APPROVED
by the Senate in VUC
decision _____
on _____, 2018

Parallel Computing (ENG)
Paralēlā programmēšana (LV)

Course short name: PC2018

Author	PhD cand., M.Sc.comp. Juris Kļonovs
LAIS course code	
Form of evaluation	Exam
Academic credit points (ECTS credit points)	3 ECTS
The total number of contact lessons	16
The number of lectures	6
The number of practical classes	10
Prerequisites	Java/Python/C programming skills, OOP
Part of the study program	Industry study courses

Introduction and objective of the study course

This is a practical course made to introduce the students with the aspects of Parallel Computing and Parallel Algorithms, their appliances in modern world technologies and applications, and to get familiar with cloud computing. The course will also serve as an introduction to existing real-world parallel programming models including Java Concurrency, MapReduce, MPI, OpenCL and CUDA.

This course incorporates a Project-Oriented Problem Based Learning (POPBL) model that is intended to align students' views with industry needs, which ensures that the student will gain necessary practical experience and knowledge throughout realistic project work.

Study results

Having acquired the study course, students are expected to gain the following competences and skills:

- Understanding the basic principles of parallel computing and programming;
- Ability to use parallel programming languages to solve various problems;

- Ability to realize various parallel reprogramming models that can be applied in modern device architectures;
- Knowledge of basic functionality in OpenMP and MPI languages;
- Ability to differentiate applications from the undivided memory models;
- Understanding the basic concepts of cloud computing;
- Ability to create IaaS, PaaS and SaaS applications.

Organization mode of students' individual work

The independent work of students includes:

- a regular learning of the course by using lecture materials, study literature, internet resources and teamwork;
- course project development;
- work with various libraries;
- preparations for the exam.

Evaluation of study results

The final result is made of:

- Team projects and homework 20%,
- Milestone tests 20%,
- Course paper 20%,
- Final exam 40%.

Study course outline

No.	Title of the topic
1.	Introduction to Parallel Programming Basics (need for parallelism). Main terms. Introduction to the implementation of different programming language parallelism in basic principles. Activity: C-code parsing; Parallel task.
2.	Introduction to multi-core processor architectures. Simulation of parallel programming applications.
3.	Abstract models: parallel computations, computation graphs, Flynn's taxonomy (instruction vs. data parallelism), PRAM model
4.	OpenMP Syntax - Directives, sections, tasks. Data sharing, thread security, synchronization.
5.	Parallel algorithms for data structures, such as arrays, lists, strings, trees, graphs, and key-value pairs.
6.	Common parallel programming patterns, such as task parallelism, pipeline parallelism, data parallelism, divide-and-conquer parallelism, map-reduce, concurrent event processing including graphical user interfaces.
7.	Building innovative application using parallel programming for solving real-world problem (teamwork projects).

Study course schedule

No. of the class	Title of the topic	Type of class, amount of academic hours
1.	Introduction to Parallel Programming Basics (need for parallelism). Introduction to the implementation of different programming language parallelism in basic principles.C-code parsing; Parallel tasks.	1x lecture, 1x practical class
2.	Introduction to multi-core processor architectures. Simulation of parallel programming applications.	1x lecture, 1x practical class
3.	Abstract models: parallel computations, computation graphs, Flynn's taxonomy (instruction vs. data parallelism), PRAM model	1x lecture, 1x practical class
4.	OpenMP Syntax - Directives, sections, tasks. Data sharing, thread security, synchronization.	1x lecture, 1x practical class
5.	Parallel algorithms for data structures, such as arrays, lists, strings, trees, graphs, and key-value pairs.	1x lecture, 1x practical class
6.	Common parallel programming patterns, such as task parallelism, pipeline parallelism, data parallelism, divide-and-conquer parallelism, map-reduce, concurrent event processing including graphical user interfaces.	1x lecture, 1x practical class
7.	GPU programming, thread and memory hierarchy. An Introduction to CUDA, OpenCL, Boost.compute, Thrust, APARAPI and other parallel programming language syntaxes.	1x practical class
8.	Building innovative application using parallel programming for solving real-world problem (teamwork projects).	2x practical classes
9.	Course summary. Preparations for the exam.	1x practical class

Basic literature

1. Doug Lea. Concurrent Programming in Java: Design Principles and Pattern, 2nd Edition 2nd Edition, Addison-Wesley Pub, 1999.
2. Rauber, Thomas, and Gudula Rünger. Parallel programming: For multicore and cluster systems. Springer Science & Business, 2013.
3. Tay, Raymond. OpenCL Parallel Programming Development Cookbook. Packt Publishing Ltd, 2013.
4. Yang, Chao-Tung, Chih-Lin Huang, and Cheng-Fang Lin. "Hybrid CUDA, OpenMP, and MPI parallel programming on multicore GPU clusters."Computer Physics Communications 182.1 (2011): 266-269.
5. Chandra, Rohit, ed.Parallel programming in OpenMP. Morgan Kaufmann, 2001.

Supplementary literature:

1. Nickolls, John, et al. "Scalable parallel programming with CUDA."Queue 6.2 (2008): 40-53.

2. Raynal, Michel. Distributed algorithms for message-passing systems. Vol. 500. Springer, 2013.

Other sources of information

Alternative Parallel Programming languages;
Julia - http://julialang.org/
StreamIT - http://groups.csail.mit.edu/cag/streamit/
Parallel Python - http://www.parallelpython.com/
Matlab - http://www.mathworks.se/products/parallel-computing/?s_cid=sol_compbio_sub2_relprod4_parallel_computing_toolbox
Amazon Web Service - http://aws.amazon.com/free/
AWS resource availability check - http://www.cloudping.info/

Law on Institutions of Higher Education, Section 56.[1] Study Course

(1) Institutions of higher education and colleges shall determine the procedures by which study courses shall be developed and included in study programmes, in order to ensure the achievement of the common study results. The description of a study course shall be prepared and approved in accordance with the procedures specified by the institution of higher education and college.
(2) The study course description shall:
1) define the requirements for the commencement of the acquisition of the study course;
2) determine the aims for the implementation of the study course and the planned study results;
3) outline the content of the study course necessary for the achievement of study results, contain the study course calendar, mandatory and supplementary literature, indicate other sources of information;
4) describe the organisation and tasks for the independent work of students; and
5) determine the evaluation criteria of study results.
[14 July 2011]