

سیستم عامل و برنامه نویسی سیستم

Operating Systems and System Programming
Sharif University of Technology, EE Department
Fall 2018

Iman Gholampour

Focus

This course firstly focuses on the general concepts of operating systems, the roles, elements and types. Afterwards, we get into the details of the widely-used open-source OS, Linux. Linux kernel and device drivers are fully investigated and the students learn to write proper, standard kernel modules and device drivers of various types. The main titles can be summarized as follows:

- Operating Systems, General Concepts, Elements and Types
- Hardware Resources and User's Needs
- Process, Thread and Memory Management
- Fair-to-All, Real-time Race Mechanisms
- File Systems
- Linux Architecture
- System Programming
- Kernel Modules and Interfaces
- Kernel Programming
- Writing Device Drivers
- Core Drivers Programming

In programming terms, these titles live on the 4th and the 5th floors of Linux programming architecture, specially the 5th:

- 1) Command Line Scripts; to automate the boring routines, mostly using bash scripts
- 2) Command Line/User Space Application Development; to handle the everyday needs of expert users, using any sort of programming languages and various libraries
- 3) Desktop/User Space Application Development; to make a better user experience through creating a Graphical User Interface, mostly by GTK+, QT or Electron
- 4) **System Programming; using system APIs, system calls and other administrative access to accomplish a serious task like process management**
- 5) **Linux Kernel Module/Driver Programming; it is the deepest you can get, manipulating the parts of Linux Kernel and create a wide range of interesting programs that run quickly and quietly in the background and do all the important stuff**

Syllabus

- Operating Systems, Definitions, Introduction
- Operating System Elements
- Threads and Processes
- Concurrency and Synchronization
- Scheduling and Resource Sharing
- Race
- Virtual Memory
- I/O
- Disks
- File systems
- Protection & Security

- Virtual machines
- UNIX and UNIX Like OSes
- User Space and Kernel Space
- Operating Systems Classification
- Monolithic Operating Systems, Early Linux Versions
- Micro Kernel Operating Systems, MINIX
- Dynamic Kernel, Current Linux Versions
- Make, Building the Kernel
- System Kernel and Device Drivers
- User Space, Applications and the System Call
- System Programming, Using System Kernel Calls (APIs), Interfaces
- Linux Subsystems
- Linux File System
- Debugging Techniques
- Atomic Context vs Process Context
- Kernel Data Structures
- System Calls and Interfaces
- Notifier Chains
- Time, Delay, Deferred Works and Timers, Time Stamps, Sleep and Busy Waiting, Blocking IO
- Interrupt Handling
- Process Management, Multitasking, Process Scheduling, Signals
- Memory Management
- Multi Core Environment and Jiffies
- Embedded Linux, Tailoring the OS
- Linux Kernel Modules/Driver Programming
- Device Drivers
- Core Driver Concept and Methodology
- Char Device Drivers: Parallel Ports, TTY, PCI, USB

- Block Device Drivers, Hard Disks, CD/DVD ROMs, ...
- Network Drivers, Addressing, Protocols, Interfaces, Filter Hooks
- Miscellaneous Drivers, Power Management Interface, Platform Interface, Miscellaneous Driver Interfaces, SystemCall Interface
- Student Projects

References

- 1) **Operating System Concepts, A. Silberschatz et.al, 9th Edition, Wiley 2013.**
- 2) **Modern Operating Systems, A.S. Tanenbaum, 4th Edition, Pearson 2015.**
- 3) Linux System Programming, R. Love, 2nd Edition, O'Reilly 2013.
- 4) Linux Kernel Development, R. Love, 3rd Edition, Addison Wesley 2010.
- 5) The Art of Linux Kernel Design, Y. Lixiang et al., CRC Press 2014.
- 6) Linux Device Drivers, J. Corbet, A. Rubini, and G. K. Hartman, 3rd Edition, O'Reilly 2005.
- 7) The Linux Programming Interface, M. Kerrisk, No Starch Press, 2010
- 8) Many more books and other resources:
 - Hundreds of Films, Papers, Reports, Presentations, Module/Driver Examples, Code Samples by Professionals and the Course Instructor.
 - Free Electrons Resources
 - Github: Kernel Source code and other Samples