

Synchronization Examples



Practice Exercises

- 7.1 Explain why Windows and Linux implement multiple locking mechanisms. Describe the circumstances under which they use spinlocks, mutex locks, semaphores, and condition variables. In each case, explain why the mechanism is needed.
- 7.2 Windows provides a lightweight synchronization tool called **slim reader–writer** locks. Whereas most implementations of reader–writer locks favor either readers or writers, or perhaps order waiting threads using a FIFO policy, slim reader–writer locks favor neither readers nor writers, nor are waiting threads ordered in a FIFO queue. Explain the benefits of providing such a synchronization tool.
- 7.3 Describe what changes would be necessary to the producer and consumer processes in Figure 7.1 and Figure 7.2 so that a mutex lock could be used instead of a binary semaphore.
- 7.4 Describe how deadlock is possible with the dining-philosophers problem.
- 7.5 Explain the difference between signaled and non-signaled states with Windows dispatcher objects.
- 7.6 Assume `val` is an atomic integer in a Linux system. What is the value of `val` after the following operations have been completed?

```
atomic_set(&val, 10);  
atomic_sub(8, &val);  
atomic_inc(&val);  
atomic_inc(&val);  
atomic_add(6, &val);  
atomic_sub(3, &val);
```

