Semaphores

A variable that has an integer value upon which only three operations are defined:

- There is no way to inspect or manipulate semaphores other than these three operations

1) A semaphore may be initialized to a nonnegative integer value
2) The semWait operation decrements the semaphore value
3) The semSignal operation increments the semaphore value

*Figure 5.6 A Definition of Semaphore Primitives*
A queue is used to hold processes waiting on the semaphore

- The process that has been blocked the longest is released from the queue first (FIFO)
- The order in which processes are removed from the queue is not specified

Figure 5.7 A Definition of Binary Semaphore Primitives

Strong/Weak Semaphores

Figure 5.9 Mutual Exclusion Using Semaphores

/* program mutualexclusion */
const int n = /* number of processes */;
semaphore s = 1;
void P(int i)
{
    while (true) {
        semWait(s);
        /* critical section */;
        semSignal(s);
        /* remainder */;
    }
} 
void main()
{
    parbegin (P(1), P(2), ..., P(n));
}
Producer/Consumer Problem

**General Statement:**
One or more producers are generating data and placing these in a buffer.

A single consumer is taking items out of the buffer one at a time.

Only one producer or consumer may access the buffer at any one time.

**The Problem:**
Ensure that the producer won’t try to add data into the buffer if its full, and that the consumer won’t try to remove data from an empty buffer.

---

```
/* program producerconsumer */
int a = 1, delay = 0;

void producer()
{
    while (true)
    {
        produce();
        semaphore();
        append();
        n[3] = 1; semaphore();
    }
}

void consumer()
{
    semaphore();
    while (true)
    {
        consume();
        semaphore();
        take();
        if (n[2] == 0) semaphore();
    }
}

void main()
{
    n[0] = 0;
    parbegin (producer, consumer);
}
```
Table 5.4
Possible Scenario for the Program of Figure 5.12

<table>
<thead>
<tr>
<th></th>
<th>Producer</th>
<th>Consumer</th>
<th>n</th>
<th>d</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Note: White areas represent the critical section controlled by semaphore s.

© 2017 Pearson Education, Inc., Hoboken, NJ. All rights reserved.