Today

• Questions compiling / running Linda on cluster?

• Semaphores in Linda
  • binary, counting

• Producer / Consumer examples
  • with Semaphores
Producer/Consumer

• Eval two processes: Producer and Consumer

• Each process has its own array of n elements.

• Between the two processes, a shared buffer exists that will be used to transfer the contents of the producer’s buffer to the consumer’s buffer, one element at a time
Producer/Consumer using shared variables

Here’s the pseudo code for producer and consumer:

```cpp
//shared variables
int buf, n = 80, p = 0, c = 0;

process Producer {
    int a[n];
    while (p < n) {
        << await (p == c); >>
        buf = a[p];
        p = p+1;
    }
}

process Consumer {
    int b[n];
    while (c < n) {
        << await (p > c); >>
        b[c] = buf;
        c = c+1;
    }
}
```
Semaphores in Linda

- P(s): `in(“sem”)`
  - attempts to match/remove a one-field tuple in TS
- V(s): `out(“sem”)`
  - places a one-field tuple in TS
- For multiple semaphores
  - you decide how to implement...
Producers/Consumers using semaphores

- Here's the pseudo code for producer and consumer procs:

```plaintext
process Producer(i) {
    while (true) {
        . . .
        // produce data,
        // deposit in buf.
        P(empty);
        buf = data;
        V(full);
    }
}

process Consumer(i) {
    while (true) {
        . . .
        // fetch data from buf,
        // and consume it.
        P(full);
        result = buf;
        V(empty);
    }
}
```

//shared variables
int buf;
sem empty = 1;
sem full = 0; //binary semaphores: 0 or 1
Bounded Buffer using semaphores

• Here’s the pseudo code for producer and consumer procs:

```c
//shared variables
int buf[n],  //counting semaphores
int front = 0, rear = 0;  //range from 0 to n
sem empty = n, full = 0;

process Producer {
    while (true) {
        . . .
        // produce data,
        // deposit in buf.
        P(empty);
        buf[rear] = data;
        rear = (rear+1)%n;
        V(full);
    }
}

process Consumer {
    while (true) {
        . . .
        //fetch data from buf,
        //and consume it.
        P(full);
        result = buf[front];
        front = (front+1)%n;
        V(empty);
        . . .
    }
}
```
Programming Assignment
Due: ?

• Implement C-Linda versions of the producer/consumer and bounded buffer problems using semaphores.

• Replace the “...” with print statements indicating who is producing / consuming what.

• Your program should run under CDS and LINDA-TCP
Question

• How would you handle a bounded buffer with multiple producers and consumers?