Unit 1: Overview of Operating Systems

1.1. Windows Operating System Internals Course Overview

Copyright Notice

- These materials are part of the CRK (aka *Windows Operating System Internals Curriculum Resource Kit*), developed by David A. Solomon and Mark E. Russinovich with Andreas Polze
- Microsoft has licensed these materials from David Solomon Expert Seminars, Inc. for distribution to academic organizations solely for use in academic environments (and not for commercial use)

Basic Information

- Windows Operating Systems Internals is a pool of material and resources that explains operating systems (OS) concepts based on the Microsoft Windows NT operating system family,

  structured following the ACM/IEEE Operating System Body of Knowledge (“BOK”) as defined in Computing Curriculum 2001 project by the Joint IEEE and ACM Task Force (“CC2001”).

- This curriculum is based on the book “Windows Internals”, by Mark Russinovich and David Solomon.

Roadmap for Section 1.1.

- ACM/IEEE Operating Systems Body of Knowledge - Core units
- ACM/IEEE Operating Systems Body of Knowledge - Elective units
- Supplementary units

Unit 1:
Overview of Operating Systems

- 1.1. Windows Operating System Internals Course Overview (Core)
- 1.2. The Evolution of Operating Systems (Core)
- 1.3. Windows Operating System Family – Concepts & Tools (Core)

Unit 2:
Operating System Principles

- 2.1. Structuring of the Windows Operating System (Core)
- 2.2. Windows Core System Mechanisms (Core)
- 2.3. Windows on Windows - OS Personalities (Core)
- 2.4. The Windows API – Naming Conventions, Types (Core)

Unit 3: Concurrency

- 3.1. Concurrency, Critical Sections, Semaphores (Core)
- 3.2. Windows Trap Dispatching, Interrupts, Synchronization (Core)
- 3.3. Advanced Windows Synchronization (Core/Advanced)
- 3.4. Windows APIs for Synchronization and Inter-Process Communication (Core/Advanced)

Unit 4: Scheduling and Dispatch

- 4.1. The Concept of Processes and Threads (Core)
- 4.2. Windows Processes and Threads (Core)
- 4.3. Windows Process and Thread Internals (Core/Advanced)
- 4.4. Windows Thread Scheduling (Core)
- 4.5. Advanced Windows Thread Scheduling (Core/Advanced)

Unit 5: Memory Management

- 5.1. Memory Management for Multiprogramming (Core)
- 5.2. Windows Memory Management Fundamentals (Core)
- 5.3. Virtual Address Translation (Core)
- 5.4. Physical Memory Management (Core/Advanced)

Unit 6: Device Management - The Input/Output System

- 6.1. Principles of I/O Systems (Elective)
- 6.2. The Windows I/O System Components (Elective)
- 6.3. Windows I/O Processing (Elective/Advanced)

Unit 7: Protection and Security

- 7.1. The Security Problem (Elective)
- 7.2. Windows Security Components and Concepts (Elective)
- 7.3. Windows Security Descriptors (Elective/Advanced)

Unit 8: File System

- 8.1. Background: Unix File Systems (Elective)
- 8.2. The Windows File System (NTFS) (Elective)
- 8.3. Encrypting File System Security in Windows OS (Elective/Advanced)
- 8.4. NTFS – Recovery Support (Elective/Advanced)
- 8.5. Windows File and Directory Management APIs (Elective)

Unit 9: Real-time and Embedded Systems

- 9.1. Introduction and Vocabulary (Elective)
- 9.2. Real-Time Systems with Windows (Elective)
- 9.3. Embedded Systems with Windows XP Embedded (Elective)

Unit 10: Fault-tolerance

- 10. Fault-tolerance (Elective)
- 10.1. Fault-tolerance in Windows (Elective)

Unit 11: Performance Evaluation

- 11. System Performance Evaluation and Troubleshooting (Elective)
- 11.1. System Performance
- 11.2. Boot/Startup Troubleshooting

Unit 12: Scripting

- 12. Scripting (Elective)
- 12.1. Windows Management Instrumentation
- 12.2. The Registry

Supplementary Unit A: Windows Networking

- A.1. Networking Components in Windows OS (Supplementary/Advanced)
- A.2. Windows Socket Programming (Supplementary/Advanced)
- A.3. Microsoft-specific extensions to Sockets and other Networking APIs (Supplementary/Advanced)

Supplementary Units B: Comparing the Linux and Windows Kernels

- B. Comparing the Linux and Windows Kernels (Supplementary/Advanced)
- B.1. Linux and Windows Comparison (Supplementary/Advanced)

Supplementary Units C: Windows – Unix Interoperability

- C. Windows – Unix Interoperability (Supplementary/Advanced)
- C.1. File and Command Interoperability (Supplementary/Advanced)
- C.2. Programming (Supplementary/Advanced)

Suggested Reading

- Pavel Yosifovich, Alex Ionescu, et al., “Windows Internals”, 7th Edition, Microsoft Press, 2017.
  - (This book is the course’s direct companion - required reading).
- Abraham Silberschatz, Peter B. Galvin, Operating System Concepts, John Wiley & Sons, 6th Ed., 2003;
  - (This book gives a platform-neutral overview on operating system design and implementation – suggested reading).
- Jeffrey Richter, Programming Applications for Microsoft Windows, 4th Edition, Microsoft Press, September 1999.
  - (This book provides a comprehensive discussion of the Windows API – suggested reading).

Suggested Reading (contd.)

- Johnson M. Hart, Win32 System Programming: A Windows® 2000 Application Developer's Guide, 2nd Edition, Addison-Wesley, 2000.
  - (This book discusses select Windows programming problems and addresses the problem of portable programming by comparing Windows and Unix approaches).
- Andrew S. Tanenbaum, Distributed Operating Systems, Prentice Hall, 1995.
  - (This text accompanies the course’s advanced modules and offers students an explanation of the concepts of distributed operating systems).
- http://www.sysinternals.com
  - (Tools useful for experiments with Windows internals)
- http://msdn.microsoft.com
  - (Online resource for information on APIs required during work on labs).