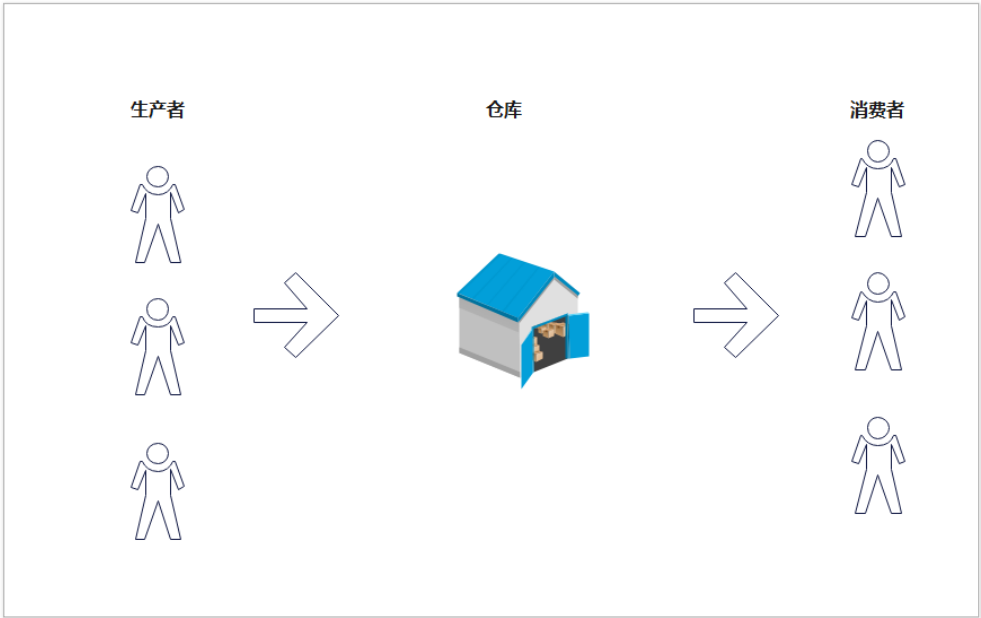


10.2 线程同步_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 10.2 线程同步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。

- 线程同步：是指在互斥的基础上（大多数情况），通过其它机制实现访问者对 资源的有序访问。
- 条件变量： 线程库提供的专门针对线程同步的机制
- 线程同步比较典型的应用场合就是 生产者与消费者
- 在这个模型中, 分为 生产者线程 与 消费者线程, 通过这个线程来模拟多个线程同步的过程。
- 在这个模型中, 需要以下组件:
 - 仓库：用于存储产品, 一般作为共享资源
 - 生产者线程：用于生产产品
 - 消费者线程：用于消费产品
 -
- 原理
 - 当仓库没有产品时, 则消费者线程需要等待, 直到有产品时才能消费
 - 当仓库已经装满产品时, 则生产者线程需要等待, 直到消费者线程消费产品之后



示例

- 基于互斥锁实现生产者与消费者模型
- - 主线程为消费者
 - n 个子线程作为生产者

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <pthread.h>
#include <unistd.h>
```

```
static int number = 0;

pthread_mutex_t mtx = PTHREAD_MUTEX_INITIALIZER;

void *thread_handler(void *arg)
{
    int cnt = atoi((char *)arg);
    int i,tmp;

    for(i = 0;i < cnt;i++){

        pthread_mutex_lock(&mtx);

        printf("线程 [%ld] 生产一个产品,产品数量为:%d\n",pthread_self(),++number);

        pthread_mutex_unlock(&mtx);

    }

    pthread_exit((void *)0);
}

int main(int argc,char *argv[])
{
    pthread_t tid;
    int i;
    int err;
    int total_of_produce = 0;
    int total_of_consume = 0;
    bool done = false;

    for (i = 1;i < argc;i++){
        total_of_produce += atoi(argv[i]);
        err = pthread_create(&tid,NULL,thread_handler,(void *)argv[i]);
        if (err != 0)
        {
            perror("pthread_create()");
            exit(EXIT_FAILURE);
        }
    }

    for (;;) {
        pthread_mutex_lock(&mtx);

        while(number > 0){
            total_of_consume++;
            printf("消费一个产品,产品数量为:%d\n",--number);
            done = total_of_consume >= total_of_produce;
            sleep(1);
        }

        pthread_mutex_unlock(&mtx);

        if (done)
            break;
    }

    return 0;
}
~
```

练习

实现生产者与消费者模型，并解决上述代码中线程资源的释放

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

