Learning Objectives in this Part of the Module

- Appreciate the concept of semaphores
Learning Objectives in this Part of the Module

• Appreciate the concept of semaphores
• Recognize the two types of semaphores
Learning Objectives in this Part of the Module

• Appreciate the concept of semaphores
• Recognize the two types of semaphores
• Know a human known use of semaphores
Overview of Semaphores
Overview of Semaphores

• A semaphore is conceptually an “object” that can be atomically incremented & decremented to control access to a shared resource
A semaphore is conceptually an “object” that can be atomically incremented & decremented to control access to a shared resource

- e.g., originally used to control access to a shared railroad track

See en.wikipedia.org/wiki/Railway_semaphore_signal
Overview of Semaphores

- Concurrent programs use semaphores to coordinate interactions between multiple threads

See en.wikipedia.org/wiki/Semaphore_(programming)
Concurrent programs use semaphores to coordinate interactions between multiple threads, e.g.,

- A semaphore can control the access of threads to a limited # of resources

See [www.youtube.com/watch?v=D0DwWvYm12Y](http://www.youtube.com/watch?v=D0DwWvYm12Y)
Concurrent programs use semaphores to coordinate interactions between multiple threads, e.g.,

- A semaphore can control the access of threads to a limited # of resources
- It records a count ("permits") of how many units of a resource are available
Overview of Semaphores

• Concurrent programs use semaphores to coordinate interactions between multiple threads, e.g.,
  • A semaphore can control the access of threads to a limited # of resources
  • It records a count ("permits") of how many units of a resource are available
  • It provides operations to adjust the permit count atomically as units are acquired or released
Overview of Semaphores

- Concurrent programs use semaphores to coordinate interactions between multiple threads, e.g.,
  - A semaphore can control the access of threads to a limited # of resources
  - It records a count ("permits") of how many units of a resource are available
  - It provides operations to adjust the permit count atomically as units are acquired or released
Overview of Semaphores

- Concurrent programs use semaphores to coordinate interactions between multiple threads, e.g.,
  - A semaphore can control the access of threads to a limited # of resources
  - It records a count ("permits") of how many units of a resource are available
  - It provides operations to adjust the permit count atomically as units are acquired or released
  - Threads can wait (timed or blocking) until a unit of the resource is available
Concurrent programs use semaphores to coordinate interactions between multiple threads, e.g.,

- A semaphore can control the access of threads to a limited # of resources.
- It records a count (“permits”) of how many units of a resource are available.
- It provides operations to adjust the permit count atomically as units are acquired or released.
- Threads can wait (timed or blocking) until a unit of the resource is available.
- When a thread is done with a resource the permit count is incremented atomically & another waiting thread can acquire it.
Overview of Semaphores

• There are two types of semaphores
Overview of Semaphores

• There are two types of semaphores
  • **Counting semaphores**
    • Have # of permits defined by a counter (N) with precise meaning
There are two types of semaphores

- **Counting semaphores**
  - Have # of permits defined by a counter (N) with precise meaning
  - **Negative**
    - exactly -N threads queued waiting to acquire semaphore
Overview of Semaphores

• There are two types of semaphores
  • **Counting semaphores**
    • Have # of permits defined by a counter (N) with precise meaning
      • **Negative**
      • **Zero** == no waiting threads
        • an acquire operation will block the invoking thread until the counter N is positive
Overview of Semaphores

- There are two types of semaphores
  - **Counting semaphores**
    - Have # of permits defined by a counter (N) with precise meaning
      - **Negative**
      - **Zero** == no waiting threads
      - **Positive** == no waiting threads
        - an acquire operation will not block the invoking thread
Overview of Semaphores

- There are two types of semaphores
  - Counting semaphores
  - Binary semaphores
Overview of Semaphores

• There are two types of semaphores
  • Counting semaphores
  • **Binary semaphores**
    • Has only 2 states: acquired & not acquired
Overview of Semaphores

• There are two types of semaphores
  • Counting semaphores
  • **Binary semaphores**
    • Has only 2 states: acquired & not acquired
    • Restricts the counter N to the values 0 & 1
Overview of Semaphores

- We’ll analyze examples of both counting & binary semaphores later

E.g., the Palantiri gazing app use a counting semaphore
Overview of Semaphores

- We’ll analyze examples of both counting & binary semaphores later.

The Ping/Ping app uses a pair of binary semaphores.
Human Known Use of Semaphores
Human Known Uses of Semaphores

- A human known use of counting semaphores applies them to schedule access to beach volleyball courts

See en.wikipedia.org/wiki/Corona_del_Mar_State_Beach
Human Known Uses of Semaphores

- A human known use of counting semaphores applies them to schedule access to beach volleyball courts
- A bag full of balls is used to limit the number of teams that can concurrently play volleyball
End of Java Semaphores (Part 1)