

# **Системное программирование. Работа с консолью в Windows API**

# Работа с консолью в Windows API

- Консоль – интерфейс, который используется приложением для ввода/вывода текстовой информации
- Приложение, использующее консоль для обмена данными с пользователями – *консольное*
- Консоль имеет несколько связанных буферов:
  - Входной буфер – информация о событиях ввода (ввод с клавиатуры и мыши, изменение размеров окна)
  - Буфер экрана (один или несколько) – двумерный массив структур (символы и цвет)
- Ввод/вывод в консоль может осуществляться на двух уровнях в Windows API:
  - Высокий уровень (игнорируются все события)
  - Низкий уровень (события не игнорируются)

# Работа с консолью в Windows API

- В Windows существует три стандартных устройства для ввода/вывода:
  - input – ввод данных
  - output – вывод данных
  - error – вывод сообщений об ошибках
- По умолчанию эти устройства связаны с консолью
- Стандартный ввод/вывод можно перенаправлять (на другие устройства, в файлы и т.п.)
- Если стандартный ввод/вывод перенаправлен, то доступ к консоли можно получить через файлы:
  - "CONIN\$" – для ввода информации с консоли
  - "CONOUT\$" – для вывода информации на консоль

# Работа с консолью в Windows API

- **Функции для создания и освобождения консоли (не для консольных приложений):**
  - `BOOL AllocConsole(VOID)` – создание новой консоли, ассоциированной с дескрипторами стандартных устройств ввода/вывода
  - `BOOL FreeConsole(VOID)` – отключение процесса от консоли
- **Функции для работы со стандартными устройствами ввода/вывода:**
  - `HANDLE GetStdHandle(DWORD nStdHandle)` – получение дескриптора стандартного устройства ввода/вывода
  - `BOOL SetStdHandle(DWORD nStdHandle, HANDLE hHandle)` – перенаправление стандартного устройства ввода/вывода на другое устройство
  - Дескрипторы стандартных устройств не закрываются!

# Работа с консолью в Windows API

- `HANDLE GetStdHandle(DWORD nStdHandle):`
  - `nStdHandle` – номер стандартного устройства:
    - `STD_INPUT_HANDLE` – устройство ввода
    - `STD_OUTPUT_HANDLE` – устройство вывода
    - `STD_ERROR_HANDLE` – устройство вывода ошибок
- `BOOL SetStdHandle(DWORD nStdHandle, HANDLE hHandle):`
  - `nStdHandle` – номер стандартного устройства
  - `hHandle` – дескриптор связываемого устройства (файла)
- Дескрипторы стандартных устройств ввода/вывода можно получить и с помощью функции `CreateFile`:
  - `hStdOut = CreateFile("CONOUT$", ...)`
- В Windows API имеются функции для работы с:
  - Окном консоли: `GetConsoleWindow`, `GetConsoleTitle`, `SetConsoleTitle` и др.

# Работа с консолью в Windows API

## ● **Функции высокоуровневого ввода/вывода:**

- `BOOL SetConsoleMode(HANDLE hConsoleHandle, DWORD fdevMode)` – **установка режима консоли**

- `BOOL ReadConsole(HANDLE hConsoleInput, LPVOID lpBuffer, DWORD cchToRead, LPDWORD lpcchRead, LPVOID lpReserved)` – **КОНСОЛЬНЫЙ ВВОД**

- `BOOL WriteConsole(HANDLE hConsoleOutput, LPVOID lpBuffer, DWORD cchToWrite, LPDWORD lpcchWrite, LPVOID lpReserved)` – **КОНСОЛЬНЫЙ ВЫВОД**

- **Возможно использование** `ReadFile` **и** `WriteFile`

- **Лучше использовать функции** `ReadConsole` **и** `WriteConsole` **(а не файловые** `ReadFile` **и** `WriteFile`**), так как:**

- Они используют обобщенные символы
- Учитывают текущий режим консоли

## ● **Функции низкоуровневого ввода/вывода:**

`ReadConsoleInput`, `WriteConsoleInput`, `PeekConsoleInput`, `GetNumberOfConsoleInputEvents` **и др.**

# Работа с консолью в Windows API

## ● **Задание 3:**

### ● Самостоятельно изучить функции Windows API:

- SetConsoleMode
- ReadConsole
- WriteConsole

### ● Написать программу для последовательного выполнения следующих действий с консолью (использовать `GetStdHandle`, `ReadConsole` и `WriteConsole`):

1. Установить режим консоли без эхо-вывода (при вводе)
2. Ввести строку символов с клавиатуры
3. Вывести введенную строку на экран
4. Сохранить введенную строку с консоли в файл
5. Считать строку из файла и вывести на консоль (`CreateFile`, `ReadFile` и `WriteFile`)

# Работа с консолью в Windows API

## ● **Задание 4:**

- Написать программу, перенаправляющую стандартный поток вывода в файл:
  1. Создать файл, получить его дескриптор
  2. Связать стандартный поток вывода, определяемый номером `STD_OUTPUT_HANDLE`, с дескриптором созданного файла
  3. Получить дескриптор стандартного устройства вывода
  4. Записать любую строку в стандартное устройство вывода
  5. Закрыть дескриптор созданного файла
- Вывод в стандартный поток производить при помощи функции `WriteFile` (использовать символы ASCII)



Спасибо за внимание.

