





vidyakv / Network Programming


Commits

 **vidyakv** committed **7f19f91**
2011-11-23

Multiclient chat server

 01a91eb


 default

 [View raw commit](#)

Comments (0)

Files changed (2)

+79	-0	N_client.c
+126	-0	server.c

 N_client.c

Side-by-side diffView file⋮

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include <sys/types.h>
5  #include <sys/socket.h>
6  #include <netinet/in.h>
7  #include <arpa/inet.h>
8  #include <unistd.h>
9  #include <errno.h>
10 +
11 #define BUFSIZE 1024
12 +
13 +void send_recv(int i, int sockfd)
14 +{
15 +    char send_buf[BUFSIZE];
16 +    char recv_buf[BUFSIZE];
17 +    int nbyte_rcvd;
18 +
19 +    if (i == 0){
20 +        fgets(send_buf, BUFSIZE, stdin);
21 +        if (strcmp(send_buf, "quit\n") == 0) {
22 +            exit(0);
23 +        }else
24 +            send(sockfd, send_buf, strlen(send_buf), 0);
25 +    }else {
26 +        nbyte_rcvd = recv(sockfd, recv_buf, BUFSIZE, 0);
27 +        recv_buf[nbyte_rcvd] = '\0';
28 +        printf("%s\n", recv_buf);
29 +        fflush(stdout);
30 +    }
31 +}
32 +
33 +
34 +void connect_request(int *sockfd, struct sockaddr_in *server_addr)
35 +{
36 +    if ((*sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
37 +        perror("Socket");
38 +        exit(1);
39 +    }
40 +    server_addr->sin_family = AF_INET;
41 +    server_addr->sin_port = htons(4950);
42 +    server_addr->sin_addr.s_addr = inet_addr("127.0.0.1");
43 +    memset(server_addr->sin_zero, '\0', sizeof server_addr->sin_zero);
44 +
45 +    if(connect(*sockfd, (struct sockaddr *)server_addr, sizeof(struct sockaddr)) ==
46 +        perror("connect");
47 +        exit(1);
48 +    }
49 +}
50 +
51 +int main()
```

```

52 +{
53 +    int sockfd, fdmax, i;
54 +    struct sockaddr_in server_addr;
55 +    fd_set master;
56 +    fd_set read_fds;
57 +
58 +    connect_request(&sockfd, &server_addr);
59 +    FD_ZERO(&master);
60 +    FD_ZERO(&read_fds);
61 +    FD_SET(0, &master);
62 +    FD_SET(sockfd, &master);
63 +    fdmax = sockfd;
64 +
65 +    while(1){
66 +        read_fds = master;
67 +        if(select(fdmax+1, &read_fds, NULL, NULL, NULL) == -1){
68 +            perror("select");
69 +            exit(4);
70 +        }
71 +
72 +        for(i=0; i <= fdmax; i++ )
73 +            if(FD_ISSET(i, &read_fds))
74 +                send_rcv(i, sockfd);
75 +    }
76 +    printf("client-quit\n");
77 +    close(sockfd);
78 +    return 0;
79 +}

```

server.c

Side-by-side diff View file ...

```

1  +#include <stdio.h>
2  +#include <stdlib.h>
3  +#include <string.h>
4  +#include <unistd.h>
5  +#include <sys/types.h>
6  +#include <sys/socket.h>
7  +#include <netinet/in.h>
8  +#include <arpa/inet.h>
9  +#include <netdb.h>
10 +
11 +#define PORT 4950
12 +#define BUFSIZE 1024
13 +
14 +void send_to_all(int j, int i, int sockfd, int nbytes_rcvd, char *rcv_buf, fd_set *ma
15 +{
16 +    if (FD_ISSET(j, master)){
17 +        if (j != sockfd && j != i) {
18 +            if (send(j, rcv_buf, nbytes_rcvd, 0) == -1) {
19 +                perror("send");
20 +            }
21 +        }
22 +    }
23 +}
24 +
25 +void send_rcv(int i, fd_set *master, int sockfd, int fdmax)
26 +{
27 +    int nbytes_rcvd, j;
28 +    char rcv_buf[BUFSIZE], buf[BUFSIZE];
29 +
30 +    if ((nbytes_rcvd = recv(i, rcv_buf, BUFSIZE, 0)) <= 0) {
31 +        if (nbytes_rcvd == 0) {
32 +            printf("socket %d hung up\n", i);
33 +        }else {
34 +            perror("rcv");
35 +        }
36 +        close(i);
37 +        FD_CLR(i, master);
38 +    }else {
39 +        // printf("%s\n", rcv_buf);
40 +        for(j = 0; j <= fdmax; j++){
41 +            send_to_all(j, i, sockfd, nbytes_rcvd, rcv_buf, master );
42 +        }
43 +    }
44 +}
45 +
46 +void connection_accept(fd_set *master, int *fdmax, int sockfd, struct sockaddr_in *cli
47 +{
48 +    socklen_t addrlen;
49 +    int newsockfd;

```



Network Programming

ACTIONS

- Clone
- Compare
- Fork

NAVIGATION


- Overview
- Source
- Commits
- Branches
- Pull requests
- Downloads

<<

```


50 +
51 +     addrln = sizeof(struct sockaddr_in);
52 +     if((newsockfd = accept(sockfd, (struct sockaddr *)client_addr, &addrln)) == -1)
53 +         perror("accept");
54 +         exit(1);
55 +     }else {
56 +         FD_SET(newsockfd, master);
57 +         if(newsockfd > *fdmax){
58 +             *fdmax = newsockfd;
59 +         }
60 +         printf("new connection from %s on port %d \n",inet_ntoa(client_addr->sin_addr),
61 +             client_addr->sin_port);
62 +     }
63 + }
64 +void connect_request(int *sockfd, struct sockaddr_in *my_addr)
65 +{
66 +     int yes = 1;
67 +
68 +     if ((*sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
69 +         perror("Socket");
70 +         exit(1);
71 +     }
72 +
73 +     my_addr->sin_family = AF_INET;
74 +     my_addr->sin_port = htons(4950);
75 +     my_addr->sin_addr.s_addr = INADDR_ANY;
76 +     memset(my_addr->sin_zero, '\0', sizeof my_addr->sin_zero);
77 +
78 +     if (setsockopt(*sockfd, SOL_SOCKET, SO_REUSEADDR, &yes, sizeof(int)) == -1) {
79 +         perror("setsockopt");
80 +         exit(1);
81 +     }
82 +
83 +     if (bind(*sockfd, (struct sockaddr *)my_addr, sizeof(struct sockaddr)) == -1) {
84 +         perror("Unable to bind");
85 +         exit(1);
86 +     }
87 +     if (listen(*sockfd, 10) == -1) {
88 +         perror("listen");
89 +         exit(1);
90 +     }
91 +     printf("\nTCPServer Waiting for client on port 4950\n");
92 +     fflush(stdout);
93 + }
94 +int main()
95 +{
96 +     fd_set master;
97 +     fd_set read_fds;
98 +     int fdmax, i;
99 +     int sockfd= 0;
100 +     struct sockaddr_in my_addr, client_addr;
101 +
102 +     FD_ZERO(&master);
103 +     FD_ZERO(&read_fds);
104 +     connect_request(&sockfd, &my_addr);
105 +     FD_SET(sockfd, &master);
106 +
107 +     fdmax = sockfd;
108 +     while(1){
109 +         read_fds = master;
110 +         if(select(fdmax+1, &read_fds, NULL, NULL, NULL) == -1){
111 +             perror("select");
112 +             exit(4);
113 +         }
114 +
115 +         for (i = 0; i <= fdmax; i++){
116 +             if (FD_ISSET(i, &read_fds)){
117 +                 if (i == sockfd)
118 +                     connection_accept(&master, &fdmax, sockfd, &client_addr);
119 +                 else
120 +                     send_recv(i, &master, sockfd, fdmax);
121 +             }
122 +         }
123 +     }
124 +     return 0;
125 + }
126 +

```




Network Programming


ACTIONS



Clone




Compare




Fork


NAVIGATION




Overview




Source




Commits



Branches



Pull requests



Downloads

<<