Software system development

Academic Year: (2019 / 2020) **Review date: 04-04-2019**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: EXPOSITO SINGH, DAVID

Type: Electives ECTS Credits : 6.0

Year : 4 Semester :

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Operating Systems
Operating Systems Design
Computer Architecture

OBJECTIVES

The objective of this course is to introduce students to different topics related to the domain of system software development. These topics are are the C ++ programming language, the existing services in the Linux / UNIX operating systems, and various development, monitoring and performance analysis tools for developing the systems software. The objective of the course is that the students learn how to develop efficient system software. In order to archive this goal, the student have to acquire several generic skills, knowledge, capacities and attitudes.

1. Generic Skills:
- Capacity for analysis and synthesis (PO a, c, e)
- Strong organisational and planning (PO c, d)
- Capacity for solving problems (PO c, g)
- Teamwork (PO d)
- Ability to apply theoretical concepts (PO a, c)

2. Specific Skills:
a. Cognitive (PO a, c, e)
- Ability to design software systems
- Ability to implement software systems
- Knowledge of I/O techniques
- Knowledge of concurrent programming
- Use of services offered by operating systems
b. Procedural / Instrumental (PO a, c, g, k)
- Solve problems of concurrency.
- Develop low-level system software.
c. Attitudinal (PO: c, e)
- Creativity
- Critical view of software system development
- Proactivity
- Interest in acquiring new knowledge and information

3.- General and Transversal Competences
- To use in an efficient way electronic tools for writing technical report, project memos and reports about computer science, including high quality presentations (CG9)
- Basic knowledge about the usage and the programming of computers, operating systems, data bases, and computer applications with engineering applications (CGB4)

4.- Computer Science's related competences
- Knowledge of characteristics, functionality and structure of operating systems, and to design and implement software based on its services (CECRI10)

5.- Computer Engineering related competences
- Knowledge to design and implement system software and communication software (CEIC4)

DESCRIPTION OF CONTENTS: PROGRAMME

T1. Introduction to Programming Systems
1.1 Definition of programming systems
1.2 History of programming languages

T2. Systems Programming languages: C ++
2.1 Objects, types and values
2.2 Classes and inheritance. Operator overloading
2.3 Containers, arrays and Free Store
2.4 Copy and move operations
2.4 Templates, exceptions and resource management
2.5 Metaprogramming
2.6 STL, function objects and lambdas
2.8 C++11 and C++14

T3. Concurrency
3.1 Thread-based concurrency
3.2 Task-based concurrency

T4. Management and memory optimization
4.1 Error control and debugging
4.2 Shared memory
4.3 Memory mapping

T5. Input / output
5.1 Vectorized I/O
5.2 Asyncronous I/O
5.3 I/O Buffering

T6. Signals and timers
6.1 I/O mutiplexing
6.2 Signal-based I/O
6.3 No re-entry functions and non-local goto

T7. Management of libraries and utilities
7.1 Static and shared libraries
7.2 Dynamic libraries

T8. Monitoring and performance analysis
8.1 Performance metrics
8.2 Performance tools
8.3 Linux's monitoring tools

LEARNING ACTIVITIES AND METHODOLOGY

1. Theoretical lectures 1 ECTS (PO a, c, e)
- Presentation of the theoretical concepts
- Students will have available for students support material to extend and increase their knowledge in the course topics.

2. Assignments 1.5 ECTS (PO a, c, d, e, g, k)
- There will be several work assignments. Some of the will be completed in the lab classes with teacher support.
- Students will have to analyse the requirements and provide a correct solution to each assignment
- Assignments will be completed in groups of 2 students (to promote the team work).

3. Exercises solved by the teacher 1 ECTS (PO a, c, e)
- Several exercises will be solved during class-time with the student participation

4. Student work 2 ECTS. (PO a, c, e, k)
- Studies to understand the theoretical contents
- Completion of the proposed exercises

5. Exams 0.5 ECTS (PO a, c, e, g)
- Final exam

ASSESSMENT SYSTEM

40% of the final grade is obtained through a final exam evaluation of acquired knowledge. The remaining 60% will be the result of a process of continuous assessment.

The continuous assessment process includes:
- Lab assignments with a weight of 30% (PO: a, c, d, e, g, k).
- Exercises and laboratory work, with a weight of 30% (PO: a, c, d, e, g, k).

Is considered to have followed the continuous assessment if they have delivered all the practices and there have been at least two thirds of the proposed exercises and laboratory work.

To pass the las assignments it will be necessary to obtain a minimum score of 2 in each. The final assessment test has a weight of 40%. To pass the exam you must obtain a minimum grade of 4.0 out of 10.

In case of detection of copy, the students involved will lose continuous assessment. The copy is meant peer or copying practices from Internet.

Continuous assessment will be lost if all practices are not delivered, not the minimum is reached in practices or exercises and laboratory work.

If the minimum is not reached, the student will fail with a score of 4.5 out of 10.

For the calculation of the final grade in the resit the following cases are considered:
A-Students who have followed the process of continuous assessment:
a-The rating for the special exam will count 40%.
b-The remaining 60% will come from continuous assessment score.
c-will be required to take a minimum score of 4 points out of 10 for grade continuous evaluation
B-Students who have not followed completely continuous evaluation process:
The special examination will count 100% and may include a portion of practice to verify that they have this knowledge. The ordinary exam will represent 60% of the grade for those students who choose not to join the previous system of continuous assessment.

% end-of-term-examination:	40
% of continuous assessment (assigments, laboratory, practicals…):	60

BASIC BIBLIOGRAPHY

- A. Hoover System Programming with C and Unix, Pearson, 2009
- Bjarne Stroustrup Programming: Principles and Practice using C++, Addison-Wesley, 2009
- Michael Kerrisk The Linux Programming Interface. A Linux and UNIX System Programming Handbook, William Polock, 2010

ADDITIONAL BIBLIOGRAPHY

- Bjarne Stroustrup The C++ Programming Language, Addison-Wesley, 2013
- G. R. Andrews Foundations of Multithreaded, Parallel, and Distributed Programming, Addison-Wesley, 1999
- Kay A. Robbins and Steve Robbins UNIX Systems Programming: Communication, Concurrency and Threads, Pearson , 2003
- L. L. Beck System Software: An Introduction to Systems Programming, Pearson , 1996
- M. Barr Programming Embedded Systems in C and C++, O'Reilly Media, 1999
- Nicolai M. Josuttis The C++ Standard Library, Addison-Wesley, 2012

BASIC ELECTRONIC RESOURCES

- Tenouk . The Tenouk's C, C++: http://www.tenouk.com/