

Lecture 17 — Deadlock Detection & Recovery

Jeff Zarnett

Deadlock Detection

Thus far we have examined several ways to prevent or avoid deadlock, but all of those solutions have come with significant drawbacks or limitations. Avoidance analyses are also conservative; they will prevent a request from taking place if there is even a small chance it could lead to a deadlock. If we cannot stop deadlock from happening or cannot live with the performance reduction that avoidance mandates, then perhaps the next best thing is to let all resource requests proceed and then determine later if a deadlock exists, and if so, do something about it.

Deadlock Recovery

Once a deadlock has been detected, a system can recover from that deadlock by “breaking” the deadlock. There are several strategies that we could apply, which we will refer to in humorous terms as:

- Robbery
- Murder
- Mass Murder
- Time travel; and
- Armageddon.

Robbery

Murder

Mass Murder

Time Travel

Armageddon

Miscarriages of Justice

It turns out that our deadlock detection algorithms do not have to be perfect if we have chosen an appropriate recovery strategy.