

Course code: **PS/WIN**

Course title: **System programming in Windows**

Days: 5

Description:

Course intended for:

The training is intended for programmers developing applications in the Microsoft Windows class systems, who would like to get familiar with and use the characteristic system mechanisms for this family of operating systems.

Course objective:

The users will learn to take advantage of various system mechanisms in the Microsoft Windows environment, in particular, associated with multithreading, synchronization, inter-process communication and network services. The trainers will discuss the individual areas of application of system mechanism. For each of the topics, the mode of its use will be discussed with practical examples realized by the users in form of short programming tasks. Among other things, the participants will prepare daemon applications and client-server application software.

In particular, the participants will perform practical exercises to get familiar with the issues of design and implementation of applications using the available system mechanisms in Microsoft Windows.

Requirements:

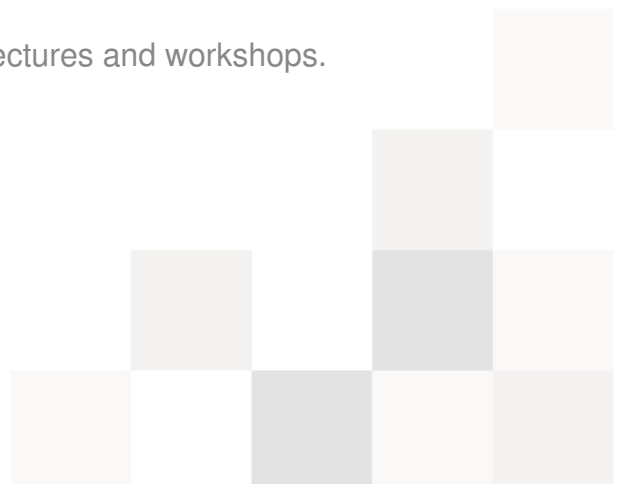
The training participants are required to have the C language programming skills and basic knowledge on operation of the Windows system (logging in, console management, compiling and launching of applications).

Course parameters:

5*8 hours (5*7 net hours). The training consists of lectures and workshops.

Course curriculum:

1. Introduction to Windows system



- I. System architecture and evolution
 - II. Features of an operating system: kernel, virtual memory, multitasking, multithreading, sharing of resources, synchronization, concurrence
 - III. Working with the console
 - IV. Windows SDK and Visual Studio (cl compiler)
 - V. Gcc compiler, gdb debugger
 - VI. Windows API (Win32 and Win64 API)
 - VII. Creation and use of static and dynamic libraries
 - VIII. Basic data types
 - IX. Tracking of system calls, reporting of errors, management of exceptions
 - X. Obtaining of system information
 - XI. Register handling
 - XII. 32 and 64 bit application handling
 - XIII. .NET Framework
2. Message queue
 - I. Message types
 - II. Message management in the application
 3. Processes and threads
 - I. Development of processes and threads, child processes
 - II. User space and kernel space
 - III. Process states
 - IV. Process management, process groups
 - V. Process priorities
 - VI. Process statistics, timers



VII. Process suspending, time management

VIII. Console applications

IX. Windows services

X. Aspects of security: process rights

4. Synchronization

I. The concept of race

II. Critical sections, deadlock

III. Mutex

IV. Semaphores

V. events

5. memory

I. memory access rights

II. memory locking

III. shared memory

IV. memory mapping

6. Operations on files

I. File system management

II. Write/read buffers

III. Symbolic and hard links

IV. Temporary files

V. Asynchronous input/output

VI. Aspects of security: file access rights

7. Inter-process communication



- I. Anonymous and named streams
- II. Queues
- III. Mailslot mechanism
- IV. Other mechanisms (clipboard, events)

8. Network services

- I. Client-service architecture
- II. Winsock and Berkeley sockets
- III. TCP socket programming
- IV. UDP socket programming
- V. IPv4 and IPv6 protocol management

9. Device management

- I. Access to controllers
- II. Interaction with devices

10. Security

- I. Use of intelligent cards (Personal Computer/ Smart Card - PC/SC)
- II. Microsoft CryptoAPI, Cryptography Next Generation (CNG)
- III. Typical programming errors
- IV. Static and dynamic analysis tools

11. System expansion

- I. Development of controllers
- II. Development of system libraries

