COMPUTER NETWORKS

1.Chat Application

Server:

```c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <string.h>
#include <netinet/in.h>

#define TRUE 1
#define FALSE 0

#define MAX 4

void e(char *msg)
{
  perror(msg);
  exit(1);
}

int tcp_server(int PORT)
{
  int server;
  server = socket(AF_INET, SOCK_STREAM, 0);
  if (server < 0)
    e("Socket Creation error");

  int opt = TRUE;
  if (setsockopt(server, SOL_SOCKET, SO_REUSEADDR, &opt, (socklen_t)sizeof(opt)) < 0)
    e("Option Set Unsuccessful");

  struct sockaddr_in address;
  socklen_t addrlen = sizeof(address);

  bzero(&address, sizeof(address));

  address.sin_family = AF_INET;
  address.sin_addr.s_addr = htonl(INADDR_ANY);
  address.sin_port = htons(PORT);

  if (bind(server, (struct sockaddr *)&address, addrlen) < 0)
    e("Bind Unsuccessfull");
```

```c
  if (listen(server, MAX) < 0)
    e("Can't Listen");

  return server;
}

void r_and_w(int client)
{

  int n;
  char buffer[255] = {0};
  while ((n = recv(client, &buffer, 255, 0)) > 0)
  {
    printf("\nClient : %s", buffer);
    bzero(buffer, 255);

    printf("Server : ");
    fgets(buffer, 255, stdin);
    send(client, buffer, 255, 0);
  }

  // recv(client, &buffer, 255, 0);
}

int main()
{
  int port = 2020;
  int server = tcp_server(port);

  struct sockaddr_in cliaddress;
  socklen_t addrlen = sizeof(cliaddress);
  bzero(&cliaddress, addrlen);
  printf("Server Listening\n");
  int client = accept(server, (struct sockaddr *)&cliaddress, &addrlen);
  if (client < 0)
    e("Can't Accept Connections");

  printf("Client Connected from IP: %s and PORT: %d\n", inet_ntoa(cliaddress.sin_addr), ntohs(cliaddress.sin_port));

  r_and_w(client);

  close(client);
  close(server);
  return 0;
}
```

Client:

```c
#include <stdio.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <arpa/inet.h>

void e(char *msg)
{
  perror(msg);
  exit(1);
}

int tcp_client(const char *addr, int port)
{

  int client;
  client = socket(AF_INET, SOCK_STREAM, 0);
  if (client < 0)
    e("Socket creation error");

  //Optional Bind
  struct sockaddr_in address;
  socklen_t addrlen = sizeof(address);
  bzero(&address, addrlen);

  address.sin_family = AF_INET;
  address.sin_addr.s_addr = htonl(INADDR_ANY);
  address.sin_port = htons(14523);

  if (bind(client, (struct sockaddr *)&address, addrlen))
    e("Bind Error");

  if (inet_pton(AF_INET, addr, &address.sin_addr) <= 0)
    e("Address error");

  address.sin_port = htons(port);

  if (connect(client, (struct sockaddr *)&address, addrlen))
    e("Connection error");

  return client;
}

void r_and_w(int client)
{
  int n;
  char buffer[255] = {0};
```

```c
  do
  {
    printf("\nClient : ");
    fgets(buffer, 255, stdin);
    send(client, buffer, 255, 0);
    bzero(buffer, 255);

    recv(client, &buffer, 255, 0);
    printf("Server : %s", buffer);
  } while (strncmp(buffer, "bye", 3));
}

int main()
{
  int client = tcp_client("127.0.0.1", 2020);

  r_and_w(client);
  close(client);
}
```

Terminal:

```
saisrini@saisrinis-MacBook-Air chatApp % cd "/Users/saisrini/IdeaProjects/lab1/chatApp/" && gcc server.c -o server && "/Users/saisrini/IdeaProjects/lab1/chatApp/"server
Server Listening
Client Connected from IP: 127.0.0.1 and PORT: 14523

Client : Good morning server
Server : GM! what you want?

Client : Ntg! Just to say hi!
Server : Oh..Bye..then

Client : bye
Server : bye
saisrini@saisrinis-MacBook-Air chatApp %
```

```
/"server
Bind Unsuccessfull: Address already in use
saisrini@saisrinis-MacBook-Air chatApp % cd "/Users/saisrini/IdeaProjects/lab1/chatApp/" && gcc client.c -o client && "/Users/saisrini/IdeaProjects/lab1/chatApp/"client

Client : Good morning server
Server : GM! what you want?

Client : Ntg! Just to say hi!
Server : Oh..Bye..then

Client : bye
Server : bye
saisrini@saisrinis-MacBook-Air chatApp %
```

2.Date and Time

Server:

```c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <string.h>
#include <netinet/in.h>
#include <time.h>

#define TRUE 1
#define FALSE 0
#define MAX 4

void e(char *msg)
{
  perror(msg);
  exit(1);
}

int tcp_server(int PORT)
{

  int server;
  server = socket(AF_INET, SOCK_STREAM, 0);
  if (server < 0)
    e("Socket Creation error");

  int opt = TRUE;
  if (setsockopt(server, SOL_SOCKET, SO_REUSEADDR, &opt, (socklen_t)sizeof(opt)) < 0)
    e("Option Set Unsuccessful");

  struct sockaddr_in address;
  socklen_t addrlen = sizeof(address);

  bzero(&address, sizeof(address));

  address.sin_family = AF_INET;
  address.sin_addr.s_addr = htonl(INADDR_ANY);
  address.sin_port = htons(PORT);

  if (bind(server, (struct sockaddr *)&address, addrlen) < 0)
    e("Bind Unsuccessfull");

  if (listen(server, MAX) < 0)
    e("Can't Listen");

  return server;
}
```

```c
void r_and_w(int client)
{

  int n;
  char buffer[255] = "Hello";
  char buffer1[255] = {0};

  time_t timec;
  struct tm *lc_time;

  timec = time(NULL);
  lc_time = localtime(&timec);

  strcpy(buffer, asctime(lc_time));

  send(client, buffer, 255, 0);
  printf("Response Sent!");
}

int main()
{
  int port = 2020;
  int server = tcp_server(port);

  struct sockaddr_in cliaddress;
  socklen_t addrlen = sizeof(cliaddress);
  bzero(&cliaddress, addrlen);
  printf("Server Listening\n");

  int client = accept(server, (struct sockaddr *)&cliaddress, &addrlen);
  if (client < 0)
    e("Can't Accept Connections");

  printf("\nClient Connected from IP: %s and PORT: %d\n", inet_ntoa(cliaddress.sin_addr), ntohs(cliaddress.sin_port));

  r_and_w(client);

  close(client);
  close(server);
  return 0;
}
```

Client:

```c
#include <stdio.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <arpa/inet.h>

void e(char *msg)
{
  perror(msg);
  exit(1);
}

int tcp_client(const char *addr, int port)
{

  int client;
  client = socket(AF_INET, SOCK_STREAM, 0);
  if (client < 0)
    e("Socket creation error");

  //Optional Bind
  struct sockaddr_in address;
  socklen_t addrlen = sizeof(address);
  bzero(&address, addrlen);

  address.sin_family = AF_INET;
  address.sin_addr.s_addr = htonl(INADDR_ANY);
  address.sin_port = htons(18523);

  int opt = 1;
  if (setsockopt(client, SOL_SOCKET, SO_REUSEADDR, &opt, sizeof(opt)) < 0)
    e("Options error");

  if (bind(client, (struct sockaddr *)&address, addrlen))
    e("Bind Error");

  if (inet_pton(AF_INET, addr, &address.sin_addr) <= 0)
    e("Address error");

  address.sin_port = htons(port);

  if (connect(client, (struct sockaddr *)&address, addrlen))
    e("Connection error");

  return client;
}
```

```c
void r_and_w(int client)
{
  int n;
  char buffer[255] = {0};

  printf("\nRequest Sent");

  recv(client, &buffer, 255, 0);
  printf("\nFrom Server : %s", buffer);
}

int main()
{

  int client = tcp_client("127.0.0.1", 2020);
  printf("Connected");
  r_and_w(client);
  close(client);

}
```

Terminal:

```
saisrini@saisrinis-MacBook-Air Date_time % cd "/Users/saisrini/IdeaProjects/lab1
/Date_time/" && gcc server.c -o server && "/Users/saisrini/IdeaProjects/lab1/Dat
e_time/"server
Server Listening

Client Connected from IP: 127.0.0.1 and PORT: 18523
Response Sent!
```

```
saisrini@saisrinis-MacBook-Air Date_time % cd "/Users/saisrini/IdeaProjects/lab1
/Date_time/" && gcc client.c -o client && "/Users/saisrini/IdeaProjects/lab1/Dat
e_time/"client
Connected
Request Sent
From Server : Sat Oct  9 21:40:01 2021
saisrini@saisrinis-MacBook-Air Date_time %
```

3.Arthimetic Operations

Server:

```c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <string.h>
#include <netinet/in.h>
#include <time.h>

#define TRUE 1
#define FALSE 0
#define MAX 4

void e(char *msg)
{
  perror(msg);
  exit(1);
}

int tcp_server(int PORT)
{

  int server;
  server = socket(AF_INET, SOCK_STREAM, 0);
  if (server < 0)
    e("Socket Creation error");

  int opt = TRUE;
  if (setsockopt(server, SOL_SOCKET, SO_REUSEADDR, &opt, (socklen_t)sizeof(opt)) < 0)
    e("Option Set Unsuccessful");

  struct sockaddr_in address;
  socklen_t addrlen = sizeof(address);

  bzero(&address, sizeof(address));

  address.sin_family = AF_INET;
  address.sin_addr.s_addr = htonl(INADDR_ANY);
  address.sin_port = htons(PORT);

  if (bind(server, (struct sockaddr *)&address, addrlen) < 0)
    e("Bind Unsuccessfull");

  if (listen(server, MAX) < 0)
    e("Can't Listen");

  return server;
}
```

```c
void r_and_w(int client)
{

  int n, choice, num1, num2, result;
  write(client, "Enter required operation:\n1.Addition\n2.Subtraction\n3.Division\n4.Multiplication\n5.Exit",
  Multiplication\n5.Exit"));
  read(client, &choice, sizeof(int));

  printf("Client- choice is: %d\n", choice);
  n = write(client, "Enter number 1: ", strlen("Enter number 1: "));

  read(client, &num1, sizeof(num1));
  printf("Client- Number 1 is %d\n", num1);
  n = write(client, "Enter number 2: ", strlen("Enter number 2: "));

  read(client, &num2, sizeof(num2));
  printf("Client- Number 2 is %d\n", num2);

  switch (choice)
  {
  case 1:
    result = num1 + num2;
    printf("Result is %d + %d = %d\n", num1, num2, result);
    break;
  case 2:
    result = num1 - num2;
    printf("Result is %d - %d = %d\n", num1, num2, result);
    break;
  case 3:
    result = num1 / num2;
    printf("Result is %d / %d = %d\n", num1, num2, result);
    break;
  case 4:
    result = num1 * num2;
    printf("Result is %d * %d = %d\n", num1, num2, result);
    break;
  case 5:
    break;
  }
  write(client, &result, sizeof(result));
}
```

```c
int main()
{
  int port = 2020;
  int server = tcp_server(port);

  struct sockaddr_in cliaddress;
  socklen_t addrlen = sizeof(cliaddress);
  bzero(&cliaddress, addrlen);
  printf("Server Listening\n");

  int client = accept(server, (struct sockaddr *)&cliaddress, &addrlen);
  if (client < 0)
    e("Can't Accept Connections");

  printf("\nClient Connected from IP: %s and PORT: %d\n", inet_ntoa(cliaddress.sin_addr), ntohs(cliaddress.sin_port));

  r_and_w(client);

  close(client);
  close(server);
  return 0;
}
```

Client:

```c
#include <stdio.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <arpa/inet.h>

void e(char *msg)
{
  perror(msg);
  exit(1);
}

int tcp_client(const char *addr, int port)
{

  int client;
  client = socket(AF_INET, SOCK_STREAM, 0);
  if (client < 0)
    e("Socket creation error");

  //Optional Bind
  struct sockaddr_in address;
  socklen_t addrlen = sizeof(address);
  bzero(&address, addrlen);

  address.sin_family = AF_INET;
  address.sin_addr.s_addr = htonl(INADDR_ANY);
  address.sin_port = htons(18523);

  int opt = 1;
  if (setsockopt(client, SOL_SOCKET, SO_REUSEADDR, &opt, sizeof(opt)) < 0)
    e("Options error");

  if (bind(client, (struct sockaddr *)&address, addrlen))
    e("Bind Error");

  if (inet_pton(AF_INET, addr, &address.sin_addr) <= 0)
    e("Address error");

  address.sin_port = htons(port);

  if (connect(client, (struct sockaddr *)&address, addrlen))
    e("Connection error");

  return client;
}
```

```c
void r_and_w(int client)
{
  int n, choice, num1, num2, result;
  char buffer[256] = {0};

  read(client, buffer, 256);
  printf("Server- %s\n", buffer);
  scanf("%d", &choice);

  write(client, &choice, sizeof(int));
  bzero(buffer, 256);

  read(client, buffer, 256);
  printf("Server- %s\n", buffer);
  scanf("%d", &num1);

  write(client, &num1, sizeof(int));
  bzero(buffer, 256);

  n = read(client, buffer, 256);
  printf("Server- %s\n", buffer);
  scanf("%d", &num2);
  write(client, &num2, sizeof(int));

  read(client, &result, sizeof(result));
  printf("Server- The answer is %d\n", result);
}

int main()
{
  int client = tcp_client("127.0.0.1", 2020);
  printf("Connected");
  r_and_w(client);
  close(client);
}
```

Terminal:

```
cd "/Users/saisrini/IdeaProjects/lab1/" && gcc ds.c -o ds
 && "/Users/saisrini/IdeaProjects/lab1/"ds
saisrini@saisrinis-MacBook-Air lab1 % cd "/Users/saisrini
/IdeaProjects/lab1/" && gcc ds.c -o ds && "/Users/saisrin
i/IdeaProjects/lab1/"ds
SOCKET CREATED SUCCESSFULLY
BIND SUCCESSFUL
LISTEN SUCESSFUL
ACCEPT SUCCESSFUL
Client- choice is: 3
Client- Number 1 is 10
Client- Number 2 is 2
Result is 10 / 2 = 5
Client- choice is: 1
Client- Number 1 is 34
Client- Number 2 is 10
Result is 34 + 10 = 44
Client- choice is: 1
Client- Number 1 is 34
Client- Number 2 is 10
Result is 34 + 10 = 44
Client- choice is: 1
Client- Number 1 is 34
Client- Number 2 is 10
Result is 34 + 10 = 44
Client- choice is: 1
Client- Number 1 is 34
Client- Number 2 is 10
Result is 34 + 10 = 44
Client- choice is: 1
saisrini@saisrinis-MacBook-Air lab1 %
```

```
saisrini@saisrinis-MacBook-Air lab1 % cd "/Users/saisrini/
IdeaProjects/lab1/" && gcc dc.c -o dc && "/Users/saisrini/
IdeaProjects/lab1/"dc
SOCKET CREATED SUCCESSFULLY
CONNECT SUCCESSFUL
Server- Enter required operation:
1.Addition
2.Subtraction
3.Division
4.Multiplication
5.Exit
3
Server- Enter number 1:
10
Server- Enter number 2:
2
Server- The answer is 5
Server- Enter required operation:
1.Addition
2.Subtraction
3.Division
4.Multiplication
5.Exit
1
Server- Enter number 1:
34
Server- Enter number 2:
10
Server- The answer is 44
Server- Enter required operation:
1.Addition
2.Subtraction
3.Division
4.Multiplication
5.Exit
5
You have selected Exit..
saisrini@saisrinis-MacBook-Air lab1 %
```