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

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#include <arpa/inet.h>

#include <string.h>
#include <signal.h>



int main(int argc, char *argv[]) {
    int soc;
    int scli;
    int result;
    struct sockaddr_in servidor;
    struct sockaddr_in cliente;
    socklen_t cliente_tam;
        FILE *ff;
        char buf[2000];
    int pid;

        soc = socket( PF_INET , SOCK_STREAM , 0 );
    printf("El socket es : %d \n",soc);
    if ( soc == -1 ) {
        printf("No hay soc\n");
        exit(-1);
    }

    servidor.sin_family = AF_INET;
    servidor.sin_port = htons(10000);
    servidor.sin_addr.s_addr = INADDR_ANY;


    result = bind( soc, (struct sockaddr *)&servidor , sizeof(servidor));
    printf("El resultado de bind es: %d \n",result);
    if ( result == -1 ) {
        printf("No vale el puerto\n");
        exit(-1);
    }
    printf("puerto asignado\n");


    listen(soc,5);

    while ( 1 ) {
        cliente_tam=sizeof(cliente);
        scli = accept( soc, (struct sockaddr *)&cliente , &cliente_tam );
        pid = fork();
        if ( pid == 0 ) {  /* es el hijo */
            /* Atender al cliente */
                        ff=fopen(scli,"r+");
                        if(ff==NULL) {
                                printf("No hay file\n");
                                close(scli);
                                close(soc);
                                exit(-1);
                        }
```

```
                fprintf(ff,"Hola, soy el servidor.\n");
                printf("Tenemos un cliente. Hacemos el ECHO.\n");
                while(fgets(buf,2000,ff) != NULL){
                        fprintf(ff,"%s",buf);
                    }
                fclose(ff);
                /* Fin cliente */
                exit(0);
            } /* fin de hijo */
        close(scli);
    } /* fin de while */
}
```