```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <signal.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <string.h>
#include <pthread.h>
#include <semaphore.h>
#include"header.h"

#define PORT1 "55558"



int main(int argc, char **argv){

//******variables*****
struct requete structure;

//initialisation structure
init_struct(&structure);

//initialisation du sémaphore
sem_init(&semaphore, 0, 1);



//******config socket*****************
int sock_fd, srvc_fd;
struct addrinfo s_init, *servinfo, *p;
struct sockaddr_storage client_addr;
socklen_t s_taille;

memset(&s_init, 0, sizeof(s_init));
s_init.ai_family = AF_UNSPEC;
s_init.ai_socktype = SOCK_STREAM;
s_init.ai_flags = AI_PASSIVE;
if (getaddrinfo(NULL, PORT1, &s_init, &servinfo) != 0) {
fprintf(stderr, "Erreur getaddrinfo\n");
exit(1);
}
for(p = servinfo; p != NULL; p = p->ai_next) {
if ((sock_fd = socket(p->ai_family, p->ai_socktype, p->ai_protocol)) == -1) {
perror("Serveur: socket");
continue;
}
if (bind(sock_fd, p->ai_addr, p->ai_addrlen) == -1) {
close(sock_fd);
perror("Serveur: erreur bind");
continue;
}
break;
}
if (p == NULL) {
fprintf(stderr, "Serveur: echec bind\n");
exit(2);
}
freeaddrinfo(servinfo);
if (listen(sock_fd, 5) == -1) {
        perror("listen");
```

```
exit(1);
}


//*****************************************

while(1){

s_taille = sizeof(client_addr);
srvc_fd = accept(sock_fd, (struct sockaddr *) &client_addr, &s_taille);

if (srvc_fd == -1) {
perror("accept");
continue;
}



printf("Nouvelle requete recue.\n");

//transfer d info pour fonction thread
socs.sock_fd = &sock_fd;
socs.structure= &structure;
socs.srvc_fd= &srvc_fd;

//creation du 1ier thread
        int err;
        if ((err = pthread_create(&threads[0], NULL, &thread_reception, NULL)) != 0) {
                printf("Echec de la création du thread: [%s]", strerror(err));
        }
                        printf("Création du thread reception numéro 0\n");
        pthread_join(threads[0], NULL);

}

close(sock_fd);
exit(0);
}
```