Deadlock 2: Still stuck!

10/14/15
Parallel quicksort lab

- OpenMP sections
- Running out of threads
- Tracking recursion depth (not a global variable)
- Why so many threads?
Recall: Deadlocks

• Graph-based detection algorithm for unique resources

• Matrix-based algorithm to determine safety for duplicated resources

• Ostrich Algorithm
Strategy 2: Detect and recover

- Periodically run a deadlock detection algorithm and kill an involved process if one is found
Strategy 3: Deadlock prevention

• Deny one of the conditions needed for deadlock:
  – Mutual exclusion: Resources assigned to $\leq 1$ process at a time
  – Hold and wait: Process can hold resources while waiting for more
  – No preemption: System can’t take resources back
  – Circular wait: There can be a circular chain of processes waiting for each other’s resources
Which of the conditions is prevented by the scheme below?

Two-phase locking: Acquire all locks at once; release and backoff if you can’t. Then perform critical section and release everything

A. Mutual exclusion
B. Hold and wait
C. No preemption
D. Circular waiting
E. None of the above; deadlock can still occur
Which of the conditions is prevented by the scheme below?

Two-phase locking: Acquire all locks at once; release and backoff if you can’t. Then perform critical section and release everything

A. Mutual exclusion
B. Hold and wait
C. No preemption
D. Circular waiting
E. None of the above; deadlock can still occur
Which of the conditions is prevented by the scheme below?

Assign a number to every resource and require that processes request resources in order

A. Mutual exclusion
B. Hold and wait
C. No preemption
D. Circular waiting
E. None of the above; deadlock can still occur
Which of the conditions is prevented by the scheme below?

Assign a number to every resource and require that processes request resources in order

A. Mutual exclusion
B. Hold and wait
C. No preemption
D. Circular waiting
E. None of the above; deadlock can still occur
Which of the conditions is prevented by the scheme below?

Spooling: Instead of blocking for a resource such as a printer, write the operation to a buffer which the OS handles when possible

A. Mutual exclusion
B. Hold and wait
C. No preemption
D. Circular waiting
E. None of the above; deadlock can still occur
Which of the conditions is prevented by the scheme below?

Spooling: Instead of blocking for a resource such as a printer, write the operation to a buffer which the OS handles when possible

A. Mutual exclusion
B. Hold and wait
C. No preemption
D. Circular waiting
E. None of the above; deadlock can still occur
Which of the conditions is prevented by the scheme below?

Require processes to request all needed resources when they ask to start and only start processes whose needs can be met (more appropriate for hardware resources than locks...)

A. Mutual exclusion
B. Hold and wait
C. No preemption
D. Circular waiting
E. None of the above; deadlock can still occur
Which of the conditions is prevented by the scheme below?

Require processes to request all needed resources when they ask to start and only start processes whose needs can be met (more appropriate for hardware resources than locks...)

A. Mutual exclusion
B. Hold and wait
C. No preemption
D. Circular waiting
E. None of the above; deadlock can still occur
Strategy 4: Deadlock avoidance ("Banker’s Algorithm")

• Eliminate possibility of deadlock through clever resource allocation

• For each resource request, evaluate if granting it puts system into an unsafe state (only grant it if not)