多路I/O转接服务器

多路IO转接服务器也叫做多任务IO服务器。该类服务器实现的主旨思想是，不再由应用程序自己监视客户端连接，取而代之由内核替应用程序监视文件。

select

1. select能监听的文件描述符个数受限于FD\_SETSIZE,一般为1024，单纯改变进程打开的文件描述符个数并不能改变select监听文件个数
2. 解决1024以下客户端时使用select是很合适的，但如果链接客户端过多，select采用的是轮询模型，会大大降低服务器响应效率，不应在select上投入更多精力

#include <sys/select.h>

/\* According to earlier standards \*/

#include <sys/time.h>

#include <sys/types.h>

#include <unistd.h>

int select(int nfds, fd\_set \*readfds, fd\_set \*writefds,

fd\_set \*exceptfds, struct timeval \*timeout);

nfds: 监控的文件描述符集里最大文件描述符加1，因为此参数会告诉内核检测前多少个文件描述符的状态

readfds： 监控有读数据到达文件描述符集合，传入传出参数

writefds： 监控写数据到达文件描述符集合，传入传出参数

exceptfds： 监控异常发生达文件描述符集合,如带外数据到达异常，传入传出参数

timeout： 定时阻塞监控时间，3种情况

1.NULL，永远等下去

2.设置timeval，等待固定时间

3.设置timeval里时间均为0，检查描述字后立即返回，轮询

struct timeval {

long tv\_sec; /\* seconds \*/

long tv\_usec; /\* microseconds \*/

};

void FD\_CLR(int fd, fd\_set \*set); //把文件描述符集合里fd清0

int FD\_ISSET(int fd, fd\_set \*set); //测试文件描述符集合里fd是否置1

void FD\_SET(int fd, fd\_set \*set); //把文件描述符集合里fd位置1

void FD\_ZERO(fd\_set \*set); //把文件描述符集合里所有位清0