

| | |
|-------------------------|--|
| 1. Record Nr. | UNINA9910459604703321 |
| Autore | Kolaja Jiri Thomas <1919-> |
| Titolo | A Polish factory : a case study of workers' participation in decision making / / Jiri Thomas Kolaja |
| Pubbl/distr/stampa | [Lexington, Kentucky] : , : University Press of Kentucky, , 1960 ©1960 |
| ISBN | 0-8131-6373-0 |
| Descrizione fisica | 1 online resource (176 p.) |
| Disciplina | 658.3152 |
| Soggetti | Textile workers - Poland Management - Employee participation - Poland Electronic books. |
| Lingua di pubblicazione | Inglese |
| Formato | Materiale a stampa |
| Livello bibliografico | Monografia |
| Note generali | Includes index. |
| Nota di bibliografia | Bibliographical footnotes. |
| Nota di contenuto | Cover; Title; Copyright; Foreword; Acknowledgments; Contents; Introduction; CHAPTER 1. The Polish Workers' Council; CHAPTER 2. A Lodz Textile Factory; CHAPTER 3. Past Events in the Factory; CHAPTER 4. Eight Weeks in the Factory; CHAPTER 5. Analysis of the 13 Events; CHAPTER 6. Attitudes toward Production; CHAPTER 7. Interpretation and Conclusion; APPENDIX: Bylaws of the Workers' Council; Index |
| Sommario/riassunto | <p><P>A pioneering examination of worker participation in the management of a textile factory in Lodz, Poland, this study now provides the first specific evidence to test the hypotheses based on the assumption that political ideology determines the character of management-labor relations.</P></p> |

| | |
|-------------------------|---|
| 2. Record Nr. | UNINA9910143394903321 |
| Autore | Carver Richard H. <1960-> |
| Titolo | Modern multithreading [[electronic resource]] : implementing, testing, and debugging multithreaded Java and C++/Pthreads/Win32 programs / / Richard H. Carver, Kuo-Chung Tai |
| Pubbl/distr/stampa | Hoboken, NJ, : Wiley, 2006 |
| ISBN | 1-280-27765-3 9786610277650 0-470-24456-9 0-471-74417-4 0-471-74416-6 |
| Descrizione fisica | 1 online resource (481 p.) |
| Altri autori (Persone) | TaiKuo-Chung |
| Disciplina | 005.1 005.11 |
| Soggetti | Parallel programming (Computer science) Threads (Computer programs) Electronic books. |
| Lingua di pubblicazione | Inglese |
| Formato | Materiale a stampa |
| Livello bibliografico | Monografia |
| Note generali | "Wiley-Interscience." |
| Nota di bibliografia | Includes bibliographical references and index. |
| Nota di contenuto | MODERN MULTITHREADING; CONTENTS; Preface; 1 Introduction to Concurrent Programming; 1.1 Processes and Threads: An Operating System's View; 1.2 Advantages of Multithreading; 1.3 Threads in Java; 1.4 Threads in Win32; 1.5 Pthreads; 1.6 C++ Thread Class; 1.6.1 C++ Class Thread for Win32; 1.6.2 C++ Class Thread for Pthreads; 1.7 Thread Communication; 1.7.1 Nondeterministic Execution Behavior; 1.7.2 Atomic Actions; 1.8 Testing and Debugging Multithreaded Programs; 1.8.1 Problems and Issues; 1.8.2 Class TDThread for Testing and Debugging 1.8.3 Tracing and Replaying Executions with Class Template sharedVariable1.9 Thread Synchronization; Further Reading; References; Exercises; 2 The Critical Section Problem; 2.1 Software Solutions to the Two-Thread Critical Section Problem; 2.1.1 Incorrect Solution 1; 2.1.2 Incorrect Solution 2; 2.1.3 Incorrect Solution 3; 2.1.4 Peterson's Algorithm; 2.1.5 Using the volatile Modifier; 2.2 Ticket- |

Based Solutions to the n-Thread Critical Section Problem; 2.2.1 Ticket Algorithm; 2.2.2 Bakery Algorithm; 2.3 Hardware Solutions to the n-Thread Critical Section Problem; 2.3.1 Partial Solution
2.3.2 Complete Solution 2.3.3 Note on Busy-Waiting; 2.4 Deadlock, Livelock, and Starvation; 2.4.1 Deadlock; 2.4.2 Livelock; 2.4.3 Starvation; 2.5 Tracing and Replay for Shared Variables; 2.5.1 ReadWrite-Sequences; 2.5.2 Alternative Definition of ReadWrite-Sequences; 2.5.3 Tracing and Replaying ReadWrite-Sequences; 2.5.4 Class Template sharedVariable; 2.5.5 Putting It All Together; 2.5.6 Note on Shared Memory Consistency; Further Reading; References; Exercises; 3 Semaphores and Locks; 3.1 Counting Semaphores; 3.2 Using Semaphores; 3.2.1 Resource Allocation; 3.2.2 More Semaphore Patterns 3.3 Binary Semaphores and Locks 3.4 Implementing Semaphores; 3.4.1 Implementing P() and V(); 3.4.2 VP() Operation; 3.5 Semaphore-Based Solutions to Concurrent Programming Problems; 3.5.1 Event Ordering; 3.5.2 Bounded Buffer; 3.5.3 Dining Philosophers; 3.5.4 Readers and Writers; 3.5.5 Simulating Counting Semaphores; 3.6 Semaphores and Locks in Java; 3.6.1 Class countingSemaphore; 3.6.2 Class mutexLock; 3.6.3 Class Semaphore; 3.6.4 Class ReentrantLock; 3.6.5 Example: Java Bounded Buffer; 3.7 Semaphores and Locks in Win32; 3.7.1 CRITICAL_SECTION; 3.7.2 Mutex; 3.7.3 Semaphore; 3.7.4 Events 3.7.5 Other Synchronization Functions 3.7.6 Example: C++/Win32 Bounded Buffer; 3.8 Semaphores and Locks in Pthreads; 3.8.1 Mutex; 3.8.2 Semaphore; 3.9 Another Note on Shared Memory Consistency; 3.10 Tracing, Testing, and Replay for Semaphores and Locks; 3.10.1 Nondeterministic Testing with the Lockset Algorithm; 3.10.2 Simple SYN-Sequences for Semaphores and Locks; 3.10.3 Tracing and Replaying Simple PV-Sequences and LockUnlock-Sequences; 3.10.4 Deadlock Detection; 3.10.5 Reachability Testing for Semaphores and Locks; 3.10.6 Putting It All Together; Further Reading; References; Exercises

4 Monitors

Sommario/riassunto

Master the essentials of concurrent programming, including testing and debugging. This textbook examines languages and libraries for multithreaded programming. Readers learn how to create threads in Java and C++, and develop essential concurrent programming and problem-solving skills. Moreover, the textbook sets itself apart from other comparable works by helping readers to become proficient in key testing and debugging techniques. Among the topics covered, readers are introduced to the relevant aspects of Java, the POSIX Pthreads library, and the Windows Win32 Applications Programming Interface.