

ENGI 7894/9869

Baton Passing with Split Binary Semaphores

Presented by

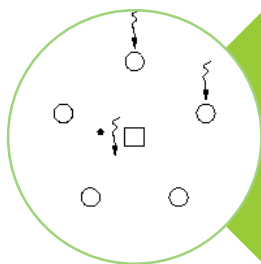
Md Monjur Ul Hasan

Question or Comments From Previous Class?

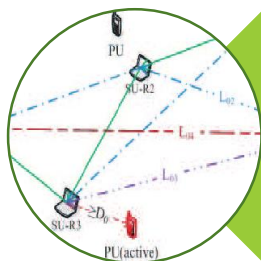
Topics

- Introduction
- Reader writer problem

Baton Passing on Reader Writer Problem



It is a technique implemented using split binary semaphores



Reduce the number of hops into and out of mutual exclusion



Increases control over allocation and reduces the cost of mutual exclusion

- When a thread enters a critical section (an await) it obtains mutual exclusion.
 - If it needs to delay, it gives up exclusive access and wait.
- When a thread finishes with the critical section it checks, to see if there is a thread waiting
 - If so, it wakes up that thread (give right) waiting but does not give up mutual exclusion. (Baton Passing).
 - It is up to the thread to give up the mutual exclusion or pass it to other thread (baton passing)
 - If not, it gives up exclusive access.

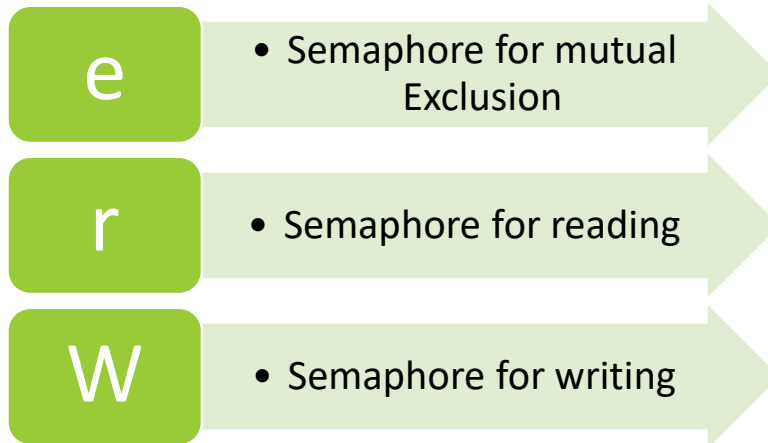
Reader Writer Global Invariant

- Can have only one writer
- Can have more than one reader
- Writer cannot proceed when reader available

$$RW : (nr = 0 \vee nw = 0) \wedge nw \leq 1$$

- There is one right (baton) that needs to run the resource access code.
- The rights to read or write on the resource are represented by the program counters, not by semaphores.

Signal Method: Split Binary Semaphores



Signal Method

```
int nw = 0; // number of writers
int nr = 0; // number of readers
```

```
Semaphore e = new Semaphore(1); // mutex
```

```
Semaphore r = new Semaphore(0); // delay readers
```

```
Semaphore w = new Semaphore(0); // delay writers
```

```
int dr = 0; // number of delayed readers
```

```
int dw = 0; // number of delayed writers
```

```
if (nw == 0 and dr > 0) {
    dr--;
    r.V();
} else if (nr == 0 && nw == 0 && dw > 0) {
    dw--;
    w.V();
} else {
    e.V();
}
```

Pass the baton to reader, if waiting reader is available

Pass the baton to writer, if waiting writer is available

Give up the mutual exclusion

Reader Writer Methods

```
e.P();  
if (nw > 0) {  
    dr++;  
    e.V();  
    r.P();  
}  
nr++;  
SIGNAL;
```

READ OPERATION

```
e.P();  
nr--;  
  
SIGNAL;
```

```
e.P();  
if ((nr > 0) || (nw > 0)) {  
    dw++;  
    e.V();  
    w.P();  
}  
nw++;  
SIGNAL;
```

WRITE OPERATION

```
e.P();  
nw--;  
  
SIGNAL;
```

Thank you for your attention

Any Questions?