使用fork()实现多线程

#include <stdio.h>

#include <assert.h>

#include <unistd.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <sys/wait.h>

void sig\_child(int sig)

{

pid\_t pid;

int stat;

while( (pid == waitpid(-1,&stat,WNOHANG)) >0 )

{

printf("child %d exit\n",pid);

}

return;

}

int main(int argc,int \*\* argv)

{

int sockfd = socket(AF\_INET,SOCK\_STREAM,0);

assert( sockfd != -1);

struct sockaddr\_in saddr,caddr;

memset(&saddr,0,sizeof(saddr));

saddr.sin\_family = AF\_INET;

saddr.sin\_port = htons(6000);

saddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

int res = bind(sockfd,(struct sockaddr\*)&saddr,sizeof(saddr));

assert( res != -1);

listen(sockfd,5);

while( 1 )

{

int len = sizeof(caddr);

int c = accept(sockfd,(struct sockaddr\*)&caddr,&len);

if( c<0 )

{

continue;

}

pid\_t pid = fork();

if( pid == 0)

{

while(1)

{

char buff[128]={0};

int n = recv(c,buff,127,0);

if( n <= 0)

{

break;

}

printf("buff(%d)=%s\n",c,buff);

send(c,"ok",2,0);

}

signal(SIGCHLD,sig\_child);

printf("one client over\n");

close(c);

}

}

close(sockfd);

}