

1. Condition Variables (Code Review)

Is it necessary for the change method to lock the mutex, to release a blocked thread? Why is 'if' incorrect?

<pre>void change() { x = 1; pthread_cond_signal(&cv); }</pre>	<pre>void wait_for_positive_x() { pthread_mutex_lock(&m); if(x < 1) pthread_cond_wait(&cv, &m); pthread_mutex_unlock(&m); }</pre>
---	---

... Implications for cond_wait implementation?

2. What is Livelock?

3. Deadlock conditions

3. Deadlock

The _____ conditions for deadlock are:

_____: "A process is currently holding at least one resource and requesting additional resources which are being held by other processes."

_____: "There is a set of waiting processes, such that P_1 is waiting for a resource held by P_2 , P_2 is waiting for a resource held by P_3 and so on until P_N is waiting for a resource held by P_1 ."

_____: "A resource can be released only voluntarily by the process holding it, after that process has completed its task"

_____: "At least one resource must be held in a non-shareable mode"

4. Deadlock (applied)

Three gardeners visit the garden shed pick up their desired tools for the day. There is a potential for deadlock. Fortunately they know about the C_____ conditions! Find four ways to solve the problem (break one condition each time). Name which condition you break in each case.

1

2

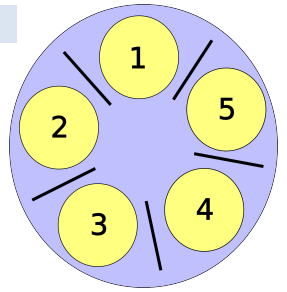
3

4

5. Think concurrently!

Remember (for example) Mergesort? How can you implement parallel Mergesort? Explain what synchronization calls you will use and when.

6. What is the "Dining Philosophers" problem?



Candidate Solutions:

1. "Pick up left chopstick. Pickup right chopstick. Eat. Release both."
2. "Pick up right. Pick up left. Eat. Release both"
3. "Eat when I tell you"
4. "Pick up left chopstick. Try to pickup right chopstick (Fail? release both and restart). Eat. Release both."
- 5?