



















PASSING A SINGLE VALUE Using this approach on your Ubuntu VM, How large (in bytes) can the primitive data type be?		
9 int rc, 10 pthread 11 pthread 12 printf 13 return 14 }	<pre>, m; d_create(&amp;p, NULL, mythread, (void *) 100); d_join(p, (void **) &amp;m); ("returned %d\n", m); 0;</pre>	
February 6, 2019	TCSS422: Operating Systems [Winter 2019] School of Engineering and Technology, University of Washington - Tacoma	L8.11



























PTHREADS LIBRARY		
<ul> <li>Compilation</li> <li>gcc -pthread</li> <li>Requires exp</li> <li>Use makefile</li> <li>List of pthrea</li> <li>man -k pthree</li> </ul>	pthread.c – o pthread licitly linking the library with compiler flag to provide compiler arguments d manpages ead	
February 6, 2019	TCSS422: Operating Systems [Winter 2019] School of Engineering and Technology, University of Washington - Tacoma	L8.25

















rsity of Washington - Tacoma

TCSS422: Operating Systems [Winter 2019] School of Engineering and Technology, Univer-

February 6, 2019





L8.34

SPIN LOCK IMPLEMENTATION				
<ul> <li>Operate</li> <li>"Do-it-yo</li> <li>Is this lo</li> </ul>	without atomic- <i>as a unit</i> assembly instructions urself" Locks ck implementation: <b>(1)Correct? (2)Fair? (3)Performant?</b>			
	<pre>1 typedef struct _lock_t { int flag; } lock_t; 2 void init(lock t *mutex) { 4  // 0 &gt; lock is available, 1 &gt; held 5  mutex&gt;sflag = 0; 6 } 7 void lock(lock_t *mutex) { 9  void lock(lock_t *mutex) { 9  mutex&gt;sflag = 1; // new set it ! 12 } 13 14 void unlock(lock_t *mutex) { 15  mutex&gt;sflag = 0; 16 } 17 18 19 10 10 10 10 10 10 10 10</pre>			
February 6, 20	19         TCSS422: Operating Systems [Winter 2019] School of Engineering and Technology, University of Washington - Tacoma         L8.37			

Ihread1	Thread2
<pre>call lock() while (flag == 1) interrupt switch to Thread 2</pre>	call lock() while (flag == 1) flag = 1; interrupt: switch to Thread 1
flag = 1; // set flag to 1 (too!)	





















OBJECTIVES		
Chapter 29 • Concurrent • Performan • Lock Grant	t Data Structures ice ularity	
February 6, 2019	TCSS422: Operating Systems [Winter 2019] School of Engineering and Technology, University of Washington - Tacoma	L8.49











PERFECT SCALING		
Achieve (N) pe	erformance gain with (N) additional resources	5
<ul> <li>Throughput:</li> <li>Transactions p</li> </ul>	per second	
<ul> <li>1 core</li> <li>N = 100 tps</li> </ul>		
<ul> <li>10 core</li> <li>N = 1000 tps</li> </ul>		
February 6, 2019	TCSS422: Operating Systems [Winter 2019] School of Engineering and Technology, University of Washington - Tacoma	L8.55



