Distributed Systems And Computing (ITD01)

Internal Lab Assessment

Submitted by :
Shiv Kumar
2016UIT2563
IT-2
8th Semester

```c
#include <stdio.h>
#include <winsock2.h>

void main()
{

  WSADATA wsaData;
  SOCKET ListenSocket;
  sockaddr_in service;


  int iResult = WSAStartup(MAKEWORD(2, 2), &wsaData);
  if(iResult != NO_ERROR)
    printf("Error at WSAStartup().\n");
  else
    printf("WSAStartup() is OK.\n");


  ListenSocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
  if (ListenSocket == INVALID_SOCKET)
  {
    printf("Error at socket().\n");
    WSACleanup();
    return;
  }
  else
      printf("socket() is OK.\n");


  hostent* thisHost;
  char* ip;
  u_short port;
  port = 55555;
  thisHost = gethostbyname("");
  ip = inet_ntoa(*(struct in_addr *)*thisHost->h_addr_list);


  printf("\nIP address is: %s.\n", ip);
  printf("Hostname is: %s.\n", thisHost->h_name);
  printf("Address type: %i.\n\n", thisHost->h_addrtype);

  service.sin_family = AF_INET;
  service.sin_addr.s_addr = inet_addr(ip);
  service.sin_port = htons(port);

  if (bind(ListenSocket,(SOCKADDR*) &service, sizeof(service)) == SOCKET_ERROR)
  {
    printf("bind() failed lol!\n");
    closesocket(ListenSocket);
    return;
  }
else
      printf("bind() is OK.\n");

  BOOL bOptVal = TRUE;
  int bOptLen = sizeof(BOOL);
  int iOptVal;
```

```c
    int iOptLen = sizeof(int);

    if (getsockopt(ListenSocket, SOL_SOCKET, SO_KEEPALIVE, (char*)&iOptVal, &iOptLen) !=
SOCKET_ERROR)
    {
        printf("SO_KEEPALIVE value: %ld.\n", iOptVal);
    }

    if (setsockopt(ListenSocket, SOL_SOCKET, SO_KEEPALIVE, (char*)&bOptVal, bOptLen) !=
SOCKET_ERROR)
    {
        printf("Sends keep-alives. Setting the SO_KEEPALIVE.\n");
    }

    if (getsockopt(ListenSocket, SOL_SOCKET, SO_KEEPALIVE, (char*)&iOptVal, &iOptLen) !=
SOCKET_ERROR)
    {
        printf("SO_KEEPALIVE value: %ld.\n", iOptVal);
    }
    // Binding to the already used socket
    BOOL bOptVal1 = TRUE;
    int bOptLen1 = sizeof(BOOL);
    int iOptVal1;
    int iOptLen1 = sizeof(int);

    if (setsockopt(ListenSocket, SOL_SOCKET, SO_REUSEADDR, (char*)&bOptVal1, bOptLen1) !=
SOCKET_ERROR)
    {
        printf("Socket can be bound to an address that is already in use.\n");
    }

    if (getsockopt(ListenSocket, SOL_SOCKET, SO_REUSEADDR, (char*)&iOptVal1, &iOptLen1) !=
SOCKET_ERROR)
    {
        printf("SO_REUSEADDR value: %ld.\n", iOptVal1);
    }

    WSACleanup();
    return;
}
```