OPERATING SYSTEMS

Tasks and Objectives

The course is intended for mastering the basic concepts, methods and algorithms of architecture and functioning of operating systems. The subject includes models and algorithms used in realization of various subsystems, application runtime environments and architecture of state-of-the-art operating systems. Special attention is given to operation of multitask systems, multiprocess and multithreaded applications, which is critical for development of programs for multiprocessor / multicore systems.

Student Requirements

Knowledge of K101 "Introduction to Programming Methods 1", K102 "Introduction to Programming Methods 2", M206 "Algorithms and Complexity Analysis ", K206 "Architecture of Computing Systems". For performance of practical and laboratory tasks knowledge of C programming methods is required.

Course Description

The main sections of the course are:

- **Theory of Operating Systems. Basic Concepts and Definitions – 14 academic hours**

  Tasks and objectives. History and evolution of OS. OS functions. OS classifications. Examples of architecture (Windows NT, UNIX). System resource. Operation environment. Process. Thread. State-chart of sequential thread execution.

- **Models of Hardware Level Objects of– 12 academic hours**

  Methods of memory addressing and corresponding controlling structures; linear; segmental; page; segmental and page. Organization of shared access to the memory of several processors. Multitask operating mode of a processor. Task context; task switching; privilege levels; control transfer between privilege levels. Breakups and special cases.

- **Central Processor Control – 4 academic hours**

  Long-term and short-term planning. Criteria of planning algorithms comparison. Classification of short-term planning algorithms. Examples of algorithms: FIFO, SJN, SRT, RR, which are preemptive and non-preemptive algorithms, priority algorithms. Planning in multiprocessor systems.

- **Synchronization of Thread / Process PAR Execution – 12 academic hours**

  Critical resources and critical process sections. Use of lockup indication. Dekker's algorithm. Test-and-set operation. Dijkstra's semaphore primitives. Mutex. Supplier-customer and reading-writing tasks. Problem of deadly embrace (deadlock).

- Control of Random Access Memory – 4 academic hours

  Simple contiguous memory allocation. One or several executed tasks. Control with partitioning. Internal and external fragmentation, swapping. using WAP based on page transformation. Algorithms of memory areas substitution: Random, FIFO, LRU, LFU, Second Chance, Clock.

- Data Transmission Between Threads/Processes – 8 academic hours

  Examples of inter-process communication in UNIX and Win32. Signals. Messages, message queues. Memory mapping. Named and anonymous pipes. Synchronization objects in Windows NT. Remote procedures calls.

- Long-Term Data Storage – 14 hours

Purpose of file systems and executed tasks. Logical structure of file systems. Types of objects in file systems and their attributes. Standard operations with objects in a file system. Classification, further capabilities and file systems comparison criteria. Disk partition. Example file systems: original UNIX FS, FFS, JFS, LSFS.

- Distributed File System Building Principles (GlusterFS, GPFS, Lustre) STEAM – 4 academic hours

The course includes expanded laboratory training.

Course materials:

- presentations for lectures;
- laboratory training sets.

Expected Training Results

Trainees who successfully completed the training will acquire the following knowledge and skills:

- principles of building of state-of-the-art operating systems;
- basic models and algorithms for OS functioning;
- development of application and system software based on the architecture and implementation features for realization of a target OS;
- development of multiprocess and multithreaded applications;
- application programming for various operational environments.

References

1. Tanenbaum A. Modern Operating Systems 2nd Ed, Piter, 2002.
2. Karpov V., Konkov K Introduction to Operating Systems, Course of lectures 2nd Ed, INTUIT.RU, 2005.
3. Advanced Windows [Richter J.], 4th Ed.
4. Solomon D., Russinovich M. Windows 2000 Internal.
5. Love R. Linux Kernel Development. SAMS, 2003.

Contacts

Liniov A., Chief of the University Laboratory, IIS GEO,
alin@unn.ru