

A Simple Stream-Oriented Server Uses TCP Port Number 3456

```
#include <stdio>;
#include <stdlib>;
#include <errno>;
#include <string>;
#include <sys/types>;
#include <netinet/in>;
#include <sys/socket>;
#include <sys/wait>;

#define MYPORT 3456    /* the port users will be connecting to */
#define BACKLOG 10    /* number of pending connections */

main()
{
    int sockfd, new_fd;          /* listen on sockfd,
                                new connection on new_fd */
    struct sockaddr_in my_addr;  /* my address information */
    struct sockaddr_in their_addr; /* client's address info */
    int sin_size;

    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1)
    {
        perror("socket");
        exit(1);
    }

    my_addr.sin_family = AF_INET;
    my_addr.sin_port = htons(MYPORT);
    my_addr.sin_addr.s_addr = INADDR_ANY; /* auto-fill with my IP */
    bzero(&(my_addr.sin_zero), 8);        /* zero the rest */

    if (bind(sockfd, (struct sockaddr *)&my_addr, sizeof(struct sockaddr)) == -1)
    {
        perror("bind");
        exit(1);
    }

    if (listen(sockfd, BACKLOG) == -1)
    {
        perror("listen");
        exit(1);
    }

    while(1)                    /* main accept() loop */
    {
        sin_size = sizeof(struct sockaddr_in);
        if ((new_fd = accept(sockfd, (struct sockaddr *)&their_addr, &sin_size)) ==
-1)
        {
            perror("accept");
            continue;
        }
        printf("server: got connection from %s\n",
                inet_ntoa(their_addr.sin_addr));

        if (!fork())             /* this is the child process */
        {
            if (send(new_fd, "Hello, world!\n", 14, 0) == -1)
                perror("send");
            close(new_fd);
            exit(0);
        }
        close(new_fd); /* parent doesn't need this */

        while(waitpid(-1, NULL, WNOHANG) > 0); /* clean up child processes */
    }
}
```

A Simple Stream-Oriented Client Uses TCP Port Number 3456

```
#include <stdio>;
#include <stdlib>;
#include <errno>;
#include <string>;
#include <netdb>;
#include <sys/types>;
#include <netinet/in>;
#include <sys/socket>;

#define PORT 3456          /* the port client will be connecting to */

#define MAXDATASIZE 100    /* max number of bytes we can get at once */

int main(int argc, char *argv[])
{
    int sockfd, numbytes;
    char buf[MAXDATASIZE];
    struct hostent *he;
    struct sockaddr_in their_addr; /* client's address information */

    if (argc != 2)
    {
        fprintf(stderr, "usage: client hostname\n");
        exit(1);
    }

    if ((he=gethostbyname(argv[1])) == NULL) /* get the host info */
    {
        perror("gethostbyname");
        exit(1);
    }

    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1)
    {
        perror("socket");
        exit(1);
    }

    their_addr.sin_family = AF_INET;
    their_addr.sin_port = htons(PORT);
    their_addr.sin_addr = *((struct in_addr *)he->h_addr);
    bzero(&(their_addr.sin_zero), 8);

    if (connect(sockfd, (struct sockaddr *)&their_addr, sizeof(struct sockaddr)) ==
-1)
    {
        perror("connect");
        exit(1);
    }

    if ((numbytes=recv(sockfd, buf, MAXDATASIZE, 0)) == -1)
    {
        perror("recv");
        exit(1);
    }

    buf[numbytes] = '\0';

    printf("Received: %s", buf);

    close(sockfd);

    return 0;
}
```

A Simple Datagram-Oriented Server Uses UDP Port Number 3456

```
#include <stdio>;
#include <stdlib>;
#include <errno>;
#include <string>;
#include <sys/types>;
#include <netinet/in>;
#include <sys/socket>;
#include <sys/wait>;

#define MYPORT 4950    /* the port users will be sending to */

#define MAXBUFLen 100

main()
{
    int sockfd;
    struct sockaddr_in my_addr;    /* my address information */
    struct sockaddr_in their_addr; /* client's address information */
    int addr_len, numbytes;
    char buf[MAXBUFLen];

    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1)
    {
        perror("socket");
        exit(1);
    }

    my_addr.sin_family = AF_INET;
    my_addr.sin_port = htons(MYPORT);
    my_addr.sin_addr.s_addr = INADDR_ANY;
    bzero(&(my_addr.sin_zero), 8);

    if (bind(sockfd, (struct sockaddr *)&my_addr, sizeof(struct sockaddr)) == -1)
    {
        perror("bind");
        exit(1);
    }

    addr_len = sizeof(struct sockaddr);
    if ((numbytes=recvfrom(sockfd, buf, MAXBUFLen, 0,
                          (struct sockaddr *)&their_addr, &addr_len)) == -1)
    {
        perror("recvfrom");
        exit(1);
    }

    printf("got packet from %s\n",inet_ntoa(their_addr.sin_addr));
    printf("packet is %d bytes long\n",numbytes);
    buf[numbytes] = '\0';
    printf("packet contains \"%s\"\n",buf);

    close(sockfd);
}
```

A Simple Datagram-Oriented Client

Uses UDP Port Number 3456

```
#include <stdio>;
#include <stdlib>;
#include <errno>;
#include <string>;
#include <sys/types>;
#include <netinet/in>;
#include <netdb>;
#include <sys/socket>;
#include <sys/wait>;

#define MYPORT 4950          /* the port users will be sending to */

int main(int argc, char *argv[])
{
    int sockfd;
    struct sockaddr_in their_addr; /* client's address information */
    struct hostent *he;
    int numbytes;

    if (argc != 3)
    {
        fprintf(stderr, "usage: talker hostname message\n");
        exit(1);
    }

    if ((he=gethostbyname(argv[1])) == NULL) /* get the host info */
    {
        perror("gethostbyname");
        exit(1);
    }

    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1)
    {
        perror("socket");
        exit(1);
    }

    their_addr.sin_family = AF_INET;
    their_addr.sin_port = htons(MYPORT);
    their_addr.sin_addr = *((struct in_addr *)he->h_addr);
    bzero(&(their_addr.sin_zero), 8);

    if ((numbytes=sendto(sockfd, argv[2], strlen(argv[2]), 0,
        (struct sockaddr *)&their_addr, sizeof(struct sockaddr))) == -1)
    {
        perror("sendto");
        exit(1);
    }

    printf("sent %d bytes to %s\n", numbytes, inet_ntoa(their_addr.sin_addr));

    close(sockfd);

    return 0;
}
```