

```

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdlib.h>
#include <strings.h>
#include <string.h>
#include <unistd.h>

#define SERVER_PORT 5432
#define MAX_LINE 256

int main(int argc, char * argv[])
{
    FILE *fp;
    struct hostent *hp;
    struct sockaddr_in sin;
    char *host;
    char buf[MAX_LINE];
    int s;
    int len;

    if (argc==2) {
        host = argv[1];
    }
    else {
        fprintf(stderr, "usage: simplex-talk host\n");
        exit(1);
    }

    /* translate host name into peer's IP address */
    hp = gethostbyname(host);
    if (hp == NULL) {
        fprintf(stderr, "simplex-talk: unknown host: %s\n", host);
        exit(1);
    }

    /* build address data structure */
    bzero((char *)&sin, sizeof(sin));
    sin.sin_family = AF_INET;
    bcopy(hp->h_addr, (char *)&sin.sin_addr, hp->h_length);
    sin.sin_port = htons(SERVER_PORT);

    /* active open */
    if ((s = socket(PF_INET, SOCK_STREAM, 0)) < 0) {
        perror("simplex-talk: socket");
        exit(1);
    }
    printf("Attempting to connect to %s.\n", hp->h_name);
    if (connect(s, (struct sockaddr *)&sin, sizeof(sin)) < 0)
    {
        perror("simplex-talk: connect");
        close(s);
        exit(1);
    }
    /* main loop: get and send lines of text */
    printf("Connected to %s.\n", hp->h_name);
    printf("Send a message to the server: ");

```

```
while (fgets(buf, sizeof(buf), stdin)) {  
    buf[MAX_LINE-1] = '\0';  
    len = strlen(buf) + 1;  
    send(s, buf, len, 0);  
    printf("Send a message to the server: ");  
}  
  
}
```