Announcements

• Program #2
  - On the web, due in one week
Readers/Writers Problem

- Data area shared by processors
- Some processes read data, others write data
  - Any number of readers may simultaneously read the data
  - Only one writer at a time may write
  - If a writer is writing to the file, no reader may read it
- Two of the possible approaches
  - Readers have priority or writers have priority
Readers have Priority

Semaphore wsem = 1, x = 1;
reader()
{
    repeat
        P(x);
        readcount = readcount + 1;
        if readcount = 1 then P (wsem);
        V(x);
        READUNIT;
        P(x);
        readcount = readcount - 1;
        if readcount = 0 V(wsem);
        V(x);
    forever
};

writer()
{
    repeat
        P(wsem);
        WRITEUNIT;
        V(wsem)
    forever
}
Comments on Reader Priority

- semaphores x, wsem are initialized to 1
- note that readers have priority - a writer can gain access to the data only if there are no readers (i.e. when readcount is zero, signal(wsem) executes)
- possibility of starvation - writers may never gain access to data
Writers Have Priority

**reader**

repeat
  P(z);
  P(rsem);
  P(x);
  readcount++;
  if (readcount == 1) then
    P(wsem);
  V(x);
  V(rsem);
  V(z);
  **readunit**;
  P(x);
  readcount- -;
  if readcount == 0 then
    V (wsem)
  V(x)
forever

**writer**

repeat
  P(y);
  writecount++:
  if writecount == 1 then
    P(rsem);
  V(y);
  P(wsem);
  **writeunit**
  V(wsem);
  P(y);
  writecount--;
  if (writecount == 0) then
    V(rsem);
  V(y);
forever;
Notes on readers/writers with writers getting priority

Semaphores \(x, y, z, w\text{sem}, r\text{sem}\) are initialized to 1

Readers queue up on semaphore \(z\); this way only a single reader queues on \(r\text{sem}\). When a writer signals \(r\text{sem}\), only a single reader is allowed through.

\[
\begin{align*}
P(z); \\
P(r\text{sem}); \\
P(x); \\
\text{readcount}++; \\
\text{if (readcount}==1) \text{ then} \\
P(w\text{sem}); \\
V(x); \\
V(r\text{sem}); \\
V(z);
\end{align*}
\]
Sample Synchronization Problem

- **Class Exercise:**
  - **CMSC 412 Midterm #1 (Spring 1998) Q#3**
  - Solution posted at: