

OBJECTIVES				
Non-deadloc	k concurrency bugs			
Deadlock causes				
Deadlock prevention				
February 15, 2017	TCSS422: Operating Systems [Winter 2017] Institute of Technology, University of Washington - Tacoma			









ORDER VIOLATION BUGS					
 Desired order between memory accesses is flipped E.g. something is checked before it is set Example: 					
<pre> 1</pre>	<pre>sad::: i init() { mThread = PR_CreateThread(mMain,); sad2:: i mMain() { mstate = mThread->State hread is not initialized?</pre>				
February 15, 2017	TCSS422: Operating Systems [Winter 2017] Institute of Technology, University of Washington - Tacoma	L11.7			













Condition	Description
Mutual Exclusion	Threads claim exclusive control of resources that they require.
Hold-and-wait	Threads hold resources allocated to them while waiting for additional resources
No preemption	Resources cannot be forcibly removed from threads that are holding them.
Circular wait	There exists a circular chain of threads such that each thread holds one more resources that are being requested by the next thread in the chain



















INTELLIGENT SCHEDULING - 2						
Scheduler produces schedule:						
CPU 1 T3 T4 CPU 2 T1 T2						
		T1 T2	T3	T4		
	L1 L2	yes yes yes yes	yes yes	no no		
February 15, 2017	TCSS422: Operation	ng Systems [Winter 20: iology, University of Wa	17] shington - Tacomi	3	111.25	

INTELLIGENT SCHEDULING - 3			
Scheduler produces schedule			
	CPU 1 T4 CPU 2 T1 T2 T3		
 Scheduler must be conservative and not task risks Slows down execution - many threads 			
There has been limited use of these approaches given the difficulty having intimate lock knowledge about every thread			
February 15, 2017	TCSS422: Operating Systems [Winter 2017] Institute of Technology, University of Washington - Tacoma		



