**进程与线程**

对于进程来说，进程由操作系统来运行程序，可以有多个线程。两个进程可以在同一个程序中同时执行代码。

进程是正在执行计算机程序的实例。每个进程都有自己的内存空间，它用来存储正在运行的指令，以及需要存储和访问执行的任何数据。

线程是进程的组件，可以并行运行。一个进程中可以有多个线程，它们共享相同的内存空间，即父进程的内存空间。

什么时候使用进程或线程？

在计算机中中央处理器或处理器管理计算机的实际计算工作中。CPU有一个或多个内核，允许CPU同时执行代码。

* 进程加速CPU密集型的操作，因为它们受益于多个内核。
* 线程最适合I/O任务或涉及外部系统的任务，因为线程可以有效地组合它们地工作。

**Pthreads简介**

POSIX Threads简称Pthreads，POSIX是"Portable Operating System Interface"（可移植操作系统接口） 的缩写，POSIX是IEEE Computer Society为了提高不同操作系统的兼容性和应用程序的可移植性而制定的一套标准。

pthread\_t类型是线程标识符。

   typedef rt\_thread\_t pthread\_t;

**创建线程**

**函数原型：**

int pthread\_create (pthread\_t \*tid,  
                  const pthread\_attr\_t \*attr,  
                  void \*(\*start) (void \*),  
                   void \*arg);

1. tid 指向线程句柄(线程标识符)的指针，不能为NULL
2. attr 指向线程属性的指针，如果使用NULL，则使用默认的线程属性
3. start 线程入口函数地址
4. arg 传递给线程入口函数的参数

**函数返回：**

创建成功返回0

此函数创建一个pthread线程。此函数会动态分配POSIX线程数据块和RT-Thread线程控制块，并把线程控制块的起始地址（线程ID）保存在参数tid指向的内存里，此线程标识符可用于在其他线程中操作此线程；并把attr指向的线程属性、start指向的线程入口函数及入口函数参数arg保存在线程数据块和线程控制块里。如果线程创建成功，线程立刻进入就绪态，参与系统的调度，如果线程创建失败，则会释放之前线程占有的资源。

**线程脱离**

**函数原型：**

   int pthread\_detach (pthread\_t thread);

**函数返回：**

只返回0，总是成功。

调用此函数，如果pthread线程没有结束，则将thread线程属性的分离状态设置为detached；当thread线程已经结束时，系统将回收pthread线程占用的资源。

使用方法：子线程调用pthread\_detach(pthread\_self())（pthread\_self()返回当前调用线程的线程句柄），或者其他线程调用pthread\_detach(thread\_id)。关于线程属性的分离状态会在后面详细介绍。

**等待线程结束**

**函数原型：**

   int pthread\_join (pthread\_t thread, void \*\*value\_ptr);

**函数返回：**

执行成功返回0，线程join自己返回EDEADLK；join一个分离状态为detached的线程返回EINVAL；找不到pthread线程返回ESRCH。

此函数会使调用该函数的线程以阻塞的方式等待线程分离属性为joinable的thread线程运行结束，并获得thread线程的返回值，返回值的地址保存在value\_ptr里，并释放thread线程占用的资源。

#include <stdio.h>  
#include <stdlib.h>  
#include <pthread.h>  
​  
void \*func(void \*arg) {  
   int i = (int)arg;  
   printf("this is %d thread\n",i);  
   // pthread\_exit(0);  
}  
​  
int main()  
{  
   pthread\_t tids[5] = {0};  
   for(int i = 0; i < 5; ++i) {  
       pthread\_create(&tids[i],NULL, func, (void\*)i);  
  }  
   for(int i = 0; i < 5; ++i) {  
       pthread\_join(tids[i], NULL);  
  }  
}

**线程退出**

**函数原型：**

   void pthread\_exit(void \*value\_ptr);

**函数返回：**

此函数没有返回值。

pthread线程调用此函数会终止执行，如同进程调用exit()函数一样，并返回一个指向线程返回值的指针。线程退出由线程自身发起。

[16.POSIX接口 - 线程 - 《RT-Thread编程手册》 - 书栈网 · BookStack](https://www.bookstack.cn/read/rtthread-manual-doc/16.2.md)

[(5条消息) 线程以及pthread库的使用\_椛茶的博客-CSDN博客](https://blog.csdn.net/weixin_38102771/article/details/91351126)