

从所有教程的词条中查询...

首页 > 慕课教程 > 物联网/嵌入式工程师 > 3.6 线程池的销毁

全部开发者教程

12.8 多路复用 io-epoll(三)-原理分析

第13周 Linux 项目实战-实现并发服务器模型与企业笔试

1.1 项目框架介绍

2.1 环形队列设计(一)-基本框架与共享内存模块设计

2.2 环形队列设计(二)-环形队列初始化与销毁

2.3 环形队列设计(三)-环形队列数据读写实现

3.1 线程池原理与框架设计

3.2 线程池的数据结构设计

3.3 线程池初始化

3.4 线程池添加任务

3.5 线程执行函数

3.6 线程池的销毁

3.7 线程池的测试



大白老师 · 更新于 2022-12-09

上一节 3.5 线程执行函数 3.7 线程池的测试 下一节

## 一、线程池的销毁函数实现

- 销毁函数主要功能是将所有的线程退出,申请的资源进行释放,具体实现如下:

<> 代码块

```
1  int thread_pool_destroy(tpool_t *pool)
2  {
3      if (!pool)
4          return -1;
5
6      pool->tp_shutdown = true;
7
8      //唤醒阻塞的工作线程,在设置销毁为 true 时,线程会退出
9      for (int i = 0; i < pool->tp_number_of_threads; i++){
10         pthread_cond_signal(&pool->tp_cond_empty);
11     }
12
13     // 释放任务队列空间
14     if (pool->tp_task)
15         free(pool->tp_task);
16
17     // 释放保存线程 id 空间
18     if (pool->tp_work_thread_ids)
19         free(pool->tp_work_thread_ids);
20
21     // 释放互斥锁与条件变量
22     pthread_mutex_destroy(&pool->tp_mutex_pool);
23     pthread_cond_destroy(&pool->tp_cond_empty);
24     pthread_cond_destroy(&pool->tp_cond_full);
25
26     // 释放线程池的数据结构空间
27     free(pool);
28     pool = NULL;
29     return 0;
30 }
31 }
```

3.5 线程执行函数 上一节 下一节 3.7 线程池的测试

我要提出意见反馈

