State Final Examinations	Academic Year: 2018/2019
Master's Degree Studies Program:	Engineering Informatics
Study Branch:	Information Technologies

Programming

Compulsory Elective Subject

1. Platform-specific and cross-platform programming techniques. Properties, utilization, advantages and disadvantages comparison. Interpreted and compiled programming languages. Basic rules of creating cross-platform source codes. Cross-platform pre-build systems and project control (CMake, Premake, qmake, ...).
2. Qt GUI toolkit - aim, main principles and properties, comparison with available alternatives (other platform-independent GUI toolkits). Main principles used for the development, compilation and linking (on different target platforms, deployment, available development tools), distribution and deployment of applications.
3. Qt GUI toolkit - events processing, a system of signals and slots. GUI development via widgets; dialogues, frame windows, dynamical placement of control elements, Qml, basic conception, QtQuick/QtQuickControls, Qml and C++ integration, document/view architecture.
4. Processing of time-consuming tasks at an application's background. Pseudo-parallel and competitive tasks. Synchronous and asynchronous operations. IDLE time, system timers.
5. Processes, subprocesses and threads. Process and subprocess control and monitoring. Inter-process communication. Types of threads, development of multithread applications, advantages and disadvantages. Thread synchronization (critical section, mutexes, conditions..).
6. Parallel algorithms, types of parallelism. Analysis of memory and time complexity of both sequential and parallel algorithms, Foster's design methodology of a parallel algorithm.
7. Analysis of parallel algorithms - parallel speed-up, cost, work and efficiency. Scaling of parallel algorithms, Amdahl's and Gustafson's laws, Iso-efficiency, Karp-Flatt metric.
8. Parallel architectures and models - basic types of parallel architectures, Flynn-Johnson's taxonomy of parallel systems, memory architectures of parallel systems.
9. Parallel architectures and models - Parallelism available in modern CPUs. PRAM models, submodels and their simulations.
10. Interconnection networks in parallel systems (ICNW), simulation, embedding. General requirements for ICNW, basic types (orthogonal, hypercubic, meshes), their properties.
11. Routing and switching in interconnection networks (ICNW) - classification of communication algorithms, HW architecture of ICNW routers.
12. Routing and switching in interconnection networks (ICNW) - Routing algorithms in orthogonal, hypercubic and mesh networks (e-Cube, XY and XYZ routing, routing in oBF and CCC networks). Dead-locks, prediction and prevention.
13. OpenMP - basic properties and specifics, basic programming techniques and paradigms, parallelization of program's regions, memory model, visibility management of data sources.
14. OpenMP - constructs for workload sharing between threads, synchronization of threads. Single sections, parallel regions, loops, implicit and explicit tasks.
15. OpenMP - inter-thread synchronization and access to shared data sources. Synchronization techniques, constructs and functions.

16. CUDA - General GPGPU architecture, reasons of deployment, general differences between CPU and GPGPU. Basic properties of NVIDIA's Fermi, Kepler, Maxwell and Pascal architectures. Methods for parallelization algorithms applicable on GPGPU.
17. CUDA - Programming and execution models of GPGPU. Threads, blocks of threads, grids, warps. The divergence of the warp. Mapping threads on streaming processors and multiprocessors. Compilation and linking of CUDA applications, an extension of C/C++ in CUDA SDK.
18. CUDA - Heterogeneous memory model. Main memory types available on NVIDIA's GPGPUs and their intended purpose and methods of use. Coalescent and unified memory access. Optimal data transfer between CPU and GPGPU.
19. Parallel algorithms - Parallel reduction and parallel scan (prefix reduction), scaled and non-scaled variants and their use in other parallel algorithms. PRAM implementation on CPU and GPGPU, optimization possibilities according to the target platform, time and memory complexity and speed-up.
20. Parallel algorithms - Discrete 1D and 2D convolution, histograms. Relevance and application in image processing and discrete signal processing. PRAM implementation on CPU and GPGPU, optimization possibilities according to the target platform, time and memory complexity and speed-up.