

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 > 8.2 线程的退出、等待与分离

全部开发者教程

4.2 进程间通讯-有名管道

4.3 进程间通讯-信号 (一)

4.4 进程间通讯-信号(二)

4.5 进程间通讯-信号 (三)

4.6 进程间通讯-消息队列(一)

4.7 进程间通讯-消息队列(二)

4.8 进程间通讯-共享内存(一)

4.9 进程间通讯-共享内存(二)

5.1 进程间通讯-信号量(一)

5.2 进程间通讯-信号量(二)

5.3 进程间通讯-信号量同步

6.1 线程的概念

7.1 线程相关命令

8.1 线程创建

8.2 线程的退出、等待与分离

8.3 创建多个线程



大白老师 · 更新于 2022-11-11

◀ 上一节 8.1 线程创建 8.3 创建多个线程 下一节 ▶

一、线程退出

- 线程退出使用 **pthread_exit** 函数

函数头文件 #include <pthread.h>

函数原型 void pthread_exit(void *retval);

函数功能 让线程退出,并返回值

函数参数

retval : 线程返回值,通过指针传递

函数返回值 成功 : 返回 0

失败 : 返回 -1

- 1.当主线程调用 pthread_exit 函数时，进程不会结束，也不会导致其他子线程退出
- 2. 任何线程调用 exit 函数会让进程结束

二、线程等待

- 主线程需要等待子线程退出,并释放子线程资源

- 线程等待调用 **pthread_join** 函数

函数头文件 #include <pthread.h>

函数原型 int pthread_join(pthread_t thread, void **retval);

函数功能 等待子进程退出,并释放子线程资源

函数参数

- thread : 线程 ID
- retval : 获取线程退出值的指针

函数返回值

- 成功 : 返回 0
- 失败 : 返回 错误码

示例

- 创建一个线程,主线程等待子线程退出

意见反馈

收藏教程

标记书签

<> 代码块

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <pthread.h>
5
6
7
8  void *do_thread(void *arg)
9  {
10     printf("Thread start.\n");
11     pthread_exit(NULL);
12 }
13
14 int main(void)
15 {
16     int err;
17     pthread_t tid = 0;
18
19     err = pthread_create(&tid,NULL,do_thread,NULL);
20     if (err != 0){
21         fprintf(stderr,"[ERROR] pthread_create : < %s >\n",strerror(err));
22         exit(EXIT_FAILURE);
23     }
24
25     printf(" tid = %ld\n",tid);
26
27     pthread_join(tid,NULL);
28
29     return 0;
30 }
```

练习

创建两个子线程，打印两个子线程的id，并等待两个子线程退出

三、线程分离

- 线程分为可结合的与可分离的

- 可结合

- 可结合的线程能够被其他线程收回其资源和杀死；在被其他线程回收之前，它的存储器资源（如栈）是不释放的。
- 线程创建的默认状态为 可结合的，可以由其他线程调用 **pthread_join** 函数等待子线程退出并释放相关资源

- 可分离

- 不能被其他线程回收或者杀死的，该线程的资源在它终止时由系统来释放。
-

- 线程分离调用 **pthread_detach** 函数

函数头文件 `#include <pthread.h>`

函数原型 `int pthread_detach(pthread_t thread);`

函数功能 设置在线程退出后，由操作系统自动释放该线程的资源

[意见反馈](#)[收藏教程](#)[标记书签](#)

函数返回值

- 成功：返回 0
- 失败：返回 -1

注意

线程分离函数不会阻塞线程的执行

示例 创建一个线程，并设置线程为可分离状态

<> 代码块

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <pthread.h>
5
6
7
8  void *do_thread(void *arg)
9  {
10     printf("Thread start.\n");
11     pthread_exit(NULL);
12 }
13
14 int main(void)
15 {
16     int err;
17     pthread_t tid = 0;
18
19     err = pthread_create(&tid, NULL, do_thread, NULL);
20     if (err != 0){
21         fprintf(stderr, "[ERROR] pthread_create : < %s >\n", strerror(err));
22         exit(EXIT_FAILURE);
23     }
24
25     printf(" tid = %ld\n", tid);
26
27     pthread_detach(tid);
28
29
30     while(1){
31         /*nothing*/
32     }
33
34     return 0;
35 }
```

练习

创建二个线程，并使两个线程分离

8.1 线程创建 ◀ 上一节

下一节 ▶ 8.3 创建多个线程

✎ 我要提出意见反馈

