• Makefile:

    finish: pa6_server.o pa6_client.o
    
    g++ -o pa6_server pa6_server.o
    g++ -o pa6_client pa6_client.o
    
    pa6_server.o: pa6_server.c
    g++ -c pa6_server.c
    
    pa6_client.o: pa6_client.c
    g++ -c pa6_client.c

• Server Code:

    #include <sys/types.h>
    #include <sys/socket.h>
    #include <stdio.h>
    #include <netinet/in.h>
    #include <sys/time.h>
    #include <sys/ioctl.h>
    #include <unistd.h>
    
    int main()
    {

        int server_sockfd, client_sockfd;
        int server_len, client_len;
        struct sockaddr_in server_address;
        struct sockaddr_in client_address;
        int result;
        int game = 1;
        fd_set readfds, testfds;
        char ch;
        char outstring[4]={' '};
        char outstring1[4] = {' '};
        char instring[5], instring1[5];
        int nread;
        int player1, player2;

        server_sockfd = socket(AF_INET, SOCK_STREAM, 0);
        
        server_address.sin_family = AF_INET;
        server_address.sin_addr.s_addr = htonl(INADDR_ANY);
        server_address.sin_port = htons(4546);
server_len = sizeof(server_address);

if( bind(server_sockfd, (struct sockaddr *)&server_address, server_len)==-1)
{
    fprintf(stderr," bind failed \n");
    return 1;
}

listen(server_sockfd, 2);

FD_ZERO(&readfds);
FD_SET(server_sockfd, &readfds);

printf("Referee Waiting For Players\n");

player1 = accept(server_sockfd, NULL, NULL);
player2 = accept(server_sockfd, NULL, NULL);

read(player1, instring, 5);
read(player2, instring1, 5);

write(player1, "GO", 2);
write(player2, "GO", 2);

while(game)
{
    read(player1, instring, 4);
    read(player2, instring1, 4);
    if(instring[0] == 'S' || instring1[0] == 'S')
    {
        write(player1, "STOP", 4);
        write(player2, "STOP", 4);
        break;
    }
    outstring[0] = instring1[0];
    outstring1[0] = instring[0];
    if(instring[0] == instring1[0])
    {
        outstring[1] = '2';
        outstring1[1] = '2';
        write(player1, outstring, 4);
        write(player2, outstring1, 4);
    }
    else if(instring[0] == '1' && instring1[0] == '2')
    {
        outstring[1] = '1';
    }
}
outstring1[1] = '0';
write(player1, outstring,4);
write(player2,outstring1,4);
}
else if(instring[0] == '1' && instring1[0] == '3')
{
    outstring[1] = '0';
    outstring1[1] = '1';
    write(player1, outstring,4);
    write(player2,outstring1,4);
}
else if(instring[0] == '2' && instring1[0] == '1')
{
    outstring[1] = '0';
    outstring1[1] = '1';
    write(player1, outstring,4);
    write(player2,outstring1,4);
}
else if(instring[0] == '2' && instring1[0] == '3')
{
    outstring[1] = '1';
    outstring1[1] = '0';
    write(player1, outstring,4);
    write(player2,outstring1,4);
}
else if(instring[0] == '3' && instring1[0] == '2')
{
    outstring[1] = '0';
    outstring1[1] = '1';
    write(player1, outstring,4);
    write(player2,outstring1,4);
}
else
{
    outstring[1] = '1';
    outstring1[1] = '0';
    write(player1, outstring,4);
    write(player2,outstring1,4);
}

close(server_sockfd);
return 0;
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netdb.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <iostream>
using namespace std;

#define PORT 4546

int main()
{
    struct hostent *server_crap;
    int sockfd;
    int len;
    int myscore = 0, otherscore = 0, rounds = 0;
    struct sockaddr_in address;
    int result;
    char choice[4]={' '};
    char ch = 'A';
    int game = 1;
    char instring[5];
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    address.sin_family = AF_INET;
    if( (server_crap = gethostbyname("csserver")) ==NULL)
    {
        fprintf(stderr,"you suck ass\n");
        return 1;
    }
    address.sin_addr = *(struct in_addr *)*server_crap->h_addr_list;
    address.sin_port = htons(PORT);
    len = sizeof(address);
    result = connect(sockfd, (struct sockaddr *)&address, len);
    if(result == -1)
    {
        perror("oops: client1");
        return 1;
    }
}
write(sockfd, "READY", 5);
printf("Waiting for game to start!\n");
while(strcmp("GO", instring)! = 0)
{
    read(sockfd, instring, 2);
}

while(game)
{
    printf("\nEnter your choice:\n");
    printf(" 0=Exit  1=Rock  2=Paper  3=Scissors\n");
    cin >> ch;
    if(ch != '0')
    {
        choice[0] = ch;
        write(sockfd, choice, 4);
    }
    else if(ch == '0')
    {
        write(sockfd, "STOP", 4);
        break;
    }
    read(sockfd, instring, 4);
    if(instring[0] == 'S')
    {
        break;
    }
    printf("\n");
    switch(ch)
    {
        case '1':
            printf("You chose Rock!\n");
            break;
        case '2':
            printf("You chose Paper!\n");
            break;
        case '3':
            printf("You chose Scissors!\n");
            break;
    }
switch(instring[0])
{
    case '0':
        game = 0;
        printf("Other player quit!
");
        break;
    case '1':
        printf("Other player chose Rock!
");
        break;
    case '2':
        printf("Other player chose Paper!
");
        break;
    case '3':
        printf("Other player chose Scissors!
");
        break;
}

switch(instring[1])
{
    case '0':
        printf("You win!
");
        myscore++;
        break;
    case '1':
        printf("You lose!
");
        otherscore++;
        break;
    case '2':
        printf("You tied!
");
        break;
}
    rounds++;
}

printf("Game Over!
");
printf("There were %d rounds completed.
", rounds);
if(myscore > otherscore)
{
    printf("You won %d to %d
", myscore, otherscore);
}
else if(myscore < otherscore)
{
    printf("You lost %d to %d
", otherscore, myscore);
}
else
```c
}  
    printf("You tied %d to %d. Now you must wrestle to decide winner!\n", myscore, otherscore);
    
}  
    close(sockfd);
    return 0;
};
```

- **Server Output:**

  [mr56@csserver matt_6]$ ./pa6_server  
  Referee Waiting For Players

- **Player 1 Output:**

  [mr56@hmrserver matt_6]# ./pa6_client  
  Waiting for game to start!  
  
  Enter your choice:  
  0=Exit  1=Rock  2=Paper  3=Scissors  
  1  
  
  You chose Rock!  
  Other player chose Paper!  
  
  You lose!  
  
  Enter your choice:  
  0=Exit  1=Rock  2=Paper  3=Scissors  
  3  
  
  You chose Scissors!  
  Other player chose Rock!  
  
  You lose!  
  
  Enter your choice:  
  0=Exit  1=Rock  2=Paper  3=Scissors  
  2  
  
  You chose Paper!  
  Other player chose Scissors!  
  
  You lose!
Enter your choice:
0=Exit 1=Rock 2=Paper 3=Scissors
0

Game Over!
There were 3 rounds completed.
You lost 3 to 0

- Player 2 Output:

  [mr56@hmrsrvr matt_6]# ./pa6_client
  Waiting for game to start!

  Enter your choice:
  0=Exit 1=Rock 2=Paper 3=Scissors
  2

  You chose Paper!
  Other player chose Rock!

  You win!

  Enter your choice:
  0=Exit 1=Rock 2=Paper 3=Scissors
  1

  You chose Rock!
  Other player chose Scissors!

  You win!

  Enter your choice:
  0=Exit 1=Rock 2=Paper 3=Scissors
  3

  You chose Scissors!
  Other player chose Paper!

  You win!

  Enter your choice:
  0=Exit 1=Rock 2=Paper 3=Scissors
  0

  Game Over!
  There were 3 rounds completed.
You won 3 to 0