

































OTTERY SCHEDULER IMPLEMENTATION		
	head $\longrightarrow$ $(JobA \ Tic 50)$ $\longrightarrow$ $(JobB \ Tic 50)$ $\longrightarrow$ $(JobC \ Tic 250)$ $\longrightarrow$ NULL	
1 2 3	// counter: used to track if we've found the winner yet int counter = $0_{\rm J}$	
4 5 6	<pre>// winner: use some call to a random number generator to // get a value, between 0 and the total # of tickets int winner = getradom(0, totaltickets);</pre>	
9 10	<pre>// current: use this to walk through the list of jobs node_t *current = head;</pre>	
11 12	<pre>// loop until the sum of ticket values is &gt; the winner while (current) {</pre>	
14 15 16	if (counter > winner) break; // found the winner	
17	} // 'current' is the winner: schedule it	

TICKET MECHANISMS			
<ul> <li>Ticket currency / exchange</li> <li>User allocates tickets in any desired way</li> <li>OS converts user currency into global currency</li> <li>Example:</li> <li>There are 200 global tickets assigned by the OS</li> </ul>			
User A $\rightarrow$ 500 (A's currency) to A1 $\rightarrow$ 50 (global currency) $\rightarrow$ 500 (A's currency) to A2 $\rightarrow$ 50 (global currency) User B $\rightarrow$ 10 (B's currency) to B1 $\rightarrow$ 100 (global currency)			
April 11, 2017 TCSS422: Operating Systems [Spring 2017] Institute of Technology, University of Washington - Tacoma	L4.20		



























