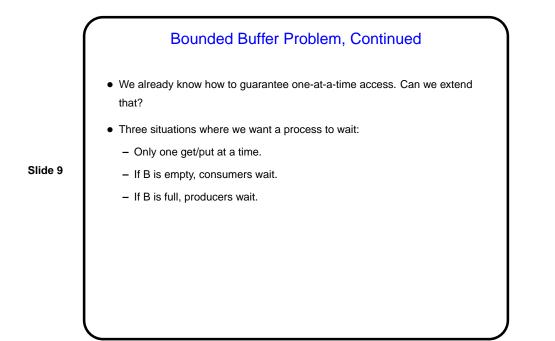
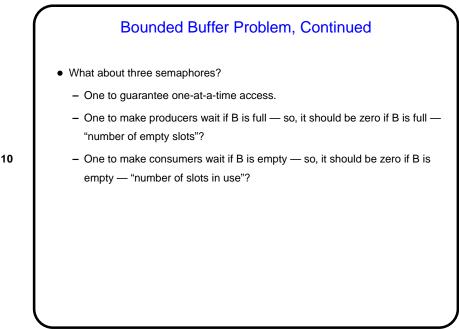
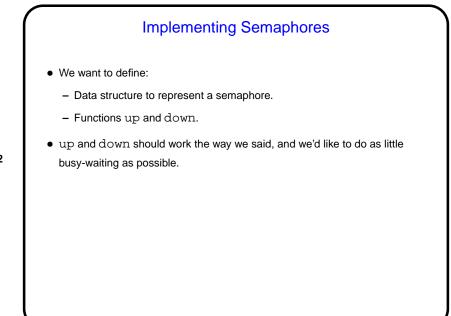


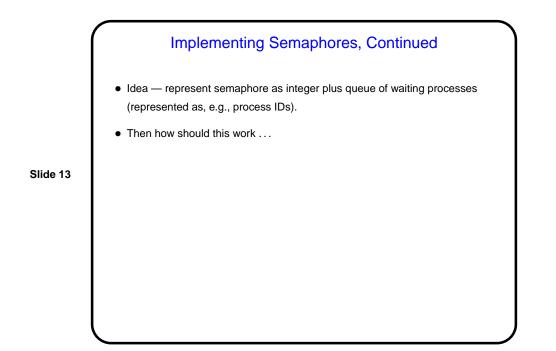
Shared variables:		
buffer B(N); // initially empty, can hold N thing:		
Pseudocode for producer:	Pseudocode for consumer:	
while (true) {	while (true) {	
<pre>item = generate();</pre>	<pre>item = get(B);</pre>	
<pre>put(item, B);</pre>	use(item);	
}	}	
Synchronization requirements:		
1. At most one process at a time accessing buffer.		
2. Never try to get from an empty buffer or put to a full one.		
3. Processes only block if they "have to".		

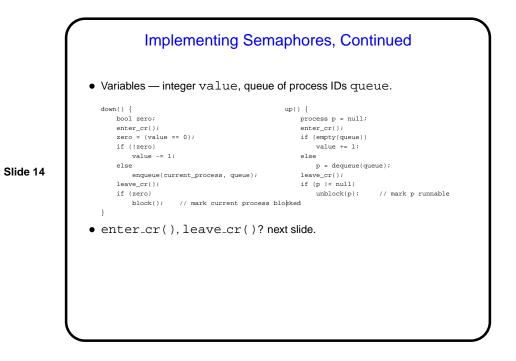


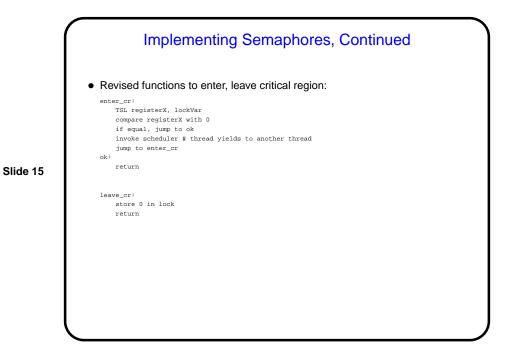


	Bounded Buffer Problem — Solution	
	<ul> <li>Shared variables: buffer B(N); // empt semaphore mutex(1); semaphore empty(N); semaphore full(0);</li> </ul>	y, capacity N
Slide 11	<pre>Pseudocode for producer: while (true) { item = generate(); down(empty); down(mutex); put(item, B); up(mutex); up(full); }</pre>	<pre>Pseudocode for consumer: while (true) { down(full); down(mutex); item = get(B); up(mutex); up(empty); use(item); }</pre>









 Minute Essay

 • Tell me about your experience (if any!) with writing programs that involve concurrency — multithreaded, message-passing, communicating over sockets, etc.

 • What (if anything) did you find interesting, difficult, or otherwise noteworthy about Homework 1?