
Concurrent Programming

Blatt 11

Abgabe: 15.7.2002

1. [15 Punkte] Erweitere das Engine-System um die Operation `obtain-lock-multiple` für selektives Warten auf Locks. `Obtain-lock-multiple` erhält als Argument eine Liste von Locks, auf die gewartet werden soll. `Obtain-lock-multiple` blockiert so lange, bis eines der Locks frei wird. Dieses Lock ist der Rückgabewert von `obtain-lock-multiple`. Programmiere außerdem ein kleines Testprogramm, das `obtain-lock-multiple` verwendet.
2. [15 Punkte] Erweitere das Engine-Systems mit Schachtelung um die optimistische Nebenläufigkeit mit Proposals! Verwende dazu keine Locks.

Anleitung:

- Assoziiere mit jedem Thread ein Proposal. Eine Prozedur `current-proposal` gibt es zurück und `set-current-proposal!` setzt es.
- Implementiere eine Repräsentation für Proposals mit einem Konstruktor `make-proposal` sowie Prozeduren `find-log-entry` und `add-log-entry!`, die Einträge im Proposal suchen bzw. neue hinzufügen. Jeder Eintrag sollte folgende Informationen enthalten: das Paar, das gelesen oder beschrieben wurde; ob es sich bei dem Zugriff um den `car` oder `cdr` handelte; der Wert, der sich zum Zugriffszeitpunkt dort befand; der Wert, der bei erfolgreichem Commit dort hineingeschrieben werden soll.
- Implementiere `provisional-car`, `provisional-cdr`, `provisional-set-car!` und `provisional-set-cdr!`.
- Implementiere `maybe-commit`.

Programmiere zu Testzwecken eine Version des Zählers für das optimistische Engine-System.