

在使用POSIX线程（`pthread`）的环境中，你可以使用 `pthread_self()` 函数来获取当前线程的线程ID。`pthread_self()` 返回的是一个类型为 `pthread_t` 的值，它表示当前线程的ID。

以下是一个示例，演示如何使用 `pthread_self()` 来获取线程ID：

```
1  #include <pthread.h>
2  #include <stdio.h>
3
4  void* printThreadId(void* arg) {
5      pthread_t threadId = pthread_self();
6      printf("Current thread ID: %lu\n", (unsigned long)threadId);
7      return NULL;
8  }
9
10 int main() {
11     pthread_t thread;
12
13     // 打印主线程的ID
14     printThreadId(NULL);
15
16     // 创建一个新线程，并打印其线程ID
17     pthread_create(&thread, NULL, printThreadId, NULL);
18     pthread_join(thread, NULL);
19
20     return 0;
21 }
```

在这个示例中：

1. `printThreadId` 函数使用 `pthread_self()` 获取当前线程的ID，并将其打印出来。
2. 在 `main` 函数中，首先打印主线程的ID。
3. 然后创建一个新线程，该线程在启动时会调用 `printThreadId` 函数，打印新线程的ID。
4. 使用 `pthread_join` 等待新线程完成。

注意：`pthread_t` 类型的实际表示方式取决于具体的实现。在某些实现中，它可能是一个整数类型，而在其他实现中，它可能是一个指针或结构体。因此，在打印时，我们将其强制转换为 `unsigned long` 来确保可读性和移植性。

通过这种方式，你可以获取并打印出每个线程的ID，方便调试和日志记录。