

多线程

aborn

2018-05-29

Contents

1 多线程	1
2 基本的线程相关函数	1
2.1 创建线程	1
2.2 thread-join	1
2.3 thread-yield	2
2.4 获取线程名	2
2.5 线程状态	2
2.6 当前线程	2
2.7 所有线程列表	2
3 互斥锁 (Mutexes)	2
3.1 创建一个互斥锁	2

1 多线程

Emacs 从 26.1 版本开始引入了多线程。它提供了一种简单（但功能有限）多线程操作。跟其他编程语言一样，在同一个 Emacs 实例里所有的线程的内存是共享的。每个线程有其自己运行 Buffer(Current Buffer) 和对应的数据 (Match Data)。

2 基本的线程相关函数

下面介绍线程操作相关的基本函数。

2.1 创建线程

我们可以通过 **make-thread** 函数来创建线程并执行对应的 task。它的语法如下：

```
(make-thread function &optional name)
```

创建一个名为 name 的线程，该线程执行 function 函数，当函数执行结束后，退出该线程。新线程的 Current Buffer 继承当前 Buffer，这个函数返回一个线程对象。可以通过 (**threadp object**) 来判断一个对象是否为线程对象。

2.2 thread-join

thread-join，它阻塞当前执行直到线程执行完成，如果线程已经退出，它立刻返回。

```
(thread-join thread)
```

2.3 thread-yield

执行下一个可执行的线程。

2.4 获取线程名

可以通过 (thread-name thread) 函数来获取线程名。

2.5 线程状态

判断一个线程是否还在执行 (alive)，可以用 (thread-alive-p thread)。

2.6 当前线程

(current-thread) 返回当前线程。

2.7 所有线程列表

获取当前所有正在运行中的线程 (all-threads)。

3 互斥锁 (Mutexes)

互斥一种排它锁 (exclusive lock)，在任何时刻，最多只允许一个线程持有互斥锁。也就是说，当一个线程试图获取一个已经被其他线程持有的互斥锁时，它会引发阻塞，直到该互斥锁被释放为止。

3.1 创建一个互斥锁

创建一个互斥锁对象，采用 **make-mutex** 函数，该函数返回一个互斥锁对象，其名字为 `name`。

```
(make-mutex &optional name)
```

判断一个对象是否为互斥锁使用 (`mutexp object`)。