ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ

Κατερίνα Μανιουδάκη

2η Εργασία στο μάθημα **Λειτουργικά Συστήματα**

Ταύρος, 23 Ιανουαρίου 2015

Άσκηση 2

O server που υλοποιήθηκε είναι:

```
#include<stdio.h>
#include<string.h>    //strlen gia sygrish pinakwn xarakthrwn
#include<stdlib.h>    //strlen
#include<sys/socket.h>//gia epipedo metaforas dhmiourgia socket
#include<arpa/inet.h> //inet_addr gia diaxeirhsh diey8unsewn pou pername sta sockets
#include<unistd.h>    //write gia sockets
#include<pthread.h> //for threading , link with lpthread gia na ftiaxnw nhmata diergasiwn, gcc -o server it21335_server.c -lpthread
#include<time.h> //tin xrisimopoiw gia to seed tis rand()
//the thread function
#define INET4_ADDRSTRLEN 100 //kathorizo oti opoudipote sto programma to INET4_ADDRSTRLEN tha einai to 100
void *connection_handler(void *ptr);

int randInt(int a,int b);

int isInteger(char *string);

int main(int argc , char *argv[]) //argc arithmos entolwn  me tis opoies kaloume to executable tis main apo ti grammi entolwn, *argv[] o pinakas dio diastaseon pou periexei ta orismata pou kalese o xristis apo to command line to executable tou programmatos -c
{
	pthread_t connection_handling_thread;

      char client_IP_address[INET4_ADDRSTRLEN];

	if (argc != 3)
	{
	 printf("Usage:it21335_server <IP address> <Port number>\n Or:it21335_server -a <Port number> (binds to all available IP's')\n");
	 exit(0);
	}
	else
	{
         printf("executable name is %s\n",argv[0]);
         int server_socket_descriptor , client_socket_descriptor , c,*new_socket;
         struct sockaddr_in server , client;
         int serverport;
         char *IP_address;
         char bindall[3];
         int on=1;
         bindall[0]='-';
         bindall[1]='a';
         bindall[2]='\0';
         if (isInteger(argv[2]))
         serverport=atoi(argv[2]); // i sinartisi atoi metatrepei se akeraio oti exei dwsei o xristis
gia port number apo ti grammi entolw otan kalese to programma.
         else
         {
             printf("The server port number must be an integer number\n");
             exit(0);
         }

        if ((serverport < 0) || (serverport > 65535))
          {
              printf("The server port number must be an integer between 0-65535!!\n");
              exit(0);
          }
         //Create socket
         server_socket_descriptor = socket(AF_INET , SOCK_STREAM , 0);
        if (server_socket_descriptor == -1)
          {
           printf("Could not create socket");
           exit(0);
          }
         //puts("Socket created");
        printf("TCP Socket Created\n");
   if (setsockopt(server_socket_descriptor, SOL_SOCKET, SO_REUSEADDR, &on,
sizeof(on)) < 0)
     {
      printf("setsockopt(SO_REUSEADDR) failed\n");
        exit(0);
     }
    server.sin_family = AF_INET; //thelo internet domain socket address oxi unix domain
address AF_UNIX
```

```
     if (strcmp(bindall,argv[1]) == 0) //elenxo ti edose o xristis gia IP address -a (akou se oles tis
diathesimes) i sigkekrimmeni IP
             {
      server.sin_addr.s_addr = INADDR_ANY;
             }
             else
             {
       IP_address = argv[1];
       server.sin_addr.s_addr = inet_addr(IP_address); //apothikeuse sto address tou socket tin
IP address pou edose o xristis apo to command line
             }
    server.sin_port = htons(serverport);
    //Bind
    if( bind(server_socket_descriptor,(struct sockaddr *)&server , sizeof(server)) < 0) //me tin
bind denomaste me tin IP kai tin thira (port) pou exoume kathorisei stin domi sockaddr_in
    {
        //print the error message
        perror("bind failed. Error");
        return 1;
    }
    puts("bind done");

    //Listen
    listen(server_socket_descriptor,0);  //kathorizei to megethos tis ouras anamonis client pou
perimenoun na sindethoun!!! 0=unlimited allios 1,2,3.....

    //Accept and incoming connection
    puts("Waiting for incoming connections...");
    c = sizeof(struct sockaddr_in);

    while( (client_socket_descriptor = accept(server_socket_descriptor, (struct sockaddr
*)&client, (socklen_t*)&c)) )
    {
        memset(client_IP_address, 0,sizeof(client_IP_address));
        if
(inet_ntop(AF_INET,&client.sin_addr.s_addr,client_IP_address,sizeof(client_IP_address)) !=
NULL)
          printf("Connection from client:%s and from port:%d
accepted.\n",client_IP_address,ntohs(client.sin_port));

        if (pthread_create( &connection_handling_thread , NULL ,  connection_handler ,
&client_socket_descriptor) < 0) //pthread_create einai i sinartisi tis vivliothikis pthread.h i
opoia dimiourgei ena neo nima stin diergasia tou server kathe fora pou i accept apodexetai
mia nea sindesi pelati, i pthread_create
        {
            perror("could not create thread");
            return 1;
        }

        //Now join the thread , so that we dont terminate before the thread
        //pthread_join( thread_id , NULL);
        puts("Handler assigned");
    }

    if (client_socket_descriptor < 0)
    {
        perror("accept failed");
        return 1;
    }

    return 0;

   }
}

/*
 * This will handle connection for each client
 * */
void *connection_handler(void *socket_desc)
{
    //Get the socket descriptor
    //char protocolcommand1[100];
    char *gamevariables[3];
    gamevariables[0]="Rock";
    gamevariables[1]="Paper";
    gamevariables[2]="Scissors";
    char *protocolcommand1;
    char *protocolresponse1;
    char *protocolresponse2;
    char *protocolresponse3;
    char *protocolresponse4;
    char *protocolresponse5;
    int serverwins=0,clientwins=0,draws=0;
   // strcpy(protocolcommand1,"disconnect\r\n");
    protocolcommand1="quit";
    protocolresponse1="disconnected";
    protocolresponse2="You sent me an unknown command!!";
    protocolresponse3="Nobody Wins,DRAW";
    protocolresponse4="Client Wins";
    protocolresponse5="Server Wins";
    printf("Thread started\n");
    int sock = *(int *)socket_desc; //i metavliti akeraiou sock dexetai tin metavliti socket me tin
opoia kalesame tin sinartisi connection_handler i opoia pairnei san orisma to
client_socket_descriptor apo tin klisi tis accept.
    int read_size=0,run=1,serverchoice=0,clientchoice=0;
    char *message , client_message[2000];
    //Send some messages to the client
   // memset(client_message, 0,sizeof(client_message));
   // message = "Greetings!,Welcome to the Rock,Paper,Scissors Game Server\n";
   // write(sock , message , strlen(message)); //i write klisi sistimatos grafei dedomena sto
client socket

    //message = "Now type something and i shall repeat what you type \n";
    //write(sock , message , strlen(message));

    //Receive a message from client
    while(run)
    {
        read_size = recv(sock , client_message ,sizeof(client_message),0);//recv i klisi sistimatos
diavazei dedomena apo to client socket
            if (read_size == 0)
            break;
        client_message[read_size] = '\0';
            if (strcmp(client_message,protocolcommand1) == 0) //strcmp elenxei isotita string an
einai isa gyrizei 0
        {
          write(sock , protocolresponse1 , strlen(protocolresponse1)); //i strlen ,ou gyrizei to
megethos tou string pou exei mesa tou apothikeumeno enas pinakas xaraktirwn
          close(sock);
          read_size=0;
          break;
        } else
        if (strcmp(client_message,gamevariables[0]) == 0)
```

```
{
  printf("Client sent:%s\n",client_message);
  clientchoice=0;
  serverchoice=randInt(0,2);
  if (serverchoice == 0)
      {
    write(sock , protocolresponse3 ,strlen(protocolresponse3));
        draws++;
      }
      else if (serverchoice == 2)
      {
    write(sock , protocolresponse4 ,strlen(protocolresponse4));
        clientwins++;
      }
      else if (serverchoice == 1)
      {
   write(sock , protocolresponse5 ,strlen(protocolresponse5));
       serverwins++;
      }
} else
if (strcmp(client_message,gamevariables[1]) == 0)
{
  printf("Client sent:%s\n",client_message);
  clientchoice=1;
  serverchoice=randInt(0,2);
  if (serverchoice == 0)
      {
   write(sock , protocolresponse4 ,strlen(protocolresponse4));
       clientwins++;
      }
      else if (serverchoice == 2)
      {
   write(sock , protocolresponse5 ,strlen(protocolresponse5));
       serverwins++;
      }
      else if (serverchoice == 1)
      {
   write(sock , protocolresponse3 ,strlen(protocolresponse3));
       draws++;
      }
} else
```

```
    if (strcmp(client_message,gamevariables[2]) == 0)
    {
     printf("Client sent:%s\n",client_message);
     clientchoice=2;
     serverchoice=randInt(0,2);
     if (serverchoice == 0)
       {
      write(sock , protocolresponse5 ,strlen(protocolresponse5));
        serverwins++;
       }
     else if (serverchoice == 2)
       {
      write(sock , protocolresponse3 ,strlen(protocolresponse3));
        draws++;
       }
     else if (serverchoice == 1)
       {
      write(sock , protocolresponse4 ,strlen(protocolresponse4));
        clientwins++;
       }
    }
    else
    {//Send a message back to client informing that he sent an unknown command!!
     write(sock , protocolresponse2 ,strlen(protocolresponse2));
    }
            //clear the message buffer,katharizo ton pinaka xaraktirwn client_message
diladi kano ola tou ta stoixeia miden.
            memset(client_message, 0,sizeof(client_message));
  }

  if (read_size == 0)
  {
    puts("Client disconnected\n");
      printf("Game Session Statistics->Server Wins:%d,Client
Wins:%d,Draws:%d\n",serverwins,clientwins,draws);
    fflush(stdout); //i fflush katahrizei to periexomeno tou arxeiou sistimatos pou tis
kathorizeis, sto Unix ola ta perifereiaka einai arxeia px stdin arxeio=keyboard stdout arxeio=
othoni
  }
  else if(read_size == -1)
  {
        perror("recv failed");
    }
    return 0;
}

int randInt(int a,int b)
{
  int max=b;
  int min=a;
  int random;
  srand(time(NULL));
  random=(rand() % (max+1-min))+min;
  return random;
}

int isInteger(char *string)
{
        int i, stringLength = strlen(string),boolean;
            boolean=1;

        for(i = 0; i < stringLength; i++)
        {
                if(isdigit(string[i]) == 0)
                       {
                         boolean=0;
                  break;
                       }
        }

      return boolean;
}
```

Ο client που υλοποιήθηκε είναι:

```
#include <stdio.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>
```

```
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <unistd.h>
#include <ctype.h>

#define MAX_SIZE 50

int isInteger(char *string);

int main(int argc,char *argv[])
{
    int sock_desc,server_port,connected=0,read_size;
    char *protocolresponse="quit";
    struct sockaddr_in serv_addr;
    char sbuff[MAX_SIZE],rbuff[MAX_SIZE];

    if (argc != 3)
    {
      printf("Usage:it21335_client <Server IP address> <Server Port number>\n");
      exit(0);
    }
    else
    {
      if (isInteger(argv[2]))
      server_port=atoi(argv[2]);
```

```
else
{
    printf("The server port number must be an integer number\n");
    exit(0);
}

if ((server_port < 0) || (server_port > 65535))
{
    printf("The server port number must be an integer between 0-65535!!\n");
    exit(0);
}

if((sock_desc = socket(AF_INET, SOCK_STREAM, 0)) < 0)
  printf("Failed creating socket\n");

bzero((char *) &serv_addr, sizeof (serv_addr));
serv_addr.sin_family = AF_INET;
serv_addr.sin_addr.s_addr = inet_addr(argv[1]);
serv_addr.sin_port = htons(server_port);

if (connect(sock_desc, (struct sockaddr *) &serv_addr, sizeof (serv_addr)) < 0) {
  printf("Failed to connect to server\n");
  return -1;
 }
 else
 {
```

```
           connected=1;
        printf("Connected successfully\n");
        printf("Write one of the following commands:(Rock | Paper | Scissors)\n");
        printf("Or type quit to disconnect from server!!\n");
        while(connected)
        {
             printf("Give command:\n");
             bzero(sbuff,MAX_SIZE);
             scanf("%s",&sbuff);
             write(sock_desc , sbuff , strlen(sbuff));
          if (strcmp(sbuff,protocolresponse) == 0)
             {
               printf("disconnected from server!!\n");
               connected=0;
               break;
             }
             read_size=recv(sock_desc,rbuff,MAX_SIZE,0);
          if(read_size ==0)
           printf("Receive Error\n");
          else
             {
              rbuff[read_size]='\0';
              printf("Server Response:%s\n",rbuff);
             }
          bzero(rbuff,MAX_SIZE);//to clean buffer-->IMP otherwise previous word characters
also came
```

```
       }
        close(sock_desc);
     }
    }

    return 0;
}

int isInteger(char *string)
{
      int i, stringLength = strlen(string),boolean;
        boolean=1;

      for(i = 0; i < stringLength; i++)
      {
           if(isdigit(string[i]) == 0)
                {
                  boolean=0;
             break;
                }
      }

     return boolean;
}
```

Μια ενδεικτική εκτέλεση είναι η παρακάτω:

server	client
it21335_server 192.168.1.113 1800 ή it21335_server -a 1800 (με το -a κάνει bind σε όλες τις διαθέσιμες IP του μηχανήματος που εκτελείται ο server)	it21335_client 192.168.1.113 1800 ή it21335_client 127.0.0.1 1800, αν έχεις τρέξει το server με -a και ο client εκτελείται στο ίδιο μηχάνημα με το server.

Σχόλια/Παρατηρήσεις

Συνοπτικός Πίνακας Υλοποιούμενων Λειτουργιών

Λειτουργία	Υλοποιήθηκε (ΝΑΙ/ΟΧΙ/ΜΕΡΙΚΩΣ)	Παρατηρήσεις
Επικοινωνία client-server μέσω network stream socket	**ΝΑΙ**	
Πολυνηματικός server	**ΝΑΙ**	
Παιχνίδι «Πέτρα-Ψαλίδι-Χαρτί»	**ΝΑΙ**	
Εμφάνιση πληροφοριών του client στον server	**ΝΑΙ**	
Bonus: εμφάνιση στατιστικών στον server	**ΝΑΙ**	