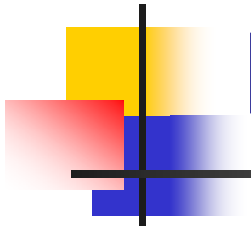
TCP Port Scan

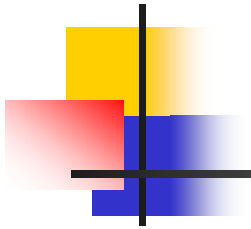
Course Name: Principle of Computer network experiment

Teacher: Li Huixian



Experiment 2

- **Subject:** Discover the TCP service opened by the remote host.
- **Purpose:** Through discovering the TCP service opened by the remote host, understand the working principle of C/S communication mode and port scanning.
- **Requirement:**
  - (1) Run as command: DOS>scanPort remote_ip.
  - (2) Output : TCP service port number opened by the remote host.

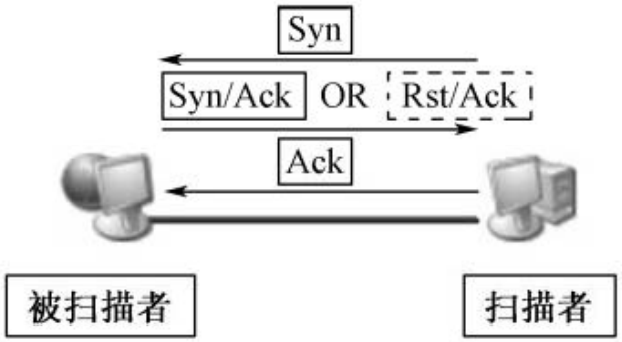


Experiment 2

- **Implementation principle**
  - **Common methods**: TCP connect, TCP SYN, TCP FIN;
  - **Method used in this experiment:** The OS provides the connect() system call to establish a connection with a port of the remote host. If the port of the remote host is in listening, the connect() connection is successful; otherwise, the port is closed;
  - **Advantage:** Do permission required. Any user can use the system call.

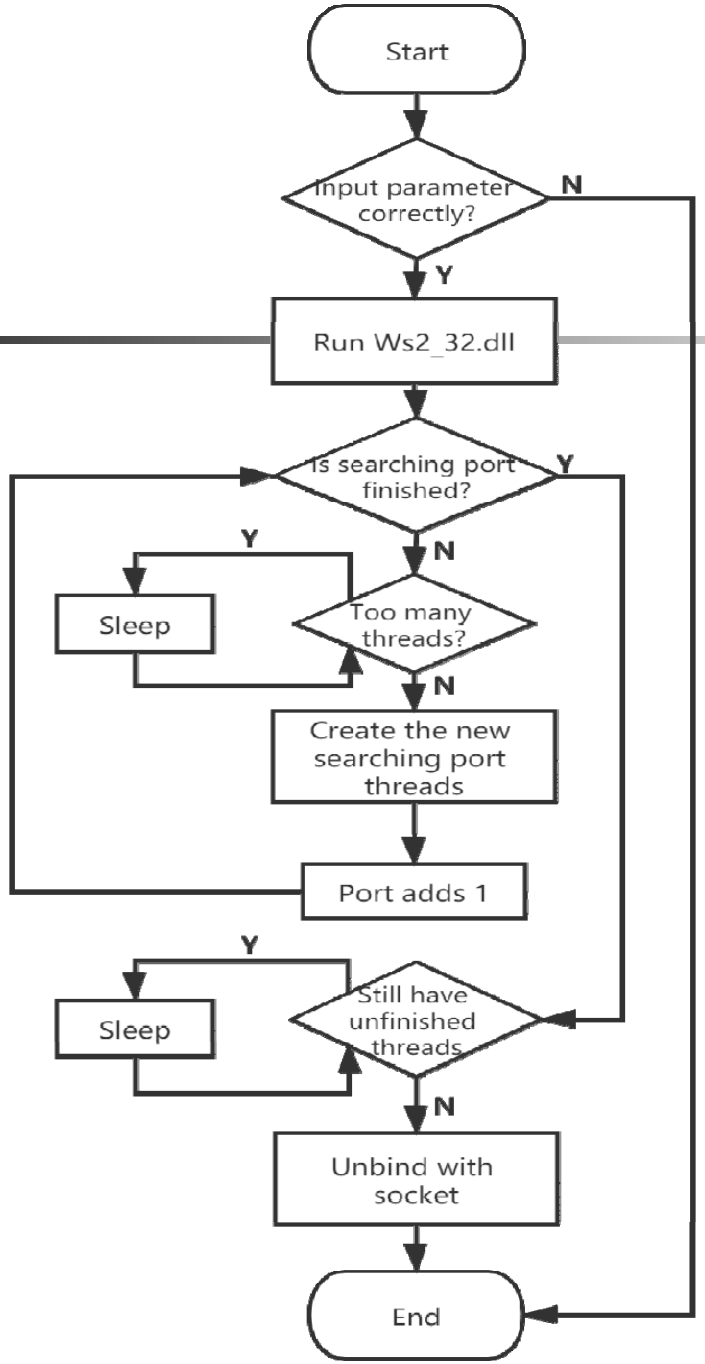
TCP Connect() Scan

- This is the most basic TCP scan, which use the connect() function provided by the system to connect to the target port and try to establish a complete three-way handshake process with the remote host. Therefore, this scanning method is also called "full scan" . If the target port is in the listening state, connect() returns successfully, otherwise -1 is returned, indicating that the port is not accessible.

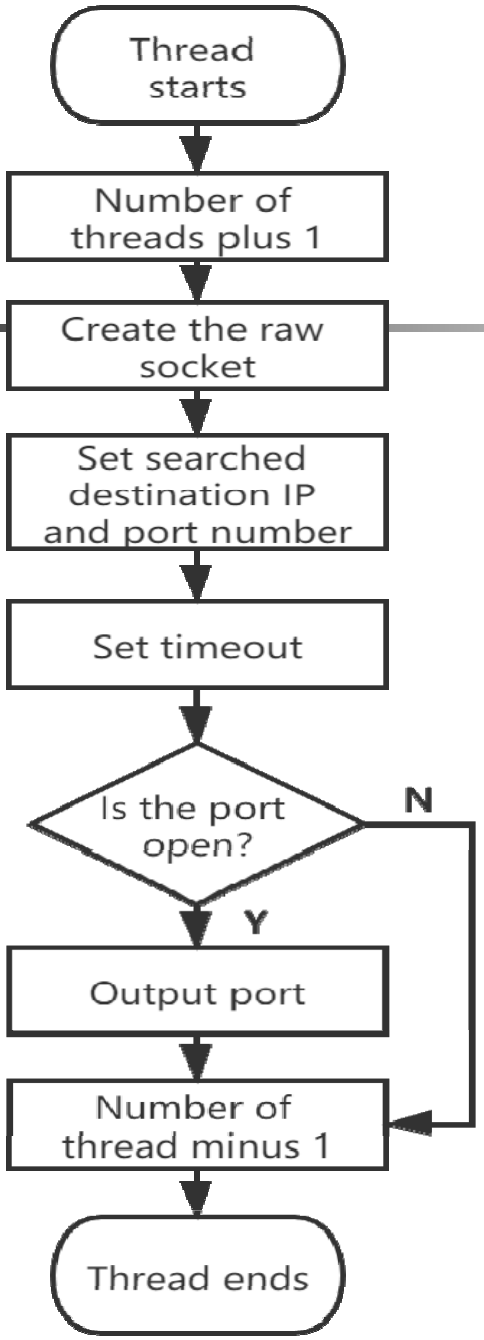


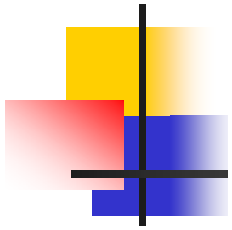
TCPconnect（）扫描

Main program flowchart



Sub-thread flowchart

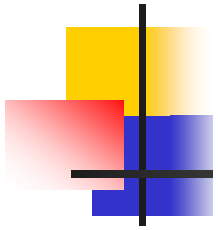




```
#include "stdafx.h"
#include<iostream>
using namespace std;
#include<winSock2.h>
#pragma comment (lib,"ws2_32.lib")
#define STAUS_FALIED 0xFFFF      //Error code
unsigned long server IP;        //Scanned server address
long MaxThread = 200;          //Number of scan threads allowed
long ThreadCount = 0;         //Number of threads being scanned
long *aa = &ThreadCount;

//Thread scanning port
DWORD WINAPI ScanPort(LPVOID lpParam) {
    short Port = *(short*)lpParam;
    InterlockedIncrement(aa);
    //Create fluid socket words
    SOCKET sock = socket(AF_INET, SOCK_STREAM, 0);
    if (sock == INVALID_SOCKET)
    {
        cout << "Failed to create socket words" << endl;
        return 0;
    }
    else
    {
        //Fill in the server address
        sockaddr_in severAddr;
        severAddr.sin_family = AF_INET;
        severAddr.sin_port = htons(Port);
        severAddr.sin_addr.S_un.S_addr = serverIP;
        //Determine whether this machine is turned on
        connect(sock, (sockaddr*)&severAddr, sizeof(severAddr));
        struct fd_set write;    //Write socket words set
        FD_ZERO(&write);
        FD_SET(sock, &write);
        //Initialization timeout
        struct timeval timeout;
        timeout.tv_sec = 100 / 1000;
        timeout.tv_usec = 0;
        if (select(0,NULL,&write,NULL,&timeout)>0)//Set timeout
        {
            cout << Port << ",";
        };
        closesocket(sock);
    }
    InterlockedDecrement(aa);
    return 0;
}
```

Sample code for scanning port thread



Sample code for main program

```
48  void main(int argc, char *argc[])
49  {
50      if (argc != 2)
51      {
52          cout << "Please enter the destination host IP address" << endl;
53      }
54      //Establish a binding with the socket library
55      WSADATA WSAData;
56      if (WSAStartup(MAKEWORD(2, 2), &WSADATA) != 0)
57      {
58          count << "WSAStartup falied!" << GetLastError() << endl;
59          ExitProcess(STATUS_FALIED);
60      }
61      severIP = inet_addr(argv[1]);
62      count << "The following ports are open:" endl;
63      for (int i = 1; i < 1024; i++) {     //Scan server 1-1024 port
64          //Exceed the maximum allowable thread waiting
65          while (ThreadCount >= MaxThread)
66              Sleep(10);
67          //Create thread, scan port
68          DWORD ThreadID;
69          CreateThread(NULL, 0, ScanPort, (LPVoID)new short(i), 0, &ThreadID);
70      }
71      //There are unfinished threads, waiting
72      while (ThreadCount > 0)
73          sleep(50);
74      //Unbind with Socket library
75      WSACleanup();
76
77  }
```

Result for Sample code