## 线程的创建

#include <pthread.h>

int pthread\_create(pthread\_t \*thread, const

pthread\_attr\_t \*attr, void \*(\*routine)(void \*), void \*arg);

成功返回0，失败时返回错误码

thread 线程对象

attr 线程属性，NULL代表默认属性

routine 线程执行的函数

arg 传递给routine的参数 ，参数是void \* ，注意传递参数格式，

## 编译错误分析：

1.

createP\_t.c:14:36: warning: passing argument 3 of ‘pthread\_create’ from incompatible pointer type [-Wincompatible-pointer-types]

ret = pthread\_create(&tid,NULL,testThread,NULL);

^

In file included from createP\_t.c:1:0:

/usr/include/pthread.h:233:12: note: expected ‘void \* (\*)(void \*)’ but argument is of type ‘int \* (\*)(char \*)’

意义：表示pthread\_create参数3的定义和实际代码不符合，期望的是void \* (\*)(void \*) ，实际的代码是int \* (\*)(char \*)

解决方法：改为pthread\_create(&tid,NULL,(void\*)testThread,NULL);

2.

createP\_t.c:(.text+0x4b)：对‘pthread\_create’未定义的引用

collect2: error: ld returned 1 exit status --------这个链接错误，

表示pthread\_create这个函数没有实现

解决方法：编译时候加 -lpthread

注意事项：1. 主进程的退出，它创建的线程也会退出。

线程创建需要时间，如果主进程马上退出，那线程不能得到执行

获取线程的id

通过pthread\_create函数的第一个参数；通过在线程里面调用pthread\_self函数

## 线程间参数传递：（重点难点）

编译错误：

createP\_t.c:8:34: warning: dereferencing ‘void \*’ pointer

printf("input arg=%d\n",(int)\*arg);

^

createP\_t.c:8:5: error: invalid use of void expression

printf("input arg=%d\n",(int)\*arg);

错误原因是void \*类型指针不能直接用\*取值（\*arg），因为编译不知道数据类型。

解决方法：转换为指定的指针类型后再用\*取值 比如：\*(int \*)arg

1. 通过地址传递参数，注意类型的转换
2. 值传递，这时候编译器会告警，需要程序员自己保证数据长度正确

## 运行错误：

\*\*\* stack smashing detected \*\*\*: ./mthread\_t terminated

已放弃 (核心已转储)

原因：栈被破坏了（数组越界）