**ITERATIVE SERVER**

ROLL NO.:1905007

NAME : R.ASWINRAJA

Ex.No:6

Date:30.09.2021

**AIM:**

To implement an Iterative Server program for the Blood Bank Management System.

**PROBLEM STATEMENT:**

Write a simple Iterative Server program to implement the Blood Bank Management System

* Each client sends a blood group for donating.
* The server receives the information and replies to the client on acceptance or denial.
* Under acceptance condition, the server updates the blood unit and process the next client.
* In denial the server just thanks to the client and processes the next client.

**PROGRAM:**

**1. SERVER CODE:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<string.h>

#include<stdlib.h>

#include<netinet/in.h>

#define PORT 8080

#define MAXSZ 100

int main()

{ int sockfd;

int newsockfd;

struct sockaddr\_in serverAddress;

struct sockaddr\_in clientAddress;

int n,n1;

char msg[MAXSZ];

char msg1[MAXSZ];

char msg2[MAXSZ];

char msg3[MAXSZ];

int units[4][2]={0};

int clientAddressLength;

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

memset(&serverAddress,0,sizeof(serverAddress));

serverAddress.sin\_family=AF\_INET;

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

serverAddress.sin\_port=htons(PORT);

bind(sockfd,(struct sockaddr \*)&serverAddress, sizeof(serverAddress));

listen(sockfd,5);

while(1)

{ printf("\n\n\*\*\*\*\*server waiting for new client connection:\*\*\*\*\*\n");

clientAddressLength=sizeof(clientAddress);

newsockfd=accept(sockfd,(struct sockaddr\*)&clientAddress,&clientAddressLength);

while(1)

{ n=recv(newsockfd,msg,MAXSZ,0);

if(n==0 || msg[0]=='#')

{ close(newsockfd); break; }

printf("\n\nBlood Group : %s",msg);

printf("Accept or reject (1/0) : ");

fgets(msg2,MAXSZ,stdin);

send(newsockfd,msg2,n,0); msg[n]=0;

recv(newsockfd,msg3,MAXSZ,0);

int unit=atoi(msg3);

if(msg[0]=='A' && msg[1]=='+')

units[0][0]=units[0][0]+unit;

else if(msg[0]=='A' && msg[1]=='-')

units[0][1]=units[0][1]+unit;

else if(msg[0]=='B' && msg[1]=='+')

units[1][0]=units[1][0]+unit;

else if(msg[0]=='B' && msg[1]=='-')

units[1][1]=units[1][1]+unit;

else if(msg[0]=='O' && msg[1]=='+')

units[2][0]=units[2][0]+unit;

else if(msg[0]=='O' && msg[1]=='-')

units[2][1]=units[2][1]+unit;

else if(msg[0]=='A' && msg[1]=='B' && msg[2]=='+')

units[3][0]=units[3][0]+unit;

else if(msg[0]=='A' && msg[1]=='B' && msg[2]=='-')

units[3][1]=units[3][1]+unit;

else printf("\nno such blood group available\n");

printf("Blood groups and units available\n");

printf("A+ve\tA-ve\tB+ve\tB-ve\tO+ve\tO-ve\tAB+ve\tAB-ve\n");

for(int ind=0;ind<4;ind++)

printf("%d\t%d\t",units[ind][0],units[ind][1]);

}

}

return 0;

}

**2.CLIENT CODE:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<string.h>

#include<stdlib.h>

#include<netinet/in.h>

#define PORT 8080

#define SERVER\_IP "127.0.0.1"

#define MAXSZ 100

int main()

{ int sockfd;

struct sockaddr\_in serverAddress;

int n,n1,n2;

char msg[MAXSZ];

char msg1[MAXSZ];

char msg2[MAXSZ];

char msg3[MAXSZ];

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

memset(&serverAddress,0,sizeof(serverAddress));

serverAddress.sin\_family=AF\_INET;

serverAddress.sin\_addr.s\_addr=inet\_addr(SERVER\_IP);

serverAddress.sin\_port=htons(PORT);

connect(sockfd,(struct sockaddr \*)&serverAddress,sizeof(serverAddress));

printf("Enter # to exit connection\n");

while(1)

{

printf("\nEnter your blood group (A/B/AB/O) (+/-) : ");

fgets(msg1,MAXSZ,stdin);

if(msg1[0]=='#')

{

send(sockfd,msg1,n,0);

break;

}

n=strlen(msg1)+1;

n1=strlen(msg2)+1;

send(sockfd,msg1,n,0);

n=recv(sockfd,msg3,MAXSZ,0);

if(msg3[0]=='1')

{

printf("Enter units:");

fgets(msg,MAXSZ,stdin);

n2=strlen(msg)+1;

send(sockfd,msg,n2,0);

printf("Accepted\n");

}

else

{

bzero(msg,MAXSZ);

send(sockfd,msg,n,0);

printf("Sry our limit for blood execeded,thank you.\n");

break;

}

}

return 0;

}

**OUTPUT:**

![Text

Description automatically generated]()**SERVER:**

**CLIENT 1:**

![Text

Description automatically generated]()

![Text

Description automatically generated]()**SERVER SIDE DURING CLIENT 1 REQUESTS:**

![Text

Description automatically generated]()**CLIENT 2:**

![Text

Description automatically generated]()**SERVER SIDE DURING CLIENT 2 REQUESTS**:

**RESULT:**

Thus, the implementation of an Iterative Server program for the Blood Bank Management System has been created and executed successfully.