

3.4 线程池添加任务_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 3.4 线程池添加任务涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

- 具体的逻辑如下:

- step 1 : 获取线程池互斥锁
 - `pthread_mutex_lock(&pool->tp_mutex_pool);`
- step 2 : 判断任务队列是否已满，如果满了则阻塞在 `tp_cond_full` 条件变量上
 - ```
while(pool->tp_qcapacity == pool->tp_qsize && !pool->tp_shutdown){
 pthread_cond_wait(&pool->tp_cond_full,&pool->tp_mutex_pool);
}
```
- step 3 : 判断销毁标志是否为 `true` , 释放线程池互斥锁, 如果是则直接返回
  - ```
while(pool->tp_qcapacity == pool->tp_qsize && !pool->tp_shutdown){
    pthread_cond_wait(&pool->tp_cond_full,&pool->tp_mutex_pool);
}
```
- step 4 : 将任务函数与参数添加到任务队列中
 - ```
pool->tp_task[pool->tp_qrear].function = task;
pool->tp_task[pool->tp_qrear].arg = arg;
```
- step 5 : 更新队尾位置
  - `pool->tp_qrear = (pool->tp_qrear + 1) % pool->tp_qcapacity;`
- step 6 : 更新队列的长度
  - `pool->tp_qsize++;`
- step 7 : 唤醒一个工作线程执行任务
  - `pthread_cond_signal(&pool->tp_cond_empty);/`
- step 8 : 释放线程池互斥锁
  - `pthread_mutex_unlock(&pool->tp_mutex_pool);`

---

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

