Race Condition

\[
\text{balance} \quad \begin{array}{c}
0 \\
10 \\
15
\end{array}
\]

Thread 1
\[ n_5 = 0 + 10 \]
\[ \text{assign} \ n_5 \rightarrow b \]

Thread 2
\[ n_5 = 0 + 15 \]
\[ \text{assign} \ n_5 \rightarrow b \]
public void deposit(int amt) {
    balance += amt;
}

// balance = balance + amt.

// This doesn't work either

assertment is not 5/10/12
balance = balance + \text{amt}.

\begin{align*}
\text{mov} & \quad \text{balance}, r1 \\
\text{mov} & \quad \text{amt}, r2 \\
\text{add} & \quad r1, r2, r1 \\
\text{st}, & \quad r1, \text{balance}
\end{align*}
Solution: make sure that only 1 thread executes deposit @ a time.
Deadlock (in pictures)

File A

Process A
lock(file A)
lock(file B)

Process B
lock(file B)
lock(file A)
(Subtle) Race Condition

web-based system to reserve conference rooms

Title: Apr 2-10:03 AM (6 of 6)