**Προηγμένες Δικτυακές Tεχνολογίες**

**Sockets**

Μέρος 3o

|  |  |
| --- | --- |
| ΗΜΕΡΟΜΗΝΙΑ ΥΠΟΒΟΛΗΣ: | **30 Νοεμβρίου 2020** |
| ΟΝΟΜΑΤΕΠΩΝΥΜΟ: | **Ντρίτσος Γεώργιος** |
| ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ: | **00078** |

1. Υλοποίηση client/server εφαρμογής με χρήση TCP

Client\_TCP

#include <netdb.h>

#include <stdio.h>

#include <string.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <stdlib.h>

#define MAX 80

#define PORT 2019

#define IP "127.0.0.1"

int main ()

{

int sockfd=socket(AF\_INET,SOCK\_STREAM,0);

if(sockfd<0)

{

printf("APOTYXIA DHMIOYRGEIAS SOCKET");

exit(0);

}

else printf("TO SOCKET DHMIOYRGHTHIKE ME EPITYXIA\n");

struct sockaddr\_in servaddr;

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_addr.s\_addr= inet\_addr(IP);

servaddr.sin\_port=htons(PORT);

printf("\n PROSPATHEIA SYNDESIS...\n");

if(connect(sockfd,(struct sockaddr\*) &servaddr,sizeof(servaddr))<0)

{

printf("\n APOTYXIA SYNDESHS ME TO DIAKOMISTH\n");

exit(0);

}

else

{

printf("SYNDEDEMENO\n");

char buff[MAX];

read(sockfd,buff,sizeof(buff));

printf("\n O DIAKOMISTHS LEEI %s\n",buff );

}

close(sockfd);

printf("H SYNDESH XATHIKE\n");

}

Server\_TCP

#include <string.h>

#include <stdio.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <unistd.h>

#include <netinet/in.h>

#include <stdlib.h>

#define MAX 80

#define PORT 2019

#define SA struct sockaddr

int main ()

{

//int connfd.len;

//struct sockaddr\_in r;

//struct sockaddr\_in cli;

int sockfd = socket(AF\_INET, SOCK\_STREAM,0);

if (sockfd==-1)

{

printf("APOTYXIA DHMIOYRGEIAS SOCKET");

exit(0);

}

else printf("TO SOCKET DHMIOYRGHTHIKE ME EPITYXIA\n");

struct sockaddr\_in servaddr;

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family= AF\_INET;

servaddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

servaddr.sin\_port= htons(PORT);

if((bind(sockfd, (struct sockaddr\*)&servaddr , sizeof(servaddr)))!=0)

{

printf("H DESMEFSI TOY SOCKET APETYXE\n");

exit(0);

}

else

printf("EPITYXHMENH DESMEYSH SOCKET\n");

if ((listen(sockfd,5))!=0)

{

printf("DEN AKOYEI...\n");

exit(0);

}

else printf("O DIAKOMISTHS AKOYEI...\n");

struct sockaddr\_in cli;

int len= sizeof(cli);

int new\_socket = accept(sockfd,(struct sockaddr\*)&cli,(socklen\_t\*) &len);

if(new\_socket <0)

{

printf("APOTYXIA SYNDESHS ME DIAKOMISTH\n");

exit(0);

}

else

{

printf("\n NEOS PELATHS SYNDETHIKE\n");

char buff[]="hello client";

write(new\_socket, &buff,sizeof(buff));

printf("%s\n", buff);

}

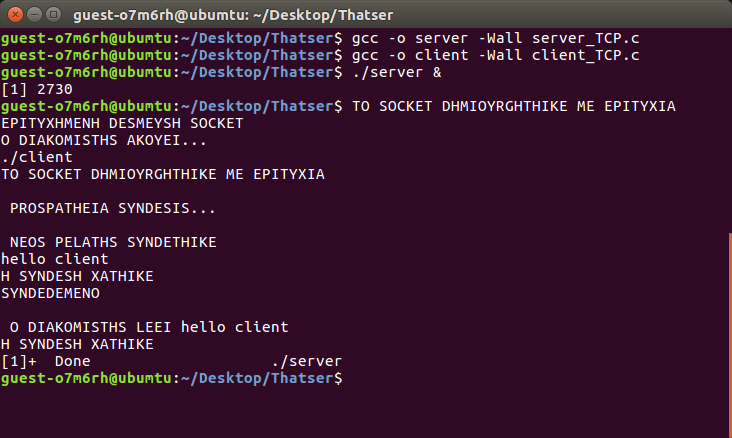
close(sockfd);

printf("H SYNDESH XATHIKE\n");

}

1. –Compile and Run

Αφού κάνουμε compile Server & client με την εντολη ./server & η λειτουργία του server ξεκινάει και λαμβάνει την πόρτα 2730 (όπως βλέπουμε και στην εικόνα 1).



Εικόνα

Η εντολή ./client δημιουργεί ένα socket που επιτρέπει την επικοινωνία του client με τον server στην πόρτα 2730 και ip την localhost 127.0.0.1 τέλος τυπώνει τα μηνύματα

3.–Τροποποίηση

server\_3

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#define maxlenrcv 2014

int main(){

int port;

printf("DOSE MOY THN PORTA:");

scanf("%d",&port);

char ip[15];

printf("DOSE MOY THN IP:");

scanf("%s",ip);

int sockfd, ret;

struct sockaddr\_in serverAddr;

int newSocket;

struct sockaddr\_in newAddr;

socklen\_t addr\_size;

char buffer[1024];

pid\_t childpid;

sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if(sockfd < 0){

printf("PROVLIMA STHN SYNDESH.\n");

exit(1);

}

printf("EPITYXHMENH DHMIOYRGEIA SOCKET.\n");

memset(&serverAddr, '\0', sizeof(serverAddr));

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(port);

serverAddr.sin\_addr.s\_addr = inet\_addr(ip);

ret = bind(sockfd, (struct sockaddr\*)&serverAddr, sizeof(serverAddr));

if(ret < 0){

printf("LATHOS STHN DESMEYSH TOY SOCKET.\n");

exit(1);

}

printf("DESMEYMENH PORTA %d\n", port);

if(listen(sockfd, 10) == 0){

printf("AKOYEI.\n");

}else{

printf("LATHOS.\n");

}

while(1){

newSocket = accept(sockfd, (struct sockaddr\*)&newAddr, &addr\_size);

if(newSocket < 0){

exit(1);

}

printf("H SYNDESH APODEXTHIKE APO %s:%d\n", inet\_ntoa(newAddr.sin\_addr), ntohs(newAddr.sin\_port));

if((childpid = fork()) == 0){

close(sockfd);

while(1){

recv(newSocket, buffer, 1024, 0);

if(strcmp(buffer, "exit\n") == 0){

printf("APOSYNDESH APO %s:%d\n", inet\_ntoa(newAddr.sin\_addr), ntohs(newAddr.sin\_port));

break;

}else{

printf("MHNYMA PELATH APO THN DIEYTHINSI %d is: %s\n", ntohs(newAddr.sin\_port), buffer);

printf("MHNYMA STON PELATH:");

//scanf("%s", buffer);

fgets(buffer, 1024, stdin);

send(newSocket, buffer, strlen(buffer), 0);

memset(buffer,'\0', sizeof(buffer));

}

}

}

}

close(newSocket);

return 0;

}

Client\_3

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#define MAXRCVLEN 1024

int main(){

int port;

printf("DOSE MOY THN PORT:");

scanf("%d",&port);

char ip[15];

printf("DOSE MOY THN IP:");

scanf("%s",ip);

int clientSocket, ret;

struct sockaddr\_in serverAddr;

char buffer[1024];

clientSocket = socket(AF\_INET, SOCK\_STREAM, 0);

if(clientSocket < 0){

printf("LATHOS STHN SYNDESH.\n");

exit(1);

}

printf("EPITYXHMENH DHMIOYRGIA SOCKET.\n");

memset(&serverAddr, '\0', sizeof(serverAddr));

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(port);

serverAddr.sin\_addr.s\_addr = inet\_addr(ip);

ret = connect(clientSocket, (struct sockaddr\*)&serverAddr, sizeof(serverAddr));

if(ret < 0){

printf("Connection Error.\n");

exit(1);

}

printf("LATHOS STHN SYNDESH.\n");

while(1){

memset(buffer,'\0', sizeof(buffer));

printf("Client: \t");

fgets(buffer, 1024, stdin);

//scanf("%s", buffer);

//scanf("%s", &buffer[0]);

send(clientSocket, buffer, strlen(buffer), 0);

if(strcmp(buffer, "exit\n") == 0){

close(clientSocket);

printf("APOSYNDESH APO TO DIAKOMISTH.\n");

send(clientSocket, buffer, strlen(buffer), 0);

exit(1);

}

if(recv(clientSocket, buffer, 1024, 0) < 0){

printf("LATHOS STHN EPIKOINWNIA\n");

}else{

printf("DIAKOMISTHS: \t%s\n", buffer);

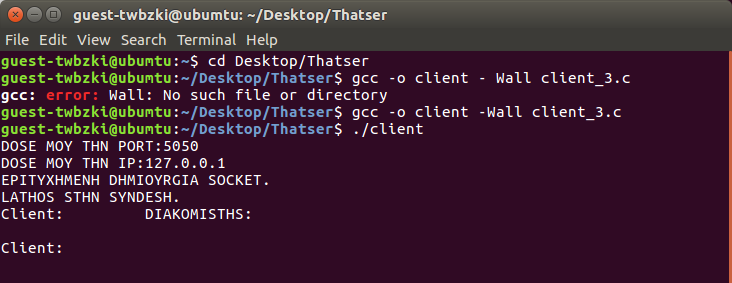
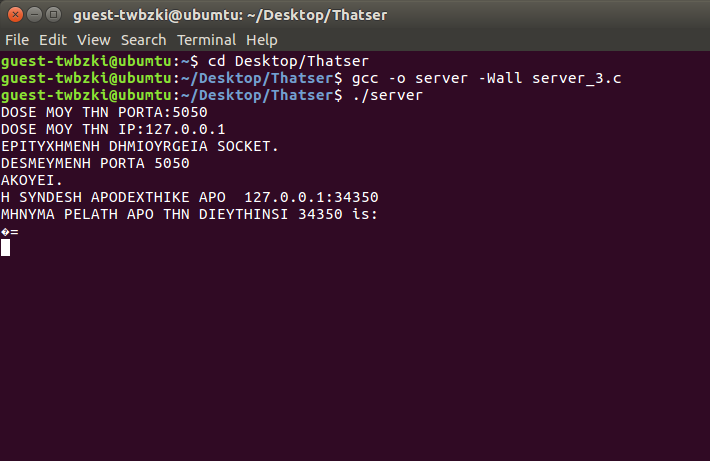
}

}

return 0;

}

Ακολουθεί screenshot από την διαδικασία εκτέλεσης των προγραμμάτων (εικόνα 2)



Εικόνα 2

4.–Υλοποίηση client/server εφαρμογής με χρήση UDP

Client

#include <netdb.h>

#include <stdio.h>

#include <string.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <stdlib.h>

#include <sys/types.h>

#define MAX 80

#define PORT 4045

#define IP "127.0.0.1"

#define MAXLINE 1024

int main ()

{

int sockfd=socket(AF\_INET,SOCK\_DGRAM,0);

if(sockfd<0)

{

printf("APOTYXIA DHMIOYRGEIAS SOCKET");

exit(0);

}

else printf("TO SOCKET DHMIOYRGHTHIKE ME EPITYXIA\n");

struct sockaddr\_in servaddr;

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_addr.s\_addr= inet\_addr(IP);

servaddr.sin\_port=htons(PORT);

char msg[]="hello server";

sendto(sockfd,(const char \*)msg, strlen(msg),

MSG\_CONFIRM,(const struct sockaddr\*)&servaddr,

sizeof(servaddr));

printf("%s\n",msg);

int len;

char buff[MAX];

int n=recvfrom(sockfd,(char\*)buff,MAXLINE,

MSG\_WAITALL,(struct sockaddr\*) &servaddr,

(socklen\_t\*)&len);

buff[n]='\n';

printf("O DIAKOMISTHS ANTAPOKRITHIKE: %s\n",buff);

close (sockfd);

printf("H SYNDESH XATHIKE\n");

}

Server

#include <netdb.h>

#include <stdio.h>

#include <string.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <stdlib.h>

#include <sys/types.h>

#define MAX 80

#define PORT 4045

#define IP "127.0.0.1"

#define MAXLINE 1024

#define SA struct sockaddr

int main ()

{

int sockfd=socket(AF\_INET,SOCK\_DGRAM,0);

if(sockfd==-1)

{

printf("APOTYXIA DHMIOYRGEIAS SOCKET");

exit(0);

}

else printf("TO SOCKET DHMIOYRGHTHIKE ME EPITYXIA\n");

struct sockaddr\_in servaddr;

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

servaddr.sin\_port=htons(PORT);

if((bind(sockfd,(struct sockaddr\*)&servaddr,sizeof(servaddr)))!=0)

{

printf("H DESMEFSI TOY SOCKET APETYXE\n");

exit(0);

}

else printf("EPITYXHMENH DESMEYSH SOCKET\n");

struct sockaddr\_in cli;

int len=sizeof(cli);

char buff[MAX];

int n=recvfrom(sockfd, (char \*)buff, MAXLINE,

MSG\_WAITALL, (struct sockaddr\* ) &cli,

(socklen\_t\*) &len);

buff[n]='\0';

printf(";\n NEOS PELATHS SYNDETHIKE: %s\n",buff);

char msg[]=" hello client";

sendto(sockfd,(const char\*)msg,strlen(msg),

MSG\_CONFIRM,( const struct sockaddr\*)&cli,

len);

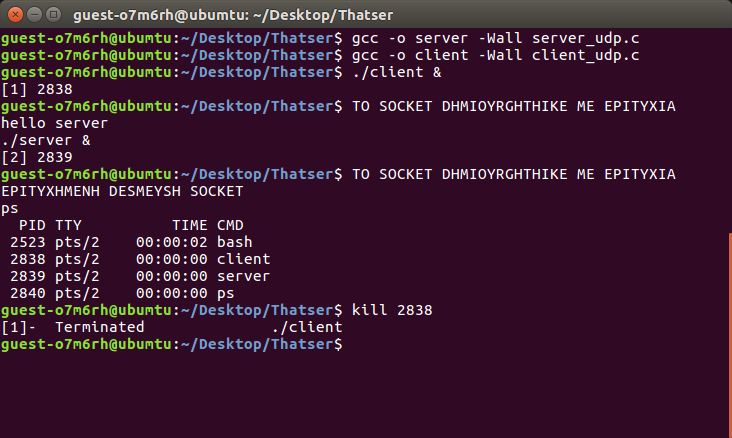
printf("%s\n",msg);

close(sockfd);

printf(" H SYNDESH XATHIKE\n");

}

Ακολουθεί screenshot (Εικόνα 3) με την εκτέλεση των server & client



Εικόνα 3

5.Υλοποίηση Client-Server εφαρμογής συνομιλίας με χρήση πρωτοκόλλου TCP ή UDP

Server\_5

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#define PORT 4045

int main(){

int sockfd, ret;

struct sockaddr\_in serverAddr;

int newSocket;

struct sockaddr\_in newAddr;

socklen\_t addr\_size;

char buffer[1024];

pid\_t childpid;

sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if(sockfd < 0){

printf("Apotyxia syndesis.\n");

exit(1);

}

printf("To Socket dimiourgithike me epitixia.\n");

memset(&serverAddr, '\0', sizeof(serverAddr));

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(PORT);

serverAddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

ret = bind(sockfd, (struct sockaddr\*)&serverAddr, sizeof(serverAddr));

if(ret < 0){

printf("Lathos stin desmeysi tou socket.\n");

exit(1);

}

printf("Desmeymeni Porta: %d\n",4045);

if(listen(sockfd, 10) == 0){

printf("Akoyei o diakomistis ..\n");

}else{

printf("Lathos stin desmeysi tou socket .\n");

}

while(1){

newSocket = accept(sockfd, (struct sockaddr\*)&newAddr, &addr\_size);

if(newSocket < 0){

exit(1);

}

printf("Apodoxi syndesis apo: %s:%d\n", inet\_ntoa(newAddr.sin\_addr), ntohs(newAddr.sin\_port));

if((childpid = fork()) == 0){

close(sockfd);

while(1){

recv(newSocket, buffer, 1024, 0);

if(strcmp (buffer, "exit\n")==0)

{

printf("Aposindesi apo %s:%d\n", inet\_ntoa(newAddr.sin\_addr), ntohs(newAddr.sin\_port));

break;

}else{

printf("Diakomistis : %s\n", buffer);

send(newSocket, buffer, strlen(buffer), 0);

bzero(buffer, sizeof(buffer));

}

}

}

}

close(newSocket);

return 0;

}

Client\_5

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#define PORT 4045

int main(){

int clientSocket, ret;

struct sockaddr\_in serverAddr;

char buffer[1024];

clientSocket = socket(AF\_INET, SOCK\_STREAM, 0);

if(clientSocket < 0){

printf("Apotyxia syndesis.\n");

exit(1);

}

printf(" To Socket dimiourgithike me epitixia.\n");

memset(&serverAddr, '\0', sizeof(serverAddr));

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(PORT);

serverAddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

ret = connect(clientSocket, (struct sockaddr\*)&serverAddr, sizeof(serverAddr));

if(ret < 0){

printf("Lathos stin syndesi.\n");

exit(1);

}

printf("O diakomistis einai syndedemenos.\n");

while(1){

printf("Peite to mynhma sas: \t");

scanf("%s", &buffer[0]);

send(clientSocket, buffer, strlen(buffer), 0);

if(strcmp(buffer, "exit\n") == 0)

{

close(clientSocket);

printf(buffer,"O diakomistis aposyndethike\n");

send(clientSocket ,buffer,strlen(buffer),0);

exit(1);

}

if(recv(clientSocket, buffer, 1024, 0) < 0){

printf("Provlima stin epikoinwnia\n");

}else{

printf("Diakomistis: \t%s\n", buffer);

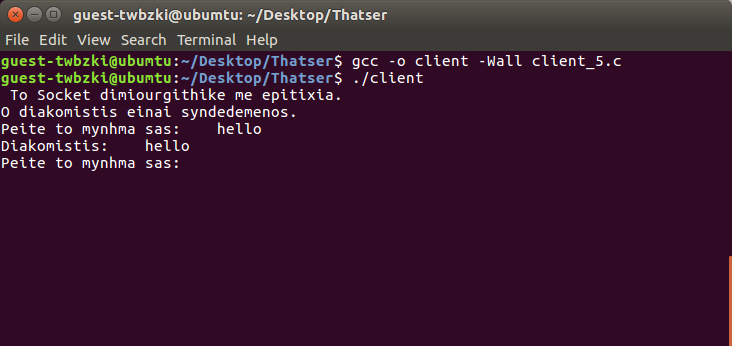
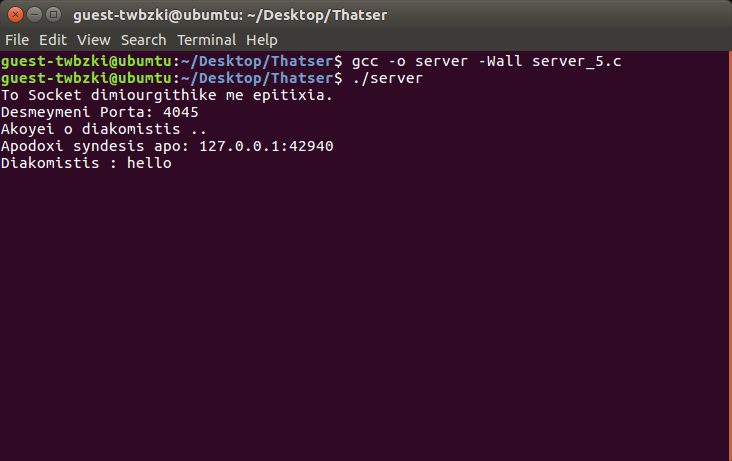
}

}

return 0;

}

**Ακολουθεί screenshot εκτέλεσης των προγραμμάτων (εικόνα 4)**



Εικόνα 4