线程不安全函数	线程不安全类	Linux 线程安全版本
rand	2	rand_r
strtok	2	strtok_r
asctime	3	asctime_r
ctime	3	ctime_r
gethostbyaddr	3	gethostbyaddr_r
gethostbyname	3	gethostbyname_r
inet_ntoa	3	(无)
localtime	3	localtime_r

图 12-41 常见的线程不安全的库函数

因此，Linux 系统提供大多数线程不安全函数的可重入版本。可重入版本的名字总是以“_r”后缀结尾。例如，asctime 的可重入版本就叫做 asctime_r。我们建议尽可能地使用这些函数。

12.7.4 竞争

当一个程序的正确性依赖于一个线程要在另一个线程到达 y 点之前到达它的控制流中的 x 点时，就会发生竞争(race)。通常发生竞争是因为程序员假定线程将按照某种特殊的轨迹线穿过执行状态空间，而忘记了另一条准则规定：多线程的程序必须对任何可行的轨迹线都正确工作。

例子是理解竞争本质的最简单的方法。让我们来看看图 12-42 中的简单程序。主线程创建了四个对等线程，并传递一个指向一个唯一的整数 ID 的指针到每个线程。每个对等线程复制它的参数中传递的 ID 到一个局部变量中(第 22 行)，然后输出一个包含这个 ID 的信息。它看上去足够简单，但是当我们在系统上运行这个程序时，我们得到以下不正确的结果：

```
linux> ./race
Hello from thread 1
Hello from thread 3
Hello from thread 2
Hello from thread 3
```

code/conc/race.c

```
1   /* WARNING: This code is buggy! */
2   #include "csapp.h"
3   #define N 4
4
5   void *thread(void *vargp);
6
7   int main()
8   {
9       pthread_t tid[N];
10      int i;
11
12      for (i = 0; i < N; i++)
13          Pthread_create(&tid[i], NULL, thread, &i);
14      for (i = 0; i < N; i++)
15          Pthread_join(tid[i], NULL);
16      exit(0);
17  }
```

图 12-42 一个具有竞争的程序