

Nov 22, 20 23:52

server.c

Page 1/3

```

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>

#include <sys/stat.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/time.h>

#include <sys/ipc.h>
#include <sys/shm.h>
#include <sys/sem.h>

#define PORT 3500

struct data
{
    char str[1024];
    int num;
};

union semun
{
    int val;
};

int main(int argc, char **argv)
{
    struct sockaddr_in client_addr, sock_addr;
    int listen_sockfd, client_sockfd;
    int addr_len;
    struct data rdata;
    int num;
    char str[1024];
    pid_t pid;
    int shmidx;
    int semid;
    int *cal_num;
    void *shared_memory = NULL;
    union semun sem_union;
    struct sembuf semopen = {0, -1, SEM_UNDO};
    struct sembuf semclose = {0, 1, SEM_UNDO};

    if( (shmidx = shmget((key_t)1234, sizeof(int), 0666|IPC_CREAT)) == -1)
    {
        return 1;
    }

    if( (semid = semget((key_t)3477, 1, IPC_CREAT|0666)) == -1)
    {
        return 1;
    }

    if( (shared_memory = shmat(shmidx, NULL, 0)) == (void*)-1)
    {
        return 1;
    }

    cal_num = (int*) shared_memory;
    sem_union.val = 1;
    if (-1 == semctl(semid, 0, SETVAL, sem_union))
    {
        return 1;
    }

    if( (listen_sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1)

```

Nov 22, 20 23:52

server.c

Page 2/3

```

    perror("Error");
    return 1;
}

memset((void *)&sock_addr, 0x00, sizeof(sock_addr));
sock_addr.sin_family = AF_INET;
sock_addr.sin_addr.s_addr = htonl(INADDR_ANY);
sock_addr.sin_port = htons(PORT);

if( bind(listen_sockfd, (struct sockaddr *)&sock_addr, sizeof(sock_addr)) == -1)
{
    perror("Error");
    return 1;
}

if(listen(listen_sockfd, 5) == -1)
{
    perror("Error");
    return 1;
}

for(;;)
{
    addr_len = sizeof(client_addr);
    client_sockfd = accept(listen_sockfd,
        (struct sockaddr *)&client_addr, &addr_len);
    if(client_sockfd == -1)
    {
        perror("accept error:");
        return 1;
    }

    printf("New Client Connect: %s\n", inet_ntoa(client_addr.sin_addr));

    pid = fork();
    if(pid == 0)
    {
        close(listen_sockfd);

        read(client_sockfd, (char *)&rdata, sizeof(rdata));
        strcpy(str, rdata.str);
        int len = strlen(str);
        printf("recieved data: %s and %d\n", str, ntohl(rdata.num));

        while(1)
        {
            if(semop(semid, &semopen, 1) == -1)
            {
                return 1;
            }
            num = ntohl(rdata.num);
            num++;
            rdata.num = htonl(num);
            strcpy(str, rdata.str);
            for(int i=0; i<strlen(str); i++){
                rdata.str[i] = str[i+1];
            }
            rdata.str[len-1] = str[0];
            rdata.str[len] = 0;

            sleep(1);
            write(client_sockfd, (char *)&rdata, sizeof(rdata));

            semop(semid, &semclose, 1);
        }
    }

    else if (pid>0)
        close(client_sockfd);
}

```

Nov 22, 20 23:52

**server.c**

Page 3/3

```
    }  
    close(listen_sockfd);  
    return 0;  
}
```