```c
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>

int main(void)
{
    int listenfd = 0;
    int connfd = 0;
    struct sockaddr_in serv_addr;
    char sendBuff[1025];
    int numrv;

    listenfd = socket(AF_INET, SOCK_STREAM, 0);

    printf("Socket retrieve success\n");

    memset(&serv_addr, '0', sizeof(serv_addr));
    memset(sendBuff, '0', sizeof(sendBuff));

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    serv_addr.sin_port = htons(5001);

    bind(listenfd, (struct sockaddr*)&serv_addr,sizeof(serv_addr));

    if(listen(listenfd, 10) == -1)
    {
        printf("Failed to listen\n");
        return -1;
    }


    while(1)
    {
        unsigned char offset_buffer[10] = {'\0'};
        unsigned char command_buffer[2] = {'\0'};
        int offset;
        int command;
        connfd = accept(listenfd, (struct sockaddr*)NULL ,NULL);


        printf("Waiting for client to send the command (Full File (0) Partial File (1)\n");

        while(read(connfd, command_buffer, 2) == 0);
            sscanf(command_buffer, "%d", &command);

        if(command == 0)
            offset = 0;
        else
        {
            printf("Waiting for client to send the offset\n");
            while(read(connfd, offset_buffer, 10) == 0);
            sscanf(offset_buffer, "%d", &offset);

        }
        /* Open the file that we wish to transfer */
        FILE *fp = fopen("source_file.txt","rb");
        if(fp==NULL)
        {
            printf("File open error");
            return 1;
        }

        /* Read data from file and send it */
            fseek(fp, offset, SEEK_SET);
        while(1)
        {
            /* First read file in chunks of 256 bytes */
            unsigned char buff[256]={0};
            int nread = fread(buff,1,256,fp);
            printf("Bytes read %d \n", nread);

            /* If read was success, send data. */
            if(nread > 0)
            {
                printf("Sending \n");
                write(connfd, buff, nread);
            }
            /*
            Either there was error, or we reached end of file.
            */
            if (nread < 256)
            {
                if (feof(fp))
                    printf("End of file\n");
                if (ferror(fp))
                    printf("Error reading\n");
                break;
            }
        }
        close(connfd);
        sleep(1);
    }
    return 0;
}
```