

บทที่ 2

ทฤษฎีที่เกี่ยวข้องกับงานวิจัย

วินโดว์ซ็อกเก็ต หรือ วินซ็อก

ในการเขียนโปรแกรมทางด้านเครือข่ายบนระบบปฏิบัติการไมโครซอฟต์วินโดวส์ จำเป็นที่จะต้องเขียนโปรแกรมที่เรียกผ่านตัวประสานสำหรับติดต่อกับระบบเครือข่าย ซึ่งได้กำหนดพื้นฐานเป็นแนวทางที่เรียกทั่วไปว่า ซ็อกเก็ต (Socket) โดย ซ็อกเก็ต ที่ใช้สำหรับการเขียนโปรแกรมบนวินโดวส์ จะเรียกว่า วินโดว์ ซ็อกเก็ต (Window socket) หรือที่เรียกสั้นๆว่า วินซ็อก (Winsock) จะหมายถึง ข้อกำหนดในการติดต่อกับการเขียนโปรแกรมทางด้านเครือข่ายบนระบบปฏิบัติการไมโครซอฟต์วินโดวส์ ซึ่งโดยมากการเขียนโปรแกรมทางด้านเครือข่าย จะอยู่บนพื้นฐานของซ็อกเก็ต ของ บีเอสดี (BSD) ซึ่งย่อมาจาก Berkeley Software Distribution จากมหาวิทยาลัยเบิร์กลีย์ ซึ่งได้รวมเอาส่วนสำคัญของการเขียน ชุดคำสั่งในแบบซ็อกเก็ต จากเบิร์กลีย์ และส่วนขยายเพิ่มเติมของข้อกำหนดของวินโดวส์ ที่ออกแบบมาเพื่อให้ผู้พัฒนาโปรแกรมสามารถที่จะใช้ประโยชน์ของธรรมชาติในการขับเคลื่อนเมสเสจ (Windows Messages) ของวินโดวส์ได้

โปรแกรมเครือข่ายที่เป็นไปตามข้อกำหนดของวินโดว์ซ็อกเก็ต และโปรแกรมที่สามารถทำงานร่วมกับตัวประสานวินโดว์ซ็อกเก็ต จะเรียกว่า โปรแกรมประยุกต์วินโดว์ซ็อกเก็ต

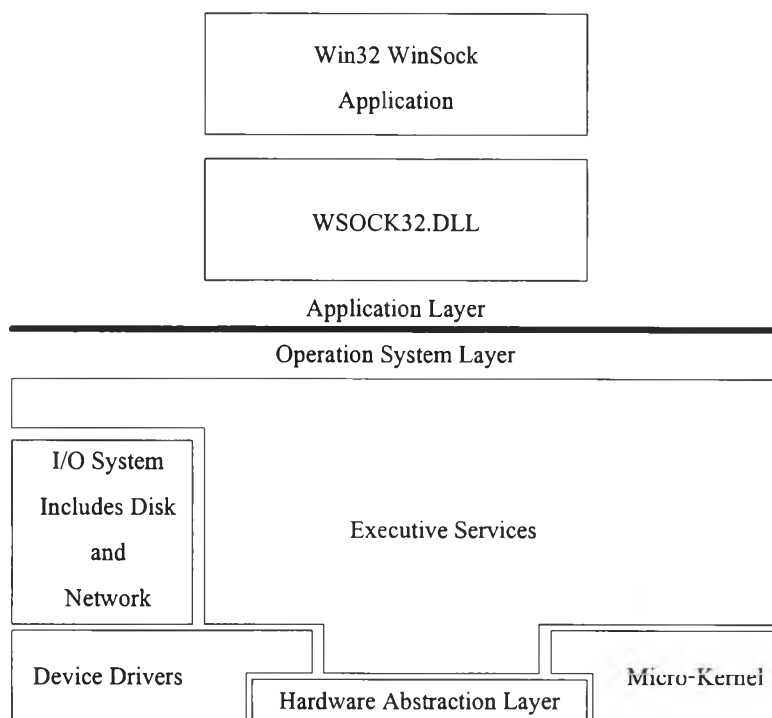
1. เบิร์กลีย์ ซ็อกเก็ต

ข้อกำหนดของวินโดว์ซ็อกเก็ต ถูกพัฒนาขึ้นมาจากแบบการเขียนโปรแกรมของทางเบิร์กลีย์ ซึ่งใช้เป็นมาตรฐานที่เป็นเครือข่ายแบบทีซีพี/ไอพี (TCP/IP) และบางส่วนได้ถูกออกแบบมาจากมหาวิทยาลัยแคลิฟอร์เนีย

2. ส่วนขยายเพิ่มเติมของข้อกำหนดของ วินซ็อก

ตัวประสานระหว่างโปรแกรมประยุกต์ของข้อกำหนดของวินซ็อก นั้นมีการทำงานที่เป็นแบบทั้ง 16 บิต และ 32 บิต ซึ่งส่วนขยายเพิ่มเติมของข้อกำหนดของวินโดวส์ที่ถูกรวบรวมไว้ในวินซ็อก จะช่วยให้ผู้พัฒนาโปรแกรมประยุกต์สามารถสร้างซอฟต์แวร์ที่อยู่ในรูปแบบการเขียนโปรแกรมบนวินโดวส์ได้ซึ่งจะทำให้โปรแกรมมีประสิทธิภาพสูงสุด และการทำงานร่วมกันแบบหลายงานได้ด้วย ผู้พัฒนาโปรแกรมสามารถเขียนโปรแกรมให้สามารถใช้งานตัวประสานวินซ็อกได้ โดยการลิงก์กับคำสั่งนำเข้าไปในแฟ้มข้อมูล winsock.lib สำหรับโปรแกรมแบบ 16 บิต ถ้าเป็นโปรแกรมแบบ 32 บิต จะใช้แฟ้มข้อมูล wsock32.lib ในขณะที่วินซ็อกทำงานจะไปเรียกใช้คำสั่งที่บรรจุอยู่ในแฟ้มข้อมูล wsock32.dll โดยที่ wsock32.dll กับโปรแกรมประยุกต์วินโดว์ซ็อกเก็ตและ

ระบบปฏิบัติการวินโดวส์เอ็นที มีแพลตฟอร์มแบบ Win32 มีการทำงานร่วมกันดังรูปที่ 2.1 (Dumas, 1995)



รูปที่ 2.1 แสดงแผนภาพการทำงานร่วมกันของ wsock32.dll โปรแกรมประยุกต์วินโดวส์ซ็อกเก็ต และระบบปฏิบัติการวินโดวส์เอ็นทีที่มีแพลตฟอร์มแบบ Win32

คลาส ซีวินซ็อก

คลาสซีวินซ็อก (Class CWinSock) (Dumas, 1995) นี้มีไว้สำหรับการเริ่มต้นการขอใช้งานวินซ็อกจากวินโดวส์ การยกเลิกวินซ็อก และรับรายละเอียดของวินซ็อกที่กำลังใช้งานอยู่

คลาส ซีวินซ็อก มีการกำหนดรายละเอียดของคลาส ไว้ดังนี้

```

////////////////////////////////////
// CWinsock
//
class CWinSock
{
private:
    WORD m_wVersionRequired;           // WinSock version required by application
    int m_nLastError;                  // last WinSock error
    WSADATA m_wsaData;                // WinSock information
public:
    CWinSock(WORD wVersionRequired = MAKEWORD(1, 1));
    int Startup( );
    int Shutdown( );
    void Information(LPWSADATA pwsaData);
    int LastError( ) { return m_nLastError; }
};

```

ในการทำงานของโปรแกรมที่มีการเรียกใช้วินซ็อก จำเป็นต้องมีการเริ่มต้นขอใช้วินซ็อกก่อน ซึ่งคลาส ซีวินซ็อก นี้มีหน้าที่ดังกล่าว โดยความหมายของฟังก์ชันสมาชิกของคลาสซีวินซ็อกเป็นดังต่อไปนี้

1. ฟังก์ชันสมาชิก CWinSock::CWinSock()

เป็นคอนสตรัคเตอร์ของออบเจกต์แบบ ซีนีวซ็อก ฟังก์ชันจะทำการกำหนดเวอร์ชันที่แอปพลิเคชันต้องการ และกำหนดให้ตัวแปร `m_nLastError` มีค่าเป็นศูนย์ โดยตัวแปร `m_nLastError` คือเลขจำนวนเต็มที่แสดงถึงความผิดพลาดครั้งหลังสุดที่เกิดขึ้นกับซ็อกเก็ตที่ขอมมาจากวิน โดวส์

2. ฟังก์ชันสมาชิก CWinSock::Startup()

วินซ็อกที่ขอมมาจากวิน โดวส์จะสามารถใช้งานจริงได้ ต้องถูกเรียกด้วยฟังก์ชัน `CWinSock::Startp()` นี้เสียก่อน ผู้พัฒนาโปรแกรมสามารถรู้ได้ว่าฟังก์ชันทำงานสำเร็จหรือไม่จากตัวแปร `m_nLastError`

3. ฟังก์ชันสมาชิก CWinSock::Shutdown()

เป็นฟังก์ชันที่ใช้สำหรับการยกเลิกการใช้งานวินซ็อกที่ขอมมาจากวิน โดวส์ ถ้าทำการยกเลิกไม่สำเร็จ ตัวแปร `m_nLastError` จะมีค่าเท่ากับ `CWINSOCK_WINSOCK_ERROR`

4. ฟังก์ชันสมาชิก CWinSock::Information()

เป็นฟังก์ชันที่ใช้รับข้อมูลที่บอกรายละเอียดของวินซ็อกที่ขอมมาจากวิน โดวส์โดยจะบรรจุรายละเอียดต่างๆ ไว้ในโครงสร้างข้อมูลที่มีชื่อว่า `WSADATA`

5. ฟังก์ชันสมาชิก CWinSock::LastError()

เป็นฟังก์ชันสมาชิกของคลาสซีนีวซ็อก ที่ถูกเขียนให้เป็นฟังก์ชันแบบ `in-line` โดยจะคืนค่าเป็นจำนวนเต็มที่แทนความหมายของความผิดพลาดครั้งหลังสุดที่เกิดขึ้นกับออบเจกต์ซีนีวซ็อก

คลาส ซีสตรีมซ็อกเก็ต

คลาส ซีสตรีมซ็อกเก็ต (Class CStreamSocket) (Dumas, 1995) คลาสนี้มีไว้สำหรับการใช้งานซ็อกเก็ตแบบสายธาร (Stream Socket) โดยมีการกำหนดรายละเอียดของคลาสไว้ดังนี้

```

////////////////////////////////////
// CStreamSocket
//
class CStreamSocket : public CWnd
{
private:
    CWnd *m_pParentWnd;           // window to receive event notification
    UINT m_uMsg;                  // message to send to m_pParentWnd on event
    SOCKET m_s;                   // socket handle
    SOCKADDR_IN m_sinLocal;       // name bound to socket m_s
    SOCKADDR_IN m_sinRemote;      // name on other side of m_s
    int m_nLastError;             // last WinSock error
    BOOL m_bServer;              // TRUE if socket m_s is bound to a name
    CPtrList m_listWrite;         // data waiting to be sent
    CPtrList m_listRead;         // data read

public:
    CStreamSocket(CWnd *pParentWnd, UINT uMsg);
    virtual ~CStreamSocket( );
    int CreateSocket(int nLocalPort);
    int CreateSocket(LPSTR pszLocalService = NULL);
    int DestroySocket( );
    int Connect(LPSTR pszRemoteName, int nRemotePort);
    int Connect(LPSTR pszRemoteName, LPSTR pszRemoteService);
    int Connect(LPSOCKADDR_IN psinRemote);

```

```

int Accept(CStreamSocket *pStreamSocket);
int Write(int nLen, LPVOID pData);
LPVOID Read(LPINT pnLen);
int GetPeerName(LPSOCKADDR_IN psinRemote);
int GetLastError() { return m_nLastError; }

private:
void InitVars(BOOL bInitLastError = TRUE);
LONG HandleRead(WPARAM wParam, LPARAM lParam);
LONG HandleWrite(WPARAM wParam, LPARAM lParam);

// message map functions
protected:
//{{AFX_MSG(CStreamSocket)
//}}AFX_MSG
LONG OnWinSockEvent(WPARAM wParam, LPARAM lParam);
DECLARE_MESSAGE_MAP()
};

```

1. ฟังก์ชันสมาชิก CStreamSocket::CStreamSocket()

เป็นคอนสตรัคเตอร์ของออบเจกต์แบบ CStreamSocket โดยออบเจกต์จะสร้างตัวเองเป็นวินโดว์ที่มองไม่เห็น การสร้างออบเจกต์แบบ CStreamSocket ต้องกำหนดวินโดว์ที่จะทำหน้าที่เป็นวินโดว์แม่ (parent window) ด้วย นอกจากนั้นต้องกำหนดเมสเสจใหม่ขึ้นมาเพื่อใช้เป็นตัวกลางในการสื่อสารกันระหว่างวินโดว์แม่ กับ CStreamSocket วินโดว์ ท้ายที่สุดก็จะเรียกใช้ฟังก์ชัน InitVar() เพื่อทำการเริ่มต้นกำหนดค่าตัวแปรต่างๆ ที่จำเป็นสำหรับการใช้งานวินโดว์

2. ฟังก์ชันสมาชิก CStreamSocket::InitVars()

เป็นฟังก์ชันสมาชิกที่เป็นแบบไพรเวท (private) เป็นฟังก์ชันที่ถูกเรียกใช้ภายในคลาสเอง โดยจะทำหน้าที่กำหนดค่าของโครงสร้างข้อมูลแบบ SOCKADDR_IN ของตัวแปร m_sinLocal และ m_sinRemote เพื่อให้พร้อมสำหรับการใช้งาน

3. ฟังก์ชันสมาชิก CStreamSocket::CreateSocket()

เป็นฟังก์ชันที่ทำหน้าที่สร้างตัวเองให้เป็นวินโดว์ที่มองไม่เห็น และมีการเลือกเมสเสจของวินโดว์ที่ต้องการใช้ ได้แก่ FD_READ, FD_WRITE, FD_ACCEPT, FD_CONNECT, และ FD_CLOSE

สำหรับฟังก์ชันสมาชิกนี้มี 2 แบบให้เลือกใช้ คือ

ก. CStreamSocket::CreateSocket (int nLocalPort)

โปรแกรมที่ขอใช้วินโดว์แล้วใช้ฟังก์ชันนี้โดยให้ nLocalPort เป็นตัวเลขจำนวนเต็มใดๆ คือ ต้องการให้โปรแกรมทำตัวเป็น เซิร์ฟเวอร์ โดยจะให้บริการตามชนิดหรือความหมายของหมายเลขที่กำหนดโดย nLocalPort

ข. CStreamSocket::CreateSocket (LPSTR pszLocalService /*NULL*/)

เรียกใช้ฟังก์ชันนี้โดยกำหนดให้ pszLocalService เป็นสตริงที่บอกชนิดของบริการที่จะให้ นั่นคือโปรแกรมจะทำตัวเป็นเซิร์ฟเวอร์ที่ให้บริการตามความหมายของสตริงที่กำหนด ถ้าผู้พัฒนาโปรแกรมไม่ใส่อะไรเลยหรือให้ pszLocalService = NULL นั่นคือโปรแกรมต้องการทำตัวเป็นไคลเอนต์

4. ฟังก์ชันสมาชิก CStreamSocket::Connect()

ฟังก์ชันนี้ใช้สำหรับทำการเชื่อมต่อกับเครื่องแม่ข่ายในกรณีที่โปรแกรมทำตัวเป็นไคลเอนต์ มี 3 แบบให้เลือกใช้ คือ

ก. int CStreamSocket::Connect (LPSTR pszRemoteName, int nRemotePort)

pszRemoteName คือ สตริงที่แทนไอพีแอดเดรส เช่น 161.200.145.4

nRemotePort คือ เลขจำนวนเต็มที่แทนหมายเลขพอร์ตที่จะขอบริการ เช่น ถ้าเป็น โพรโตคอลเทลเน็ต คือ 23

ข. int CStreamSocket::Connect(LPSTR pszRemoteName, LPSTR pszRemoteService)

โดยที่ pszRemoteName จะเป็น สตริงที่แทนไอพีแอดเดรสหรือเป็นชื่อแบบอินเตอร์เน็ตก็ได้ เช่น 161.200.145.4 หรือ mail.car.chula.ac.th ส่วน pszRemoteService คือ สตริงที่บอกชื่อของบริการที่จะขอ เช่น TELNET

ค. int CStreamSocket::Connect (LPSOCKADDR_IN psinRemote)

psinRemote คือ พอยต์เตอร์ที่ชี้ไปยังอินเตอร์เน็ตแอดเดรสของเครื่องคอมพิวเตอร์แม่ข่ายที่ต้องการเชื่อมต่อ

ฟังก์ชัน Connect() ทั้ง 3 แบบ จะคืนค่าเป็นเลขจำนวนเต็มที่แทนความหมายของ การทำงานที่สำเร็จหรือไม่ และเมื่อวินซ็อกสามารถเชื่อมต่อกับเครื่องคอมพิวเตอร์แม่ข่ายได้แล้ว วินซ็อกวินโดว์จะส่งเมสเสจ CWINSOCK_YOU_ARE_CONNECTED ให้กับวินโดว์แม่ เพื่อเป็นการบอกที่สามารถเชื่อมต่อได้แล้ว

5. ฟังก์ชันสมาชิก CStreamSocket::Write()

เป็นฟังก์ชันสมาชิกที่ใช้สำหรับเขียนข้อมูลลงซ็อกเก็ตที่ขอร้องไว้จากวินโดว์ เพื่อส่งไปยังเครื่องคอมพิวเตอร์แม่ข่าย โดยข้อมูลที่เขียนจะไม่โดนเขียนลงวินซ็อกทันที แต่จะโดนเขียนลงในโครงสร้างข้อมูลที่แทนตัวของข้อมูลที่ส่ง โดยข้อมูลจะถูกส่งก็ต่อเมื่อวินซ็อกได้รับเมสเสจ FD_WRITE ฟังก์ชันนี้จะคืนค่าเป็นเลขจำนวนเต็มที่แทนความหมายของการทำงานว่าสำเร็จหรือไม่

6. ฟังก์ชันสมาชิก CStreamSocket::Read()

ฟังก์ชันนี้ใช้สำหรับอ่านข้อมูลจากเครือข่าย วินโดว์ที่เป็นวินโดว์แม่ของวินซ็อกวินโดว์จะได้รับเมสเสจจากวินซ็อกตัววินโดว์ก่อนเพื่อจะรู้ว่าข้อมูลมาแล้วจากเครื่องคอมพิวเตอร์แม่ข่าย โดยวินโดว์แม่ จะได้เมสเสจ CWINSOCK_DONE_READING

7. ฟังก์ชันสมาชิก CStreamSocket::OnWinSockEvent()

เป็นฟังก์ชันสมาชิกที่สำคัญของคลาสซีสตรีมซ็อกเก็ต จะทำหน้าที่จัดการกับเมสเสจต่างๆ ที่มาจากวินซ็อกเก็ต เมสเสจดังกล่าวคือเมสเสจที่เลือกไว้ตอนที่เรียกฟังก์ชัน CreateSocket() โดยส่วนที่เป็นเมสเสจแมพ (Message Map) ของคลาสซีสตรีมซ็อกเก็ตจะมีส่วนต่อไปนี้คือ

```
// message map functions
protected:
   //{{AFX_MSG(CStreamSocket)
   //}}AFX_MSG
    LONG OnWinSockEvent(WPARAM wParam, LPARAM lParam);
    DECLARE_MESSAGE_MAP( )
};
```

8. ฟังก์ชันสมาชิก CStreamSocket::HandleRead()

เป็นฟังก์ชันสมาชิกที่เป็นแบบไพรเวท จะถูกเรียกใช้งานเมื่อได้รับเมสเสจ FD_READ จากระบบย่อยซ็อกเก็ต โดยจะทำหน้าที่อ่านข้อมูลจากเครือข่ายที่ส่งมาไปเก็บไว้ในโครงสร้างข้อมูลที่เตรียมไว้แล้วจะส่งเมสเสจ CWINSOCK_DONE_READING ไปให้วินโดวส์แม่ เพื่อบอกว่ามีข้อมูลมาจากเครือข่ายให้มารับไป

9. ฟังก์ชันสมาชิก CStreamSocket::HandleWrite()

เป็นฟังก์ชันสมาชิกของคลาส CStreamSocket โดยฟังก์ชันนี้จะถูกเรียกใช้เมื่อมีเมสเสจ FD_WRITE มาจากระบบย่อยซ็อกเก็ตเอง โดยถ้ามีการส่งข้อมูลได้สำเร็จก็จะส่งเมสเสจ CWINSOCK_DONE_WRITING ไปยังวินโดวส์แม่

10. ฟังก์ชันสมาชิก CStreamSocket::DestroySocket()

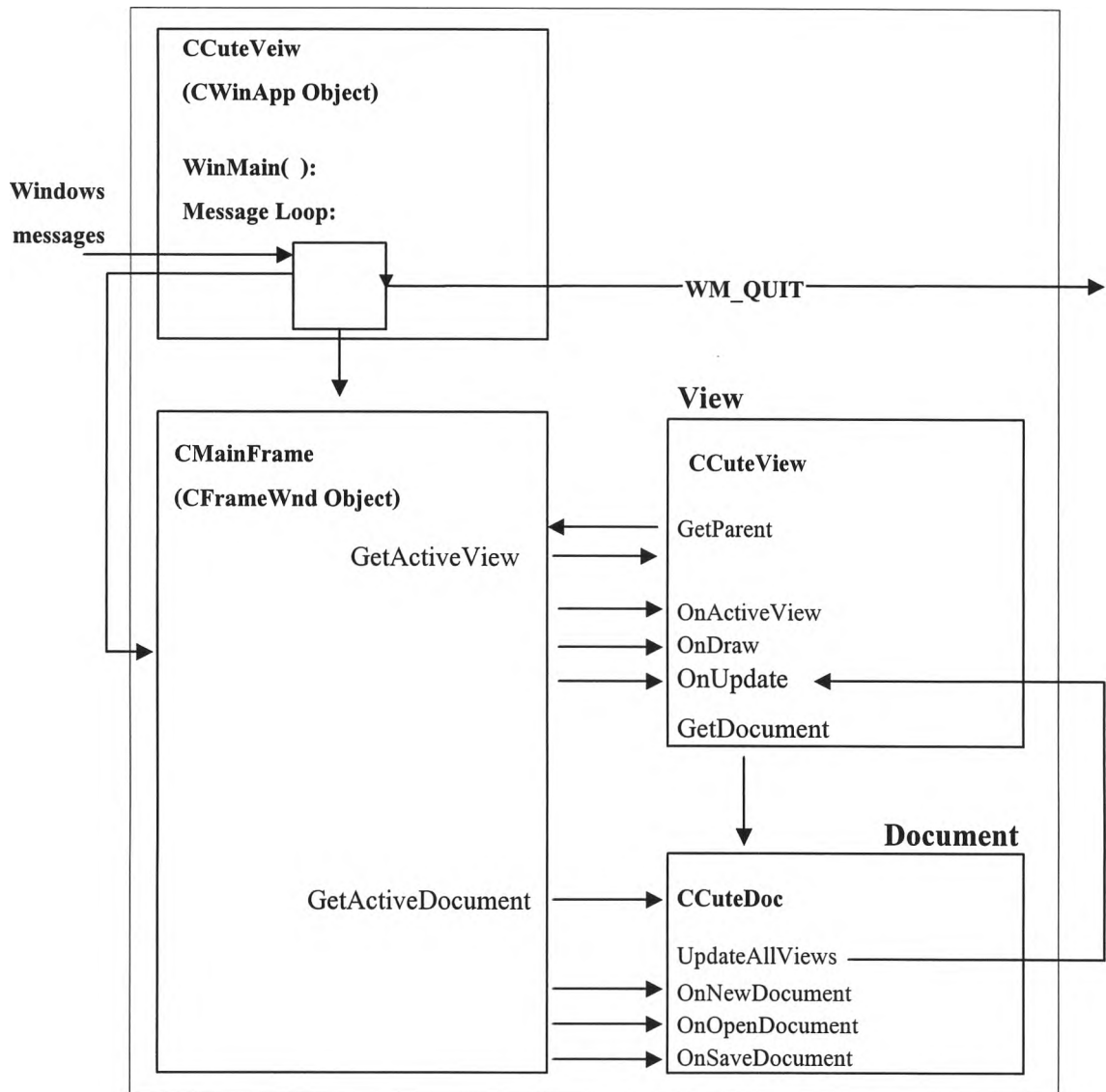
เป็นฟังก์ชันที่ถูกเรียกใช้เมื่อต้องการยกเลิกการใช้งานวินโดวซ็อกเก็ตที่ขอมมาจากวินโดวส์ โดยจะยกเลิกการขอใช้หน่วยความจำต่าง ๆ และทำลายวินโดวซ็อกเก็ตวินโดวส์ที่สร้างไว้

11. ฟังก์ชันสมาชิก CStreamSocket::LastError()

เป็นฟังก์ชันสมาชิกของคลาสซีสตรีมซ็อกเก็ต ที่ถูกเขียนเป็นฟังก์ชันแบบ in-line ใช้สำหรับตรวจสอบความผิดพลาดครั้งหลังสุดที่เกิดขึ้น โดยฟังก์ชันนี้ควรจะถูกรเรียกใช้เมื่อได้รับเมสเสจ CWINSOCK_WINSOCK_ERROR จากออบเจกต์ซีสตรีมซ็อกเก็ต

การพัฒนาโปรแกรมด้วยไมโครซอฟต์วิซิวส์ซีพลัสพลัส 4.2

ในการสร้างโปรแกรมใหม่ที่ทำงานบนระบบปฏิบัติการวินโดวส์ด้วยไมโครซอฟต์วิซิวส์ซีพลัสพลัส 4.2 (Microsoft Visual C++ 4.2) โดยปริยายเมื่อเริ่มต้นจะสร้างออบเจกต์ที่สำคัญให้ 4 ออบเจกต์ คือ ออบเจกต์แอปพลิเคชัน (Application object) ออบเจกต์ที่เป็นวินโดวหลัก (Main window object) ออบเจกต์วิว (View object) และออบเจกต์ค็อกคิวเมนต์ (Document object) สมมติว่าจะสร้างโปรแกรมที่ชื่อว่า CUTE โดย ออบเจกต์ทั้ง 4 จะมีการทำงานร่วมกันตามแผนภาพที่ 2.2



รูปที่ 2.2 แสดงการทำงานร่วมกันของออบเจกต์แอปพลิเคชัน ออบเจกต์วินโดว์หลัก ออบเจกต์วิว และออบเจกต์ดีออกิวเมนต์

1. ออบเจกต์แอปพลิเคชัน

ออบเจกต์แอปพลิเคชัน (CCuteApp) เป็นออบเจกต์ที่ถ่ายทอดมาจากคลาส CWinApp ที่อยู่ในคลังคลาสเอ็มเอฟซี (MFC) ซึ่งทำหน้าที่คล้ายกับฟังก์ชัน WinMain() เมื่อเขียนด้วยภาษาซี

2. ออบเจกต์วินโดว์หลัก

ออบเจกต์วินโดว์หลัก (MainFrame Object) เป็นออบเจกต์ที่ทำหน้าที่เป็นวินโดว์หลักของโปรแกรม ออบเจกต์นี้ถ่ายทอดมาจากคลาส CWnd ซึ่งอยู่ในคลัสเตอร์ เอ็มเอฟซี เช่นกัน ผู้พัฒนาโปรแกรมสามารถปรับเปลี่ยนคุณลักษณะของวินโดว์ของโปรแกรมได้โดยกระทำในออบเจกต์นี้

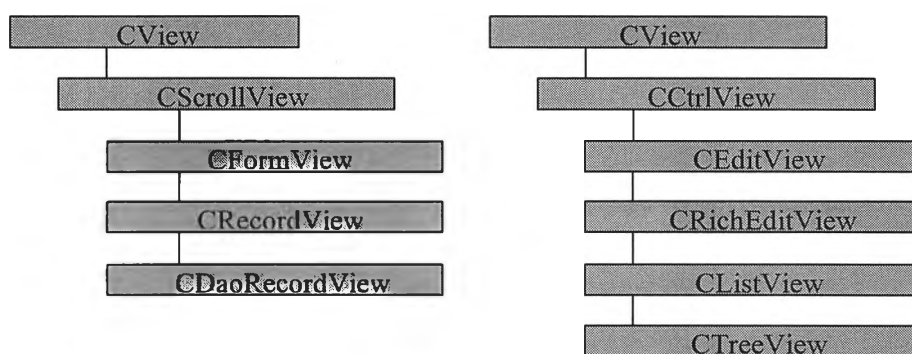
3. ออบเจกต์ด็อกคิวเมนต์

ออบเจกต์ด็อกคิวเมนต์ (Document Object) เป็นออบเจกต์ที่ถ่ายทอดมาจากคลัสเตอร์ด็อกคิวเมนต์ (CDocument) ในคลัสเตอร์ เอ็มเอฟซี ทำหน้าที่จัดการกับข้อมูล มีความสามารถในการจัดเก็บข้อมูลลงหน่วยความจำ เช่น ฮาร์ดดิสก์มีความสามารถในการทำงานกับเพิ่มข้อมูล

4. ออบเจกต์วิว

ออบเจกต์วิว (View Object) เป็นออบเจกต์ที่มีความสามารถในการแสดงผลข้อมูลออกจอภาพ สามารถตอบสนองกับเมสเสจของวินโดว์ได้หลายเมสเสจ เช่น สามารถตอบสนองเมสเสจ WM_CHAR ซึ่งเป็นเมสเสจที่เกิดขึ้นเมื่อมีการกดปุ่มอักขระบนแป้นพิมพ์

ในขณะที่สร้างโปรแกรมขึ้นมาใหม่ โปรแกรมไมโครซอฟต์วิซวลส์พลัส 4.2 จะให้ผู้ใช้พัฒนาโปรแกรมเลือกออบเจกต์วิวได้หลายแบบ ตามลักษณะการใช้งาน โดยออบเจกต์วิวชนิดต่าง ๆ นั้นมีการถ่ายทอดเป็นลำดับ ดังรูปที่ 2.3



รูปที่ 2.3 แสดงลำดับการถ่ายทอดของคลาสวิวประเภทต่าง ๆ

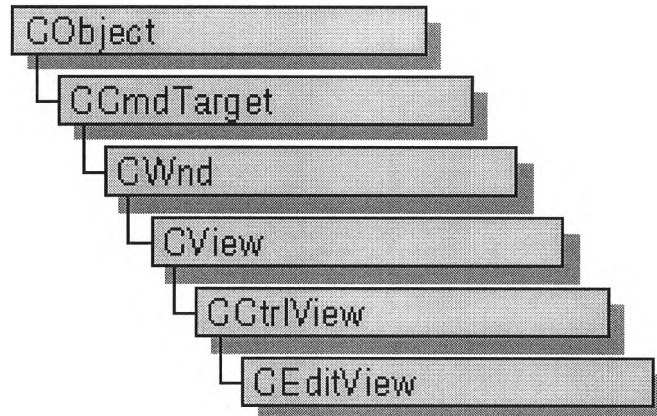
จากรูปที่ 2.3 คลาสวิวประเภทต่างๆ มีคำอธิบายดังตารางที่ 2.1

คลาส	คำอธิบาย
CView	เป็นคลาสพื้นฐานที่มีความสามารถในการแสดงผลข้อมูล คลาสวิวที่มีความสามารถพิเศษเพิ่มเติมออกไปนั้นจะถ่ายทอดมาจากคลาสชีวิวนี้
CScrollView	เป็นวิวคลาสที่ถ่ายทอดมาจากคลาสชีวิว มีฟังก์ชันในการเลื่อนจอภาพมาให้
CCtrlView	เป็นวิวคลาสที่ถ่ายทอดมาจากคลาสชีวิว มีความสามารถในการจัดการกับคอนโทรลรูปแบบต่างๆ ของวินโดวส์ นอกจากนั้นยังรวมถึงคอนโทรลชนิดใหม่ๆ ที่มีอยู่ในระบบปฏิบัติการวินโดวส์ 95 เช่น ListView, TreeView, RichView เป็นต้น
CEditView	เป็นคลาสที่ถ่ายทอดมาจากคลาส CCtrlView มีความสามารถในการเป็นเทคซ์ต้อดิเตอร์ แบบพื้นฐาน
CRichEditView	เป็นคลาสที่ถ่ายทอดมาจากคลาส CCtrlView มีความสามารถมากกว่าเทคซ์ต้อดิเตอร์ธรรมดา มีความสามารถในการควบคุม RichEdit ของวินโดวส์ 95
CTreeView	เป็นคลาสที่ถ่ายทอดมาจากคลาส CCtrlView มีความสามารถในการจัดการกับคอนโทรลแบบ ListView ของวินโดวส์ 95 ได้
CFormView	เป็นคลาสที่ถ่ายทอดมาจากคลาส CScrollView มีลักษณะความสามารถของฟอร์มที่รับข้อมูลจากผู้ใช้ในรูปแบบต่างๆ ได้หลายวิธี
CRecordView	เป็นวิวคลาสที่ถ่ายทอดมาจากคลาส CScrollView มีความสามารถแสดงระเบียบของฐานข้อมูลได้
CDaoRecordView	เป็นคลาสที่ถ่ายทอดมาจากคลาส CScrollView โดยจะเพิ่มความสามารถของ CRecordView ในเรื่องของ DAO

ตารางที่ 2.1 แสดงความหมายของออบเจกต์วิวประเภทต่าง ๆ

คลาสซีอีดิทวิว

เป็นคลาสที่เป็นส่วนหนึ่งของคลังคลาส เอ็มเอฟซี (MFC) มีลำดับการถ่ายทอด โดยแสดงดังรูปที่ 2.4



รูปที่ 2.4 แสดงลำดับการถ่ายทอดของคลาสซีอีดิทวิว

คลาสซีอีดิทวิวมีความสามารถพื้นฐานของการเป็นเทคซ์อีดิเตอร์ นอกจากนั้นยังมีความสามารถเพิ่มเติมอีก ได้แก่ การพิมพ์ การค้นหาข้อความที่ต้องการและแทนที่ข้อมูลที่กำหนดในตำแหน่งต่างๆ ได้

คลาสซีอีดิทวิวมีตัวแปรและฟังก์ชันสมาชิกต่างๆ แสดงดังตารางที่ 2.2

ตัวแปรและฟังก์ชันสมาชิก	คำอธิบาย
dwStyleDefault	เป็นรูปแบบโดยปริยายของออบเจกต์แบบซีอีดิทวิว
CEditView	เป็นคอนสตรัคเตอร์ของออบเจกต์แบบ ซีอีดิทวิว
GetEditCtrl	เป็นฟังก์ชันที่ใช้ในการควบคุม ซีอีดิทวิว โดยมีลักษณะการควบคุมที่มีลักษณะเดียวกับในออบเจกต์แบบซีอีดิท
GetPrinterFont	ใช้สำหรับรับฟอนต์ของพรินเตอร์ที่กำลังใช้อยู่

ตารางที่ 2.2 แสดงคำอธิบายของตัวแปรและฟังก์ชันสมาชิกของคลาสซีอีดิทวิว

ตัวแปรและฟังก์ชันสมาชิก	คำอธิบาย
GetSelectedText	ใช้รับข้อมูลที่กำลังเลือกอยู่
LockBuffer	ใช้ควบคุมข้อมูลในบัฟเฟอร์ไม่ให้มีการเปลี่ยนแปลง
UnlockBuffer	ใช้สำหรับยกเลิกการควบคุมข้อมูลบัฟเฟอร์
GetBufferLength	ใช้หาความยาวของบัฟเฟอร์ข้อมูล
SetPrinterFont	กำหนดฟอนต์ให้พรินเตอร์ใหม่
SetTabStops	ใช้กำหนดระยะเย็บ
FindText	ใช้ค้นหากลุ่มอักขระในเอกสาร
PrintInsideRect	แสดงข้อมูลลงในกรอบสี่เหลี่ยมที่กำหนด
SerializeRaw	เก็บข้อมูลที่กำลังแสดงอยู่บนวีวลงดิสก์
OnFindNext	ค้นหาข้อมูลครั้งถัดไป
OnReplaceAll	ใช้แทนที่ข้อมูลในวิวทั้งหมดด้วยข้อมูลที่กำหนด
OnReplaceSel	ใช้แทนที่ข้อมูลในตำแหน่งที่มีการเลือกไว้
OnTextNotFound	ถูกเรียกใช้เมื่อหาข้อมูลในวิวไม่พบ

ตารางที่ 2.2 แสดงคำอธิบายของตัวแปรและฟังก์ชันสมาชิกของคลาสซีอีดิทวิว (ต่อ)

คลาสซีอีดิท

คลาสซีอีดิท (CEdit) เป็นคลาสที่มีความสามารถในการแสดงข้อมูลและมีฟังก์ชันในการจัดการกับข้อมูลในอีดิทคอนโทรล โดยอีดิทคอนโทรล คือ วินโดว์รูปสี่เหลี่ยมที่กำหนดให้ผู้ใช้งานกรอกข้อมูลได้ มีตัวแปรและฟังก์ชันสมาชิกของคลาสดังตารางที่ 2.3

ตัวแปรและฟังก์ชันสมาชิก	คำอธิบาย
CEdit	เป็นคอนสตรัคเตอร์ของออบเจกต์แบบซีอีดิท
Create	ใช้สำหรับสร้างวินโดวซีอีดิท
GetSel	ใช้หาตำแหน่งตัวอักษรตัวแรกและตำแหน่งของตัวอักษรตัวสุดท้ายที่บริเวณที่ทำการเลือกไว้
ReplaceSel	ใช้สำหรับแทนที่บริเวณที่เลือกด้วยข้อมูลที่กำหนด
SetSel	ใช้กำหนดช่วงของการเลือกข้อมูลในอีดิทคอนโทรล
Clear	ลบข้อมูลในบริเวณที่ทำการเลือกไว้
Copy	ทำสำเนาข้อมูลที่เลือกไปเก็บไว้ที่คลิปบอร์ด
Cut	ลบข้อมูลที่เลือกไว้พร้อมทั้งทำสำเนาไปไว้ที่คลิปบอร์ด
Paste	แทรกข้อมูลจากคลิปบอร์ดลงในตำแหน่งที่เคอร์เซอร์อยู่
Undo	ยกเลิกคำสั่งก่อนหน้านี้ 1 คำสั่ง
GetFirstVisibleLine	ใช้สำหรับหาบรรทัดบนสุดที่ปรากฏอยู่ในวิวเป็นบรรทัดที่เท่าไรของเอกสาร
LineLength	หาความยาวของบรรทัดที่กำหนดโดยตำแหน่งของอักขระ
LineFromChar	ใช้หาหมายเลขบรรทัดที่กำหนดโดยตำแหน่งของอักขระ
GetRect	ใช้กำหนดกรอบสี่เหลี่ยมใดๆ ในอีดิทคอนโทรล
GetLineCount	ใช้หาจำนวนบรรทัดทั้งหมดที่อยู่ในวิว
LineIndex	ใช้สำหรับหาตำแหน่งของตัวอักษรตัวแรกของบรรทัดที่กำหนด

ตารางที่ 2.3 แสดงคำอธิบายตัวแปรและฟังก์ชันสมาชิกของคลาสซีอีดิท