P2

2018年3月24日 9:56

Pthread 库API参考：
中文 https://blog.csdn.net/future_fighter/article/details/3865071#phtread_ref
英文 https://computing.llnl.gov/tutorials/pthreads/#Abstract

Socket：
中文 https://www.cnblogs.com/liushui-sky/p/5609535.html

https://socket.io/docs/server-api/#
中文常用函数手册：
https://www.cnblogs.com/liushui-sky/p/5609535.html
pthread： https://blog.csdn.net/future_fighter/article/details/3865071#phtread_ref
简单C/S交互教程：
https://www.cnblogs.com/liushao/p/6375377.html
多线程：
https://www.jianshu.com/p/184c1847a2f9
https://blog.csdn.net/dlutbrucezhang/article/details/8872581
https://blog.csdn.net/liujiabin076/article/details/53456962
https://blog.csdn.net/future_fighter/article/details/3865071#pthreads_overview
https://www.cnblogs.com/nerohwang/p/3602233.html

select()函数以及FD_ZERO、FD_SET、FD_CLR、FD_ISSET：
fd 是(file descriptor)，这种一般是BSD Socket的用法，用在Unix/Linux系统上。在Unix/Linux系统下，一个socket句柄，可以看做是一个文件，在socket上收发数据，相当于对一个文件进行读写，所以一个socket句柄，通常也用表示文件句柄的fd来表示。

fd_set
select()机制中提供一fd_set的数据结构，实际上是一long类型的数组，每一个数组元素都能与一打开的文件句柄（不管是socket句柄，还是其他文件或命名管道或设备句柄）建立联系，建立联系的工作由程序员完成，当调用select()时，由内核根据IO状态修改fd_set的内容，由此来通知执行了select()的进程哪一socket或文件发生了可读或可写事件。

空指针和void*类型指针：
1. 空指针

1、空指针实质上是有指向的指针，但它指向的地址是很小的地址，约定俗成为地址0，我来解释一下为什么。

```objc
#include <stdio.h>
```

```objc
void main(){
    intint * str = NULL;
    gets(str);
    printf("%s",str);
}
```

这个程序在语法上是没有问题的，但是我们编译、链接都没有报错，但是我们打开的时候却出错了。

原因是空指针指向的地址是不保存数据，同时不允许程序访问的。

2. void *类型指针

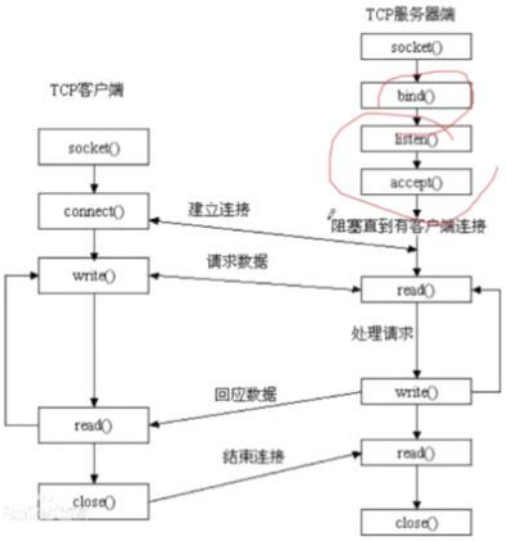
2、void * 类型指针，这个类型指针指向了实实在在的存放数据的地址，但是该地址存放的数据的数据类型我们暂时不知道。

举个例子，我们的动态内存分配就是这样，一开始只是分配地址，但没有知道这块地址用了存放什么，接着强制类型转换，使得它用来存放我们想要存放的内容。

```
char*str=(char*)malloc(sizeof(char)*13);
```

上面这条代码，malloc()函数分配的地址一开始是void * 类型的，因为我们用来存放char类型数据，所以强制转换为 char * 。

Socket通信流程图



来自 <https://www.jianshu.com/p/9cb60d675dc8>