

Nov 02, 20 1:48

echo\_server\_fork.c

Page 1/2

```

#include <sys/types.h>
#include <sys/stat.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <signal.h>

#define MAXLINE 1024
#define PORTNUM 3600

int main(int argc, char **argv)
{
    int listen_fd, client_fd;
    pid_t pid;
    socklen_t addrlen;
    int readn;
    int c[3];
    char buf[MAXLINE];
    char str[MAXLINE]="";
    struct sockaddr_in client_addr, server_addr;

    if( (listen_fd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    {
        return 1;
    }
    memset((void *)&server_addr, 0x00, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    server_addr.sin_port = htons(PORTNUM);

    if(bind(listen_fd, (struct sockaddr *)&server_addr, sizeof(server_addr))
    ==-1)
    {
        perror("bind error");
        return 1;
    }
    if(listen(listen_fd, 5) == -1)
    {
        perror("listen error");
        return 1;
    }

    signal(SIGCHLD, SIG_IGN);
    for(int i=0; i<3; i++)
    {
        addrlen = sizeof(client_addr);
        client_fd = accept(listen_fd,
            (struct sockaddr *)&client_addr, &addrlen);
        c[i] = client_fd;
        if(client_fd == -1)
        {
            printf("accept error\n");
            break;
        }
        printf("Client Connected: %s\n", inet_ntoa(client_addr.sin_addr));

        memset(buf, 0x00, MAXLINE);
        read(client_fd, buf, MAXLINE);
        if((readn = read(client_fd, buf, MAXLINE)) > 0) {

```

Nov 02, 20 1:48

echo\_server\_fork.c

Page 2/2

```

        printf("Read Data %s(%d): %s",
            inet_ntoa(client_addr.sin_addr),
            client_addr.sin_port, buf);

        }
        strcat(str,buf);
        pid = fork();
        if(pid == 0) {return 0;}
    }

    for(int i=0; i<3; i++) {
        if(write(c[i], str, MAXLINE) <=0) {
            perror("Write error: ");
            close(client_fd);
        }
        close(c[i]);
    }

    close(listen_fd);

    return 0;
}

```