# JSC «Kazakh-British Technical University» Faculty of Information Technology

**Electrical Engineering and Computer Science Department**

**SYLLABUS**

**Discipline:** System programming **Number of credits:** 3 (2/1/0) **Term:** Fall 2021

**Instructor’s full name:** Amandyk Kartbayev

# APPROVED BY

**Dean of FIT**

1. **Bisembaev**

**« » 202**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Personal Information about the Instructor** | **Time and place of classes** | | **Contact information** | |
| **Lessons** | **Office Hours** | **Tel.:** | **e-mail** |
| Amandyk Kartbayev, PhD, associate professor | According to  the schedule | According to the  schedule |  | a.kartbaev@kbtu.kz |

**Course duration:** 3 credits, 15 weeks, 45 class hours

**Course prerequisites:** “Computer Fundamentals”, “Programming languages”, “Programming technologies”

# Course description:

System programming is an essential and important foundation in any computer’s application development, and always evolving to accommodate changes in the computer hardware. This kind of programming requires some level of hardware knowledge and is machine dependent; the system programmer must therefore know the intended hardware on which the software is required to operate.

Additionally, the programmer may make some assumptions on the hardware and other system components. The software is usually written in a low-level programming language which is able to operate efficiently in a resource-constrained environment, and with little runtime overhead using a small library, or none at all. The low-level language enables direct control over memory access and allows the program to be written directly in an assembly language. The majority of programs are written using assembly languages such as C and C++.

System programming leads to the development of computer system software that manages and controls the computer operations. The low-level codes are very close to the hardware level and deal with things such as registers and memory allocations. The system programs or system software coordinates data transfer across the various components and deals with the compiling, linking, starting and stopping of programs, reading from files as well as writing to files.

The system programming enhances or extends the functions of an operating system and may comprise components such as drivers, utilities and updates. They enable efficient management of hardware resources such as memory, file access, I/O operations, device management and process management such as process administration and multi-tasking. An example is an operating system, which usually acts as the interface between the user, the application software, and computer hardware. The OS provides an environment that enables users to execute other programs efficiently. Comprising of a set of system programs, the operating system functions include storage

management, file handling, memory management, CPU and device scheduling and management, error handling, process control and more.

# Course objectives

This course will introduce students what is a System Programming. During practice session you will learn how to solve simple lab works and then complete on your SIS projects.

# Literature

 Robert Love, “Linux Kernel Development”, 3rd Edition, 2010.

 Robert Love, “Linux System Programming”, O’Reilly, 2nd Edition, 2013.

# COURSE CALENDAR

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Topic** | **Lectures** | **Labs** |
| 1 | **Introduction to System Programming** | 2 | 1 |
| 2 | **System Calls** | 2 | 1 |
| 3 | **OS Structure** | 2 | 1 |
| 4 | **The Kernel Abstraction** | 2 | 1 |
| 5 | **Process Scheduling** | 2 | 1 |
| 6 | **Concurrency and Threads** | 2 | 1 |
| 7 | **Kernel Synchronization** | 2 | 1 |
| \*8 | **Midterm Exam** | 2 | 1 |
| 9 | **Deadlocks** | 2 | 1 |
| 10 | **Memory Management** | 2 | 1 |
| 11 | **FS Abstraction** | 2 | 1 |
| 12 | **FS Implementation** | 2 | 1 |
| 13 | **Address Translation** | 2 | 1 |
| 14 | **Demand Paging** | 2 | 1 |
| **\***15 | **Endterm Exam** | 2 | 1 |
|  | **Total** | 30 | 15 |
|  | **Overall** | 45 | |

**TASKS**

for students independent study (SIS)

|  |  |  |
| --- | --- | --- |
| **Week** | **SIS** | **Cost (in points)** |
| **3** | **Getting acquainted with assembly language and the linking process.** | **5** |
| **7** | **ASCII and Hexadecimal** | **5** |
| **11** | **Unlimited Precision BCD Calculator** | **5** |
| **13** | **Understanding stack manipulations** | **5** |
|  | **Total** | **20** |

**Course assessment criteria**

|  |  |
| --- | --- |
| **Type of activity** | **1** |
| Group assignments | 40 |

|  |  |
| --- | --- |
| SIS | 20  0  0  **0** |
| TSIS |
| Midterm (SIS/TSIS) |
| Attendance / participation  Missing 20% attendance – **F grade** |
| Final exam | 40 |
| **Total** | **100** |

# Criteria for evaluation of students during semester

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Assessment criteria** | **Weeks** | | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |  |
| 1. | Laboratory works | \* | \* | \* | \* | \* | \* | \* |  | \* | \* | \* | \* | \* | \* |  | 40 |
| 2. | SIS/TSIS |  |  |  |  |  |  |  | \* |  |  |  |  |  |  | \* | 20 |
| 3. | Attendance / participation | \* | \* | \* | \* | \* | \* | \* | \* | \* | \* | \* | \* | \* | \* | \* | 0 |
| 4. | Midterm / end of term |  |  |  |  |  |  |  | \* |  |  |  |  |  |  | \* | 0 (SIS/TSIS) |
| 5. | Final exam (classmarker testing with proctor edu) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 40 |
|  | Total |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 100+10 |

**Academic Policy**

KBTU standard academic policy is used.

* Cheating, duplication, falsification of data, plagiarism, and crib are not permitted under any

circumstances!

* Attendance is mandatory.

**Attention**. Missing 20% attendance to lessons, student will be taken from discipline with filling in F (Fail) grade.

Students must participate fully in every class. While attendance is crucial, merely being in class does not constitute “participation”. Participation means reading the assigned materials, coming to class prepared to ask questions and engage in discussion.

* Students are expected to take an active role in learning.
* Written assignments (independent work) must be typewritten or written legibly and be handed in time specified. Late papers are accepted with – 35% penalty!
* Final exam will be conducted via classmarker testing with proctor edu system
* Students must arrive to class on time.
* Students are to take responsibility for making up any work missed.
* Make up tests in case of absence will not normally be allowed.
* Mobile phones must always be switched off in class.
* Students should always be appropriately dressed (in a formal/semi-formal style).
* Students should always show tolerance, consideration and mutual support towards other students.

***PhD Kartbayev A.***