TP 5 : Création d'un Serveur de Communication via un Protocole TCP (Compléments)

Voici quelques fonctions présentes dans WinSock.
Pour plus de détails, il faut aller à l'adresse :
https://msdn.microsoft.com/en-us/library/windows/desktop/ms741394(v=vs.85).aspx

The **accept** function permits an incoming connection attempt on a socket.

Syntaxe :
```
SOCKET accept(
        _In_ SOCKET s,
        _Out_ struct sockaddr *addr,
        _Inout_ int *addrlen,
);
```

The **bind** function associates a local address with a socket.

Syntaxe :
```
int bind(
        _In_ SOCKET s,
        _In_ const struct sockaddr *name,
        _In_ int namelen
);
```

The **closesocket** function closes an existing socket.

Syntaxe :
```
int closesocket(
        _In_ SOCKET s
);
```

The **connect** function establishes a connection to a specified socket.

Syntaxe :
```
int connect(
        _In_ SOCKET s,
        _In_ const struct sockaddr *name,
        _In_ int namelen
);
```

The **listen** function places a socket in a state in which it is listening for an incoming connection.

Syntaxe :
```
int listen(
        _In_ SOCKET s,
        _In_ int backlog
);
```

The **recv** function receives data from a connected socket or a bound connectionless socket. file in the process.

Syntaxe :
```
int recv(
        _In_ SOCKET s,
        _Out_ char *buf,
        _In_ int len,
        _In_ int flags
);
```

The **send** function sends data on a connected socket.

Syntaxe :
```
int send(
        _In_ SOCKET s,
        _In_ const char *buf,
        _In_ int len,
        _In_ int flags
);
```

The **socket** function creates a socket that is bound to a specific transport service provider.

Syntaxe :
```
SOCKET WSAAPI socket(
        _In_ int af,
        _In_ int type,
        _In_ int protocol
);
```

The **WSAAccept** conditionally accepts a connection based on the return value of a condition function, provides quality of service flow specifications, and allows the transfer of connection data.

Syntaxe :
```
SOCKET WSAAccept(
        _In_ SOCKET s,
        _Out_ struct sockaddr *addr,
        _InOut_ LPINT addrlen,
        _In_ LPCONDITIONPROC lpfnCondition,
        _In_ DWORD_PTR dwCallbackData
);
```

The **WSACleanup** function terminates use of the Winsock 2 DLL (Ws2_32.dll).

Syntaxe :
```
int WSACleanup(void);
```

The **WSACloseEvent** function closes an open event object handle.

Syntaxe :
```
BOOL WSACloseEvent();
```

The **WSAConnect** function establishes a connection to another socket application, exchanges connect data, and specifies required quality of service based on the specified FLOWSPEC structure.

Syntaxe :

```
int WSAConnect(
        _In_ SOCKET s,
        _In_ const struct sockaddr *name,
        _In_ int namelen,
        _In_ LPWSABUF lpCallerData,
        _Out_ LPWSABUF lpCalleeData,
        _In_ LPQOS lpSQOS,
        _In_ LPQOS lpGQOS
);
```

The **WSACreateEvent** function creates a new event object.

Syntaxe :

```
WSAEVENT WSACreateEvent(void);
```

The **WSAGetLastError** function returns the error status for the last Windows Sockets operation that failed.

Syntaxe :

```
int WSAGetLastError(void);
```

The **WSARecv** function receives data from a connected socket or a bound connectionless socket.

Syntaxe :

```
int WSARecv(
        _In_ SOCKET s,
        _Inout_ LPWSABUF lpBuffers,
        _In_ DWORD dwBufferCount,
        _Out_ LPDWORD lpNumberOfBytesRecvd,
        _Inout_ LPDWORD lpFlags,
        _In_ LPWSAOVERLAPPED lpOverlapped,
        _In_ LPWSAOVERLAPPED_COMPLETION_ROUTINE lpCompletionRoutine
);
```

The **WSASend** function sends data on a connected socket.

Syntaxe :

```
int WSASend(
        _In_ SOCKET s,
        _In_ LPWSABUF lpBuffers,
        _In_ DWORD dwBufferCount,
        _Out_ LPDWORD lpNumberOfBytesSent,
        _In_ DWORD dwFlags,
        _In_ LPWSAOVERLAPPED lpOverlapped,
        _In_ LPWSAOVERLAPPED_COMPLETION_ROUTINE lpCompletionRoutine
);
```

The **WSASocket** function creates a socket that is bound to a specific transport-service provider.

Syntaxe :

```
SOCKET WSASocket(
        _In_ int af,
        _In_ int type,
        _In_ int protocol,
        _In_ LPWSAPROTOCOL_INFO lpProtocolInfo,
        _In_ GROUP g,
        _In_ DWORD dwFlags
);
```

The **WSAStartup** function initiates use of the Winsock DLL by a process.

Syntaxe :

```
int WSAStartup(
        _In_ WORD wVersionRequested,
        _Out_ LPWSADATA lpWSAData
);
```