C/C++ Concurrency
Formal Model

Axiomatic model

Program = set of consistent executions

Execution = \langle E, po, rf, mo, \ldots \rangle

- E: events
- po: program-order
- rf: reads-from
- mo: modification-order

Derived relations

- Synchronized-with (sw)
- Happens-before (hb)
- \ldots
Synchronized-With Relation

\[ W_{\text{rel}}(X, 1) \xrightarrow{\text{sw}} R_{\text{acq}}(X, 1) \]

\[ W_{\text{rel}}(X, 1) \xrightarrow{\text{rf}} W_{\text{rlx}}(X, 2) \xrightarrow{\text{sw}} R_{\text{acq}}(X, 2) \]

\[ F_{\text{rel}} \xrightarrow{\text{sw}} W_{\text{rlx}}(X, 2) \xrightarrow{\text{rf}} R_{\text{acq}}(X, 2) \]

\[ W_{\text{rel}}(X, 1) \xrightarrow{\text{rf}} W_{\text{rlx}}(X, 1) \xrightarrow{\text{sw}} F_{\text{acq}} \]

\[ F_{\text{rel}} \xrightarrow{\text{rf}} R_{\text{rlx}}(X, 1) \]

\[ W_{\text{rlx}}(X, 1) \xrightarrow{\text{sw}} F_{\text{acq}} \]
Synchronization-With and Happens-Before

Release sequence: \( rs \triangleq [WU]; \, po \mid[\text{Loc}]; [WU \sqsupseteq \text{rlx}]; (rf; U)^* \)

Synchronization with:

\( sw \triangleq [E \sqsupseteq \text{rel}]; ([F]; po)?; \, rs; \, rf; \, [R \sqsupseteq \text{rlx}]; (po; [F])?; [E \sqsupseteq \text{acq}] \)

Happens-before: \( hb \triangleq (po \cup sw)^+ \)
Study the relevant examples from [POPL’11], [POPL’15].