Operating System (OS) CS232

Concurrency: Event based Concurrency

Dr. Muhammad Mobeen Movania

Outlines

- What is event-based concurrency?
- Why event-based concurrency?
- How to select events?
- Pros and cons of select and poll calls
- Summary

Event-based Concurrency

- Key Issues in Concurrency using Threads
  - Managing concurrency correctly in multi-threaded applications can be challenging
  - In a multi-threaded application, the developer has little or no control over what is scheduled at a given moment in time

- Key Idea
  - Can we do concurrency without threads?
    - Yes through the event loop

Event loop

- Key Idea
  - simply wait for something (i.e., an “event”) to occur
  - when it does, you check what type of event it is and do the small amount of work it requires

```
while (1) {
   events = getEvents();
   for (e in events)
      processEvent(e);
}
```

How to select events?

- Two approaches through two system calls
  - select()
  - poll()

```
int select(  int nfds,
       fd_set *restrict readfds,
       fd_set *restrict writefds,
       fd_set *restrict errorfds,
       struct timeval *restrict timeout);
```

Pros and Cons of select and poll

- No locks required
  - Only one event is being handled at a time
  - There is no need to acquire or release locks
  - The event-based server cannot be interrupted by another thread because it is decidedly single threaded.
- Blocking Systems Calls hamper performance
  - Solution asynchronous I/O
- State management has to be manual
  - All function parameters and local variables making the functions state have to be manually managed.

Summary

- We gave an introduction to a different style of concurrency based on events.
- Event-based servers give control of scheduling to the application itself
  - but do so at some cost in complexity and difficulty of integration with other aspects of modern systems.
- No single approach has emerged as best
  - both threads and events are likely to persist as two different approaches to the same concurrency problem for many years to come