс Semaphore.
7. События:
  - класс ManualResetEvent;
  - класс AutoResetEvent.
8. Критическая секция:
  - класс Monitor;
  - ключевое слово lock.
9. Взаимоисключающий доступ. Класс Interlocked.
10. Практические примеры использования.

## Урок 3

### Многопоточность и примитивы синхронизации в WinAPI

1. Многопоточность.
2. Принципы создания потоковой функции.
3. Запуск и останова потоков.
4. Приоритеты потоков.
5. Примитивы синхронизации:
  - мьютекс;
  - семафор;
  - события;
  - критическая секция;
  - Interlocked функции.

## Урок 4

### 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.

## Урок 5

### Небезопасный код, управление памятью, использование реестра, создание dll модулей

1. Небезопасный код:
  - что такое небезопасный код?
  - цели и задачи небезопасного кода;
  - примеры использования небезопасного кода.
2. Особенности управления памятью при разработке приложений платформы Microsoft.NET.
3. Основные сведения о реестре.
4. Работа с реестром с помощью Win API.
5. Работа с реестром с помощью платформы Microsoft.NET.
6. Разработка динамически подключаемых библиотек с использованием .NET Framework.
7. Создание хуков в приложениях платформы Microsoft.NET.