

Unstable-Funcons-beta: Unstable-Funcons-Index *

The PLaNCompS Project

Unstable-Funcons-Index.cbs | PLAIN | PRETTY

OUTLINE

Computations

- Normal computation
 - Value indexing
 - Memos
- Abnormal computation
 - Postponing
- Threads
 - Multithreading
 - Initialisation
 - Activation
 - Execution
 - Termination
 - Scheduling
 - Thread synchronisation
 - Syncs
 - Sync features
 - Locks
 - Notifications

Unstable Funcons, for use only in Unstable Languages

Computations

Normal computation

Value indexing

```
[ Entity value-index
  Funcon initialise-index
  Funcon allocate-index
  Funcon lookup-index ]
```

Memos

```
[ Entity memo-map
  Funcon initialise-memos
  Funcon memo-value
  Funcon initialise-memo-value
  Funcon memo-value-recall ]
```

*Suggestions for improvement: plancomps@gmail.com.
Reports of issues: <https://github.com/plancomps/CBS-beta/issues>.

Abnormal computation

Postponing

[*Entity* postpone
Funcon postpone
Funcon postpone-after-effect
Funcon after-effect]

Threads

Multithreading

[*Datatype* thread-ids
Datatype threads
Funcon thread-joinable
Funcon thread-detached]

Initialisation

[*Funcon* initialise-multithreading]

Activation

[*Funcon* multithread
Funcon thread-activate
Funcon thread-detach]

Execution

[*Funcon* current-thread
Funcon thread-atomic
Funcon thread-yield
Funcon thread-spin
Funcon thread-suspend
Funcon thread-resume]

Termination

[*Funcon* thread-terminate
Funcon is-thread-terminated
Funcon thread-value
Funcon thread-join
Funcon thread-exterminate]

Scheduling

[*Funcon* update-thread-stepping
Funcon update-thread-schedule
Funcon current-thread-schedule
Funcon is-thread-preemptible
Funcon thread-preemptible
Funcon thread-cooperative]

Thread synchronisation

Syncs

```
[ Datatype syncs
  Funcon sync-create
  Funcon sync-feature
  Funcon is-sync-feature ]
```

Sync features

```
[ Datatype sync-features
  Funcon sync-waiting-list
  Funcon sync-held
  Funcon sync-holder
  Funcon sync-count
  Funcon sync-feature-create ]
```

Locks

```
[ Funcon is-exclusive-lock-holder ]
```

Spin locks

```
[ Funcon spin-lock-create
  Funcon spin-lock-sync
  Funcon spin-lock-release ]
```

Exclusive locks

```
[ Funcon exclusive-lock-create
  Funcon exclusive-lock-sync
  Funcon exclusive-lock-sync-else-wait
  Funcon exclusive-lock-release ]
```

Reentrant locks

```
[ Funcon reentrant-lock-create
  Funcon reentrant-lock-sync
  Funcon reentrant-lock-sync-else-wait
  Funcon reentrant-lock-release
  Funcon reentrant-lock-exit ]
```

Semaphores

```
[ Funcon semaphore-create
  Funcon semaphore-sync
  Funcon semaphore-sync-else-wait
  Funcon semaphore-release ]
```

Shared-exclusive locks

```
[ Funcon rw-lock-create
  Funcon rw-lock-sync-exclusive
  Funcon rw-lock-sync-shared
  Funcon rw-lock-sync-exclusive-else-wait
  Funcon rw-lock-sync-shared-else-wait
  Funcon rw-lock-release-exclusive
  Funcon rw-lock-release-shared ]
```

Notifications

Barriers

[*Funcon* barrier-create
Funcon barrier-sync
Funcon barrier-sync-else-wait]

Conditions

[*Funcon* condition-create
Funcon condition-wait
Funcon condition-wait-with-lock
Funcon condition-notify-all
Funcon condition-notify-first]

Rendezvous

[*Funcon* rendezvous-create
Funcon rendezvous-sync
Funcon rendezvous-sync-else-wait]