Concurrency

Note Title 7/27/2010

Serializers

Mutex

test-and-set!

X = ~~5~~ ~~7~~ 6

(set! x (+ x 1))

Load X → 5

Add 1 to local copy → 6

Save local copy to X → 6

(set! x (+ x 2))

Load X → 5

Add 2 to local copy 7

Save local copy to X

```
(x-protector
  (set! x (+ x 1)))
```

```
(y-protector
  (set! y (+ y 1)))
```

Mutex:

```
(def m (make-mutex))
  (m 'acquire)
   (m 'release)

   (mutex 'acquire)
   (apply proc args)
    (mutex 'release)
(define (foo)
   (mutex 'acquire)
    ( let ((result (apply proc args))
         (mutex 'release)
          result))
```

→ Incorrect (No serialization)

→ Inefficiency (too much)

→ Deadlock

→ Unfairness