

Data Communication (CE14773)



Chungnam National University
Dept. of Computer Science and Engineering
Computer Communication Laboratory

Sangdae Kim - 00반 / Cheonyong Kim- 01반





Contents

- ◆ Network Overview
 - ❖ Data Link Layer
 - ❖ Byte Ordering
 - ❖ Common Interface – Virtual Function

- ◆ Inter-Process Communication Implement
 - ❖ IPC (inter-Process Communication)

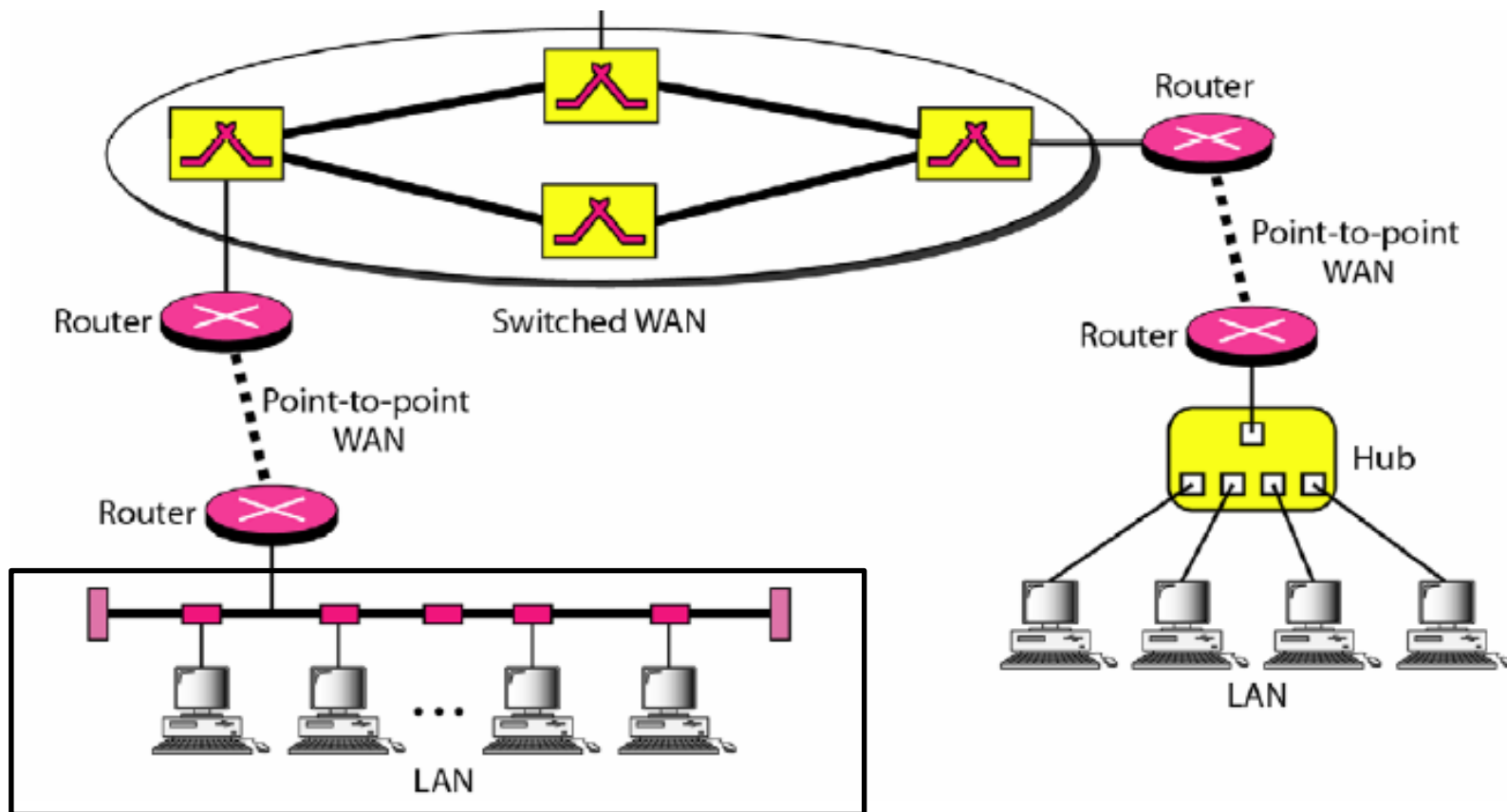


Network Overview



Data Link Layer

◆ An example of network





Byte Ordering (1/5)

- ◆ Different machine architectures sometimes store data using different byte orders.
 - ❖ Big Endian
 - ◆ The most significant byte is on the left end of a word.
 - ◆ Macintosh (Motorola) machines
 - ◆ Network Byte Order
 - ❖ Little Endian
 - ◆ The most significant byte is on the right end of a word.
 - ◆ Intel-based machines
 - ◆ Host Byte Order
- ◆ The terms “Big Endian” & “Little Endian” indicate which end of the **multi-byte** value, big endian or little endian, is stored at the starting address of the value.



Byte Ordering (2/5)

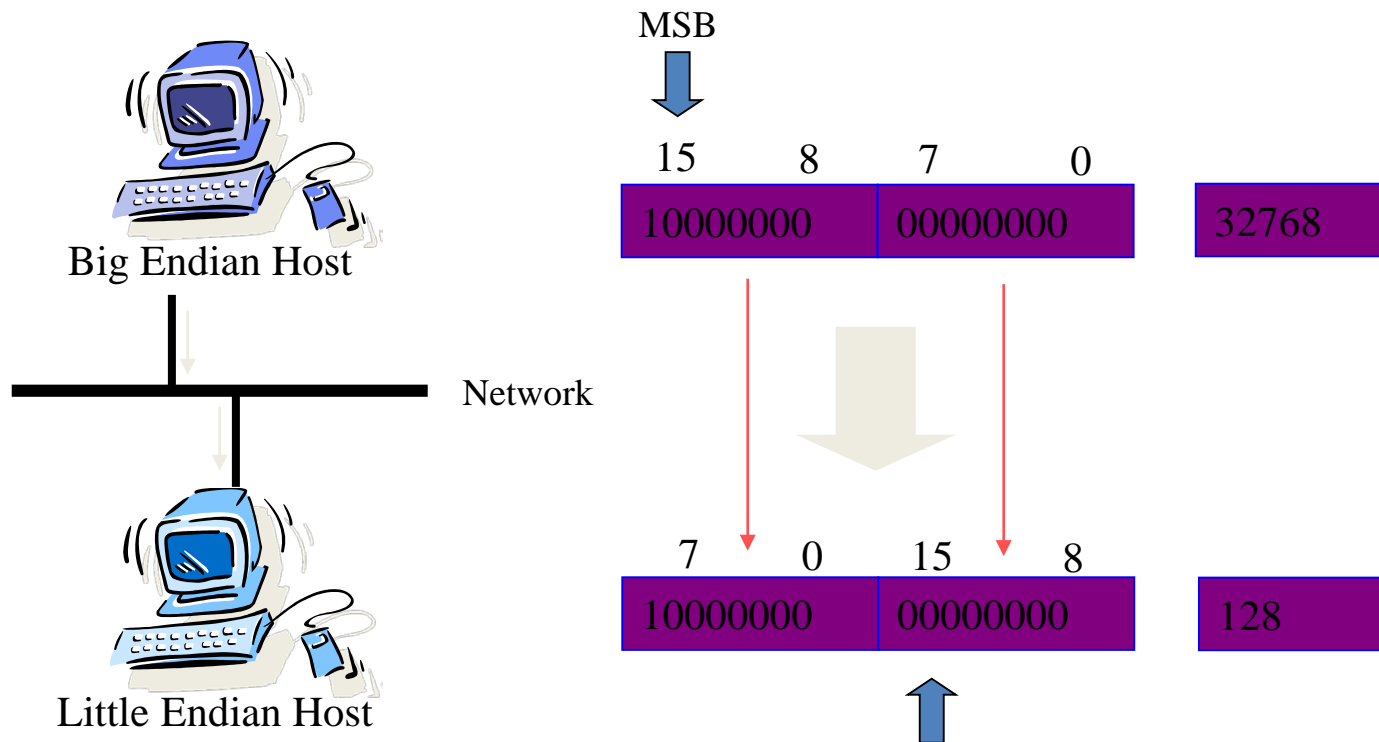
◆ Network Byte Order & Host Byte Order

- ❖ So that machines with different byte order conventions can communicate, the Internet protocols specify a canonical byte order convention for data transmitted over the network.
- ❖ If you are encoding integer data in the messages sent through the socket, you should convert this to network byte order too.
- ❖ If you don't do this, your program may fail when running on or talking to other kinds of machines.



Byte Ordering (3/5)

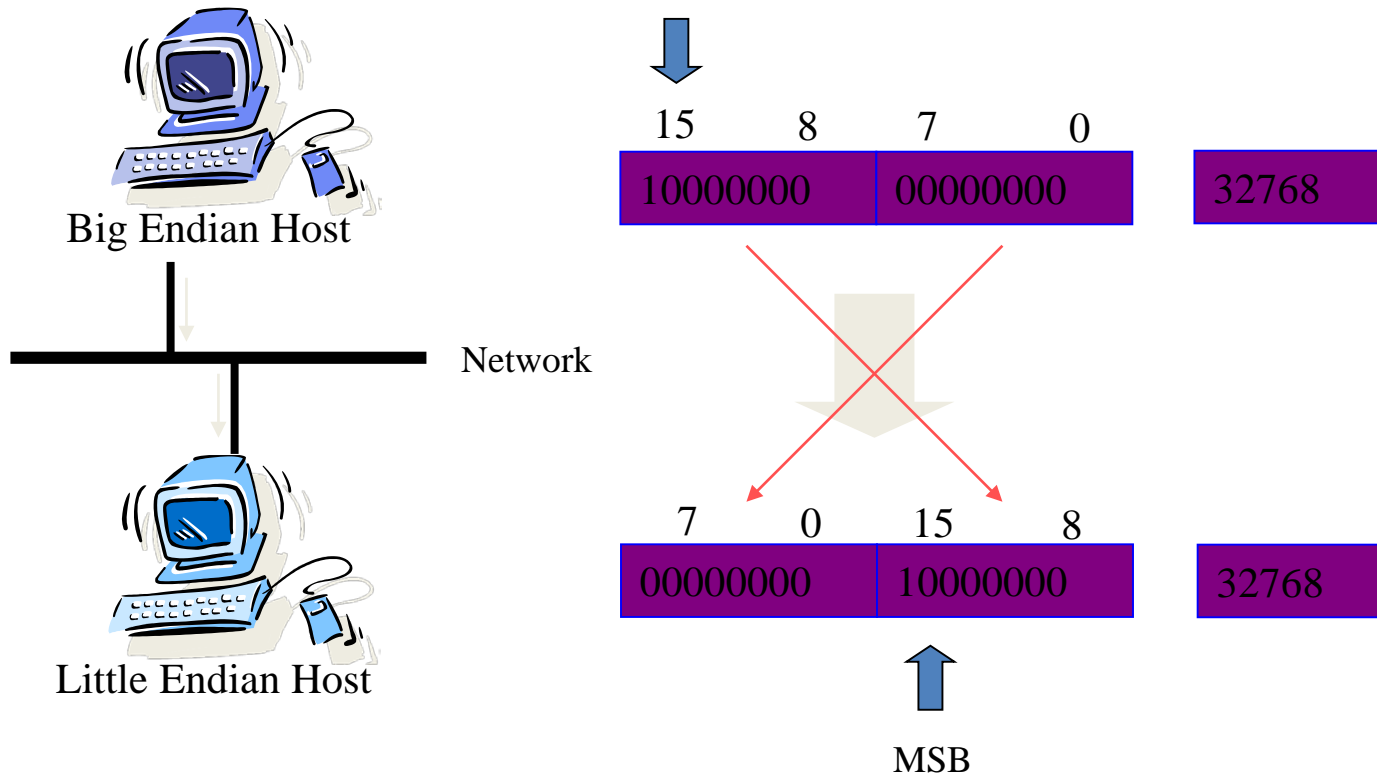
◆ When Byte ordering is **NOT** applied





Byte Ordering (4/5)

◆ When Byte ordering is applied



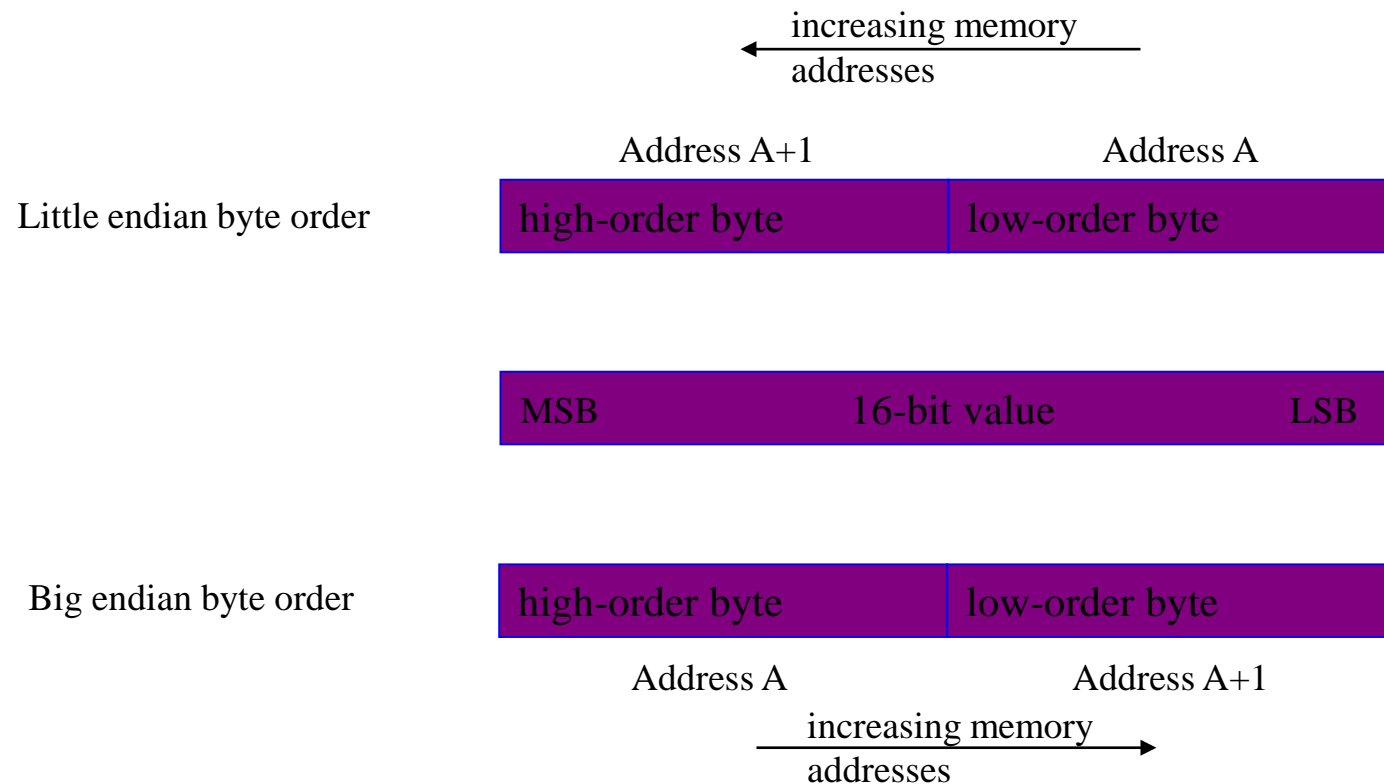


Figure . Little endian byte order and big endian byte order for 16-bit integer



Byte Ordering (5/5)

◆ 32-bit byte order

```
unsigned long int ntohl(unsigned long int x)
{
    return (
        ((x & 0x000000ffU) << 24) |
        ((x & 0x0000ff00U) << 8) |
        ((x & 0x00ff0000U) >> 8) |
        ((x & 0xff000000U) >> 24)
    );
}
```

- Little endian host must convert byte orders to 'network byte order' for sending data over the network.c



(x & 0x000000ffU) << 24)

x
0x000000ffU
&(Mask)
<<(shift)

1	2	3	4
			4
4			

(x & 0x0000ff00U) << 8)

1	2	3	4
		3	
	3		

(x & 0x00ff0000U) >> 8)

1	2	3	4
	2		
		2	

(x & 0xff000000U) >> 24)

1	2	3	4
1			
			1

((x & 0x000000ffU) << 24) |
((x & 0x0000ff00U) << 8) |
((x & 0x00ff0000U) >> 8) |
((x & 0xff000000U) >> 24)

4			
	3		
		2	
			1

Result

4	3	2	1
---	---	---	---



Common Interface – Virtual Function

◆ Virtual Function Overview

❖ 가상함수 정의

- ◆ 가상함수는 기본 클래스 내에 선언, 파생 클래스에서 의해 재정의된 클래스 멤버 함수
- ◆ 가상함수를 생성하기 위해서 키워드 `virtual`을 기본 클래스의 함수 선언 앞에 사용
- ◆ 가상함수를 포함하는 클래스가 상속될 때, 파생 클래스는 그 파생 클래스에 관련되도록 가상함수를 재정의

❖ “한가지 인터페이스를 통한 여러 가지 활용법”

- ◆ 기본 클래스 내의 가상함수: 그 함수에 대한 인터페이스(interface)의 형태 정의
- ◆ 파생 클래스 내의 가상함수: 그 파생 클래스에 관련된 연산을 구현(specific method)



Common Interface – Virtual Function

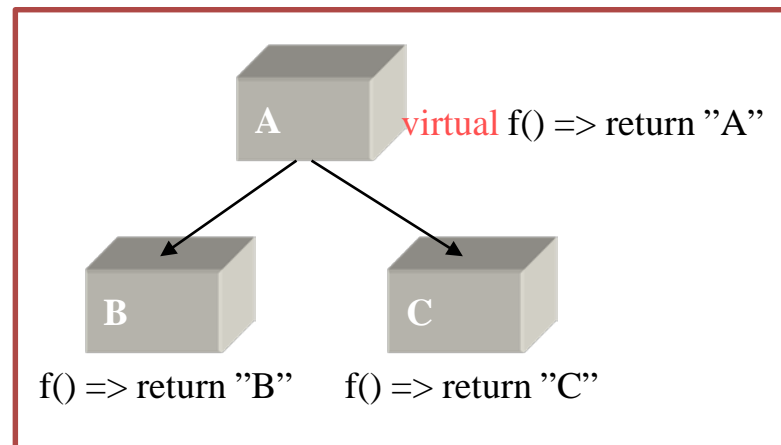
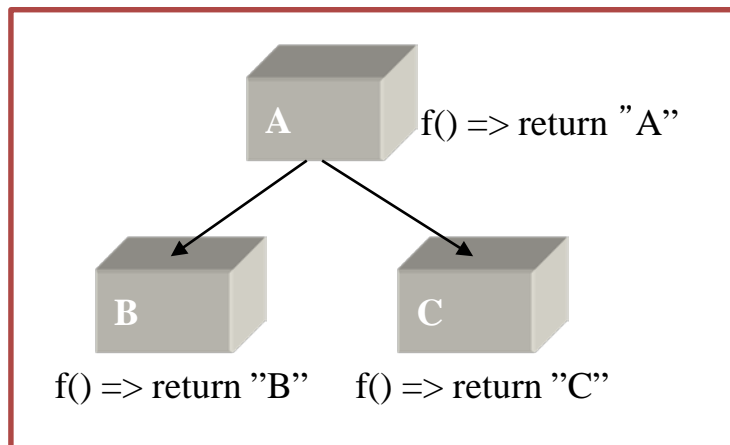
◆ Characteristic of virtual function

- ❖ 기본 포인터가 파생 클래스의 객체를 가리키고, 가상함수가 그 포인터를 통해 호출될 때, 컴파일러는 포인터가 가리키고 있는 객체의 형에 근거하여 함수의 버전을 결정한다.
- ❖ Overriding (VS. Overloading)
 - ◆ 매개변수의 형과 수, 그리고 반환 값의 형이 정확히 일치
 - ◆ 클래스의 멤버
 - ◆ 소멸자 함수는 가상함수가 될 수 있지만, 생성자 함수는 가상함수가 될 수 없다.



Common Interface – Virtual Function

◆ The Difference Between Not Virtual Function & Virtual Function



```
class A *a;  
class B b;  
class C c;  
a = &b;  
printf( a->f() ) ; // "A"  
a = &c;  
printf( a->f() ) ; // "A"
```

"B"

"C"



Common Interface - Implement

```
class CBaseLayer
{
public:
    char*                GetLayerName( );
    CBaseLayer*          GetUnderLayer( );
    CBaseLayer*          GetUpperLayer( int nIndex );
    void                 SetUnderUpperLayer( CBaseLayer* pUULayer = NULL );
    void                 SetUpperUnderLayer( CBaseLayer* pUULayer = NULL );
    void                 SetUnderLayer( CBaseLayer* pUnderLayer = NULL );
    void                 SetUpperLayer( CBaseLayer* pUpperLayer = NULL );
    CBaseLayer( char* pName = NULL );
    virtual ~CBaseLayer();
    virtual BOOL         Send( unsigned char*, int )           { return FALSE ; }
    virtual BOOL         Receive( unsigned char* ppayload )    { return FALSE ; }
    virtual BOOL         Receive( )                            { return FALSE ; }
protected:
    char*                m_pLayerName;
    CBaseLayer*          mp_UnderLayer;
    CBaseLayer*          mp_aUpperLayer[ MAX_LAYER_NUMBER ];
    int                  m_nUpperLayerCount;
};
```

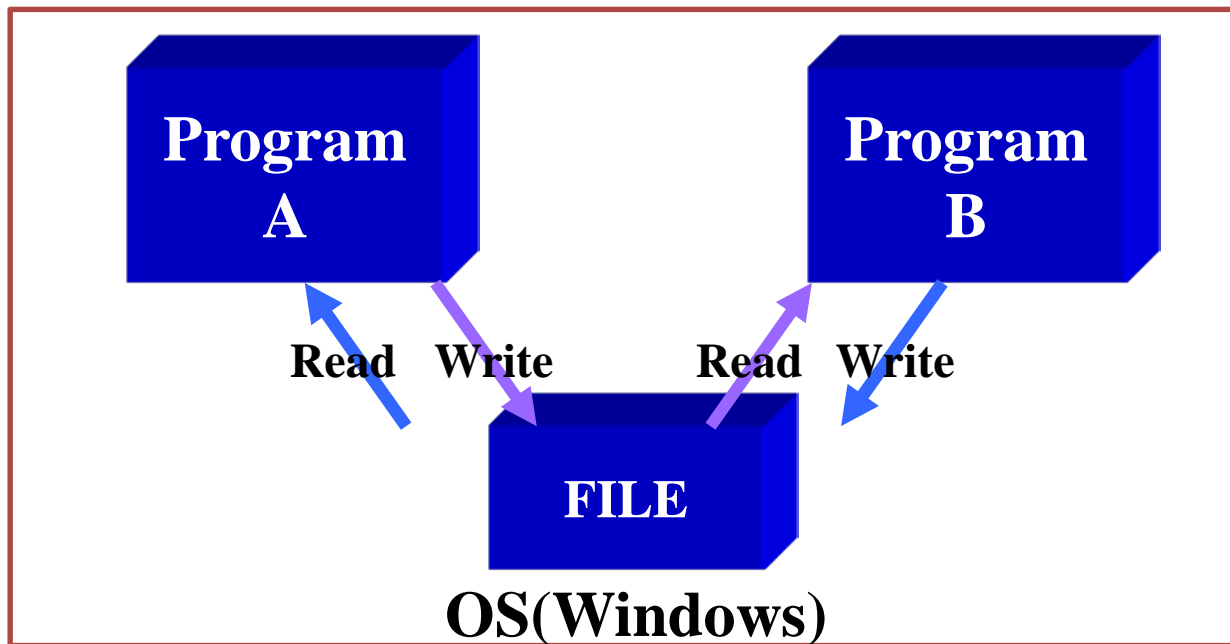


Inter-Process Communication Implement



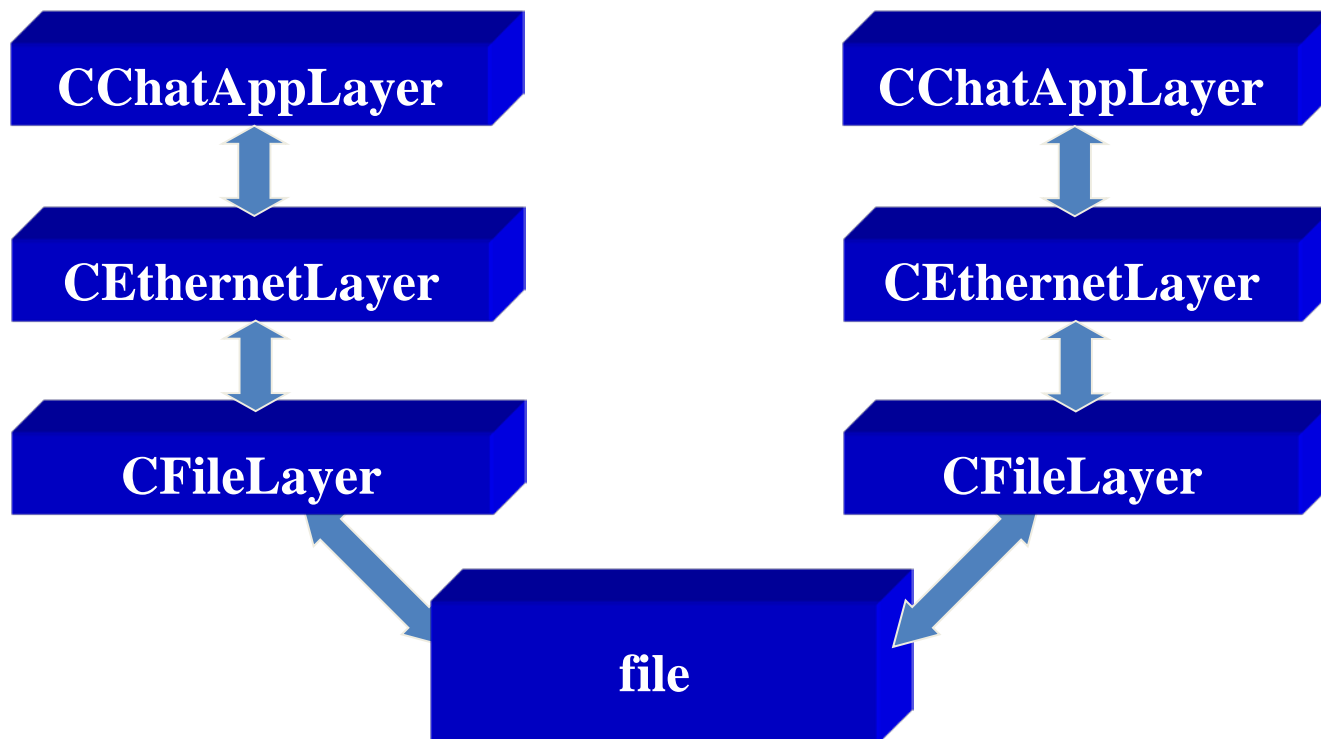
Data Propagation by Using IPC

- ◆ Window상의 두 개의 프로그램(process)이 하나의 파일을 공유하여 통신을 하고 있다.
 - ❖ Program B가 write동작을 완료하면, Program A가 read한다.
 - ❖ Program A가 write동작을 완료하면, Program B가 read한다.





Hierarchical Architecture





IPC Implement

- ◆ 제공된 기본 소스에서 IPC 구현을 위해 해야 할 일
 - ❖ Message Queue를 사용하기 위한 Register 등록
 - ❖ Layer 연결
 - ❖ 각 Layer의 send, receive 함수 구현
 - ❖ Send 버튼이 눌렸을 때의 동작 처리



IPC Implement

◆ Message Queue를 사용하기 위한 Register 등록 (IPCAppDlg.cpp)

```
// 레지스트리에 등록하기 위한 변수
```

```
UINT nRegSendMsg ;
```

```
UINT nRegAckMsg ;
```

```
BEGIN_MESSAGE_MAP(CIPCAppDlg, CDialog)
```

```
//{{AFX_MSG_MAP(CIPCAppDlg)
```

```
ON_WM_SYSCOMMAND( )
```

```
ON_WM_PAINT( )
```

```
ON_WM_QUERYDRAGICON( )
```

```
ON_BN_CLICKED( IDC_BUTTON_SEND, OnSendMessage)
```

```
ON_BN_CLICKED( IDC_BUTTON_ADDR, OnButtonAddrSet)
```

```
ON_BN_CLICKED( IDC_CHECK_TOALL, OnCheckBroadcast)
```

```
ON_WM_TIMER( )
```

```
//}}AFX_MSG_MAP
```

```
ON_REGISTERED_MESSAGE( nRegSendMsg, OnRegSendMsg )
```

```
//////////////////////////////////// fill the blank //////////////////////////////////////
```

```
ON_REGISTERED_MESSAGE( nRegAckMsg, OnRegAckMsg)
```

```
////////////////////////////////////
```



IPC Implement

◆ Message Queue를 사용하기 위한 Register 등록 (IPCAppDlg.cpp)

```
void CIPCAppDlg::SetRegistryMessage( )
{
    nRegSendMsg = RegisterWindowMessage( "Send IPC Message" );
    ////////////////////////////////// fill the blank //////////////////////////////////
    nRegAckMsg = RegisterWindowMessage( "Ack IPC Message" );
    //////////////////////////////////////////////////////////////////////
}
```



IPC Implement

◆ Layer 연결 (BaseLayer.cpp)

```
void CBaseLayer::SetUpperUnderLayer(CBaseLayer *pUULayer)
{
    if ( !pUULayer ) // if the pointer is null,
    {
#ifdef _DEBUG
        TRACE( "[CBaseLayer::SetUpperUnderLayer] The variable , 'pUULayer' is NULL" );
#endif
        return ;
    }

    ////////////////////////////////// fill the blank //////////////////////////////////
    SetUpperLayer(pUULayer);
    pUULayer->SetUnderLayer(this);
    //////////////////////////////////
}
```



IPC Implement

◆ Layer 연결 (IPCAppDlg.cpp)

```

CIPCAppDlg::CIPCAppDlg(CWnd* pParent /*=NULL*/)
: CDialog(CIPCAppDlg::IDD, pParent),
  CBaseLayer( "ChatDlg" ),
  m_bSendReady( FALSE ),
  m_nAckReady( -1 )
{
    //{AFX_DATA_INIT(CIPCAppDlg)
    m_unDstAddr = 0;
    m_unSrcAddr = 0;
    m_stMessage = _T( "" );
    //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);

    m_LayerMgr.AddLayer( new CChatAppLayer( "ChatApp" ) );
    m_LayerMgr.AddLayer( new CEthernetLayer( "Ethernet" ) );
    m_LayerMgr.AddLayer( new CFileLayer( "File" ) );
    m_LayerMgr.AddLayer( this );
    ////////////////////////////////////// fill the blank //////////////////////////////////////
    m_LayerMgr.ConnectLayers("File ( *Ethernet ( *ChatApp ( *ChatDlg ) ) )");
    //////////////////////////////////////

```



IPC Implement

◆ 각 Layer의 send, receive 함수 구현 (IPCAppDlg.cpp)

```
void CIPCAppDlg::SendData( )
{
    CString MsgHeader ;
    if ( m_unDstAddr == (unsigned int)0xff )
        MsgHeader.Format( "[%d:BROADCAST] ", m_unSrcAddr ) ;
    else
        MsgHeader.Format( "[%d:%d] ", m_unSrcAddr, m_unDstAddr ) ;

    m_ListChat.AddString( MsgHeader + m_stMessage ) ;

    ////////////////////////////////// fill the blank //////////////////////////////////
    int nlength = m_stMessage.GetLength( );
    unsigned char* ppayload = new unsigned char[nlength+1];
    memcpy(ppayload,(unsigned char*)(LPCTSTR)m_stMessage, nlength);
    ppayload[nlength] = '\0';

    m_ChatApp->Send(ppayload, m_stMessage.GetLength( ));|
    //////////////////////////////////////////////////////////////////////
}
```




IPC Implement

◆ 각 Layer의 send, receive 함수 구현 (IPCAppDlg.cpp)

```
3 LRESULT CIPCAppDlg::OnRegSendMsg(WPARAM wParam, LPARAM lParam)
{
    ////////////////////////////////////// fill the blank //////////////////////////////////////
    if( m_nAckReady )
        if(m_LayerMgr.GetLayer("File")->Receive()){
            ::SendMessage( HWND_BROADCAST, nRegAckMsg, 0, 0 );
        }
    //////////////////////////////////////
    return 0 ;
}
```



IPC Implement

◆ 각 Layer의 send, receive 함수 구현 (ChatAppLayer.cpp)

```
BOOL CChatAppLayer::Send(unsigned char *ppayload, int nlength)
{
    m_sHeader.app_length = (unsigned short) nlength ;

    BOOL bSuccess = FALSE ;
    ////////////////////////////////////// fill the blank //////////////////////////////////////
    memcpy(m_sHeader.app_data, ppayload, nlength>APP_DATA_SIZE ? APP_DATA_SIZE : nlength);
    bSuccess = mp_UnderLayer->Send( (unsigned char*)&m_sHeader, nlength+APP_HEADER_SIZE);
    //////////////////////////////////////

    return bSuccess ;
}
```



IPC Implement

◆ 각 Layer의 send, receive 함수 구현 (ChatAppLayer.cpp)

```

BOOL CChatAppLayer::Receive( unsigned char* ppayload )
{
    PCHAT_APP_HEADER app_hdr = (PCHAT_APP_HEADER) ppayload ;

    if ( app_hdr->app_dstaddr == m_sHeader.app_srcaddr ||
        ( app_hdr->app_srcaddr != m_sHeader.app_srcaddr &&
          app_hdr->app_dstaddr == (unsigned int) 0xff ) )
    {
        /////////////////////////////////// fill the blank ///////////////////////////////////
        unsigned char GetBuff[APP_DATA_SIZE];
        memset(GetBuff, '0', APP_DATA_SIZE);
        memcpy(GetBuff, app_hdr->app_data, app_hdr->app_length>APP_DATA_SIZE? APP_DATA_SIZE : app_hdr->app_length);
        CString Msg;
        if(app_hdr->app_dstaddr == (unsigned int)0xff){
            Msg.Format("[%d:BROADCAST] %s", app_hdr->app_srcaddr, (char*)GetBuff);
        }else{
            Msg.Format("[%d:%d] %s", app_hdr->app_srcaddr, app_hdr->app_dstaddr, (char*)GetBuff);
        }
        mp_aUpperLayer[0] ->Receive( (unsigned char*) Msg.GetBuffer(0));
        ///////////////////////////////////
        return TRUE ;
    }
    else
        return FALSE ;
}

```



IPC Implement

◆ 각 Layer의 send, receive 함수 구현 (EthernetLayer.cpp)

```
BOOL CEthernetLayer::Send(unsigned char *ppayload, int nlength)
{
    memcpy( m_sHeader.enet_data, ppayload, nlength );

    BOOL bSuccess = FALSE ;
    ////////////////////////////////////// fill the blank //////////////////////////////////////
    bSuccess = mp_UnderLayer->Send( (unsigned char*)&m_sHeader, nlength+ETHER_HEADER_SIZE);
    ////////////////////////////////////// fill the blank //////////////////////////////////////

    return bSuccess ;
}

BOOL CEthernetLayer::Receive( unsigned char* ppayload )
{
    PETHERNET_HEADER pFrame = (PETHERNET_HEADER) ppayload ;

    BOOL bSuccess = FALSE ;
    ////////////////////////////////////// fill the blank //////////////////////////////////////
    bSuccess = mp_aUpperLayer[0]->Receive( (unsigned char*)pFrame->enet_data); |
    ////////////////////////////////////// fill the blank //////////////////////////////////////

    return bSuccess ;
}
```



IPC Implement

◆ 각 Layer의 send, receive 함수 구현 (EthernetLayer.cpp)

```
] unsigned char* CEthernetLayer::GetDestInAddress( )  
{  
    ////////////////////////////////////// fill the blank //////////////////////////////////////  
    return m_sHeader.enet_dstaddr;  
    ////////////////////////////////////// fill the blank //////////////////////////////////////  
}  
_  
  
] void CEthernetLayer::SetSourceAddress(unsigned char *pAddress)  
{  
    ////////////////////////////////////// fill the blank //////////////////////////////////////  
    memcpy( m_sHeader.enet_srcaddr , pAddress, 6);  
    ////////////////////////////////////// fill the blank //////////////////////////////////////  
}  
_
```



IPC Implement

◆ 각 Layer의 send, receive 함수 구현 (FileLayer.cpp)

```
CFFileLayer::~CFFileLayer( )
{
    TRY
    {
        /////////////////////////////////// fill the blank ///////////////////////////////////
        CFile::Remove( "IpcBuff.txt" );
        ///////////////////////////////////
    }
}

BOOL CFFileLayer::Send(unsigned char *ppayload, int nlength)
{
    TRY
    {
        CFile m_FileDes( "IpcBuff.txt",
                        CFile::modeCreate | CFile::modeWrite );
        /////////////////////////////////// fill the blank ///////////////////////////////////
        m_FileDes.Write(ppayload,nlength);
        m_FileDes.Close();
        ///////////////////////////////////
    }
}
```



IPC Implement

◆ 각 Layer의 send, receive 함수 구현 (FileLayer.cpp)

```
BOOL CFileLayer::Send(unsigned char *ppayload, int nlength)
{
    TRY
    {
        CFile m_FileDes( "IpcBuff.txt",
                        CFile::modeCreate | CFile::modeWrite );
        /////////////////////////////////// fill the blank ///////////////////////////////////
        m_FileDes.Write(ppayload,nlength);
        m_FileDes.Close( );
        ///////////////////////////////////
```



IPC Implement

◆ 각 Layer의 send, receive 함수 구현 (FileLayer.cpp)

```
BOOL CFileLayer::Receive( )
{
    unsigned char m_File_data[ETHER_MAX_SIZE];
    memset(m_File_data, 0, ETHER_MAX_SIZE);

    TRY
    {
        CFile m_FileDes( "IpcBuff.txt", CFile::modeRead );

        //////////////////////////////////// fill the blank ////////////////////////////////////
        int nlength = ETHER_HEADER_SIZE+ETHER_MAX_DATA_SIZE;
        unsigned char* ppayload = new unsigned char[nlength+1];

        m_FileDes.Read(ppayload,nlength);
        ppayload[nlength]='\0';

        if(!mp_aUpperLayer[0]->Receive(ppayload)){
            m_FileDes.Close( );
            return FALSE;
        }

        ////////////////////////////////////
```




IPC Implement

◆ Send 버튼이 눌렸을 때의 동작 처리

```
void CIPCAppDlg::OnSendMessage()  
{  
    // TODO: Add your control notification handler code here  
    UpdateData( TRUE );  
  
    if ( !m_stMessage.IsEmpty() )  
    {  
        SetTimer( 1, 2000, NULL );  
        m_nAckReady = 0 ;  
  
        SendData( ) ;  
        m_stMessage = "" ;  
  
        (CEdit*) GetDlgItem( IDC_EDIT_MSG )->SetFocus( ) ;  
  
        ////////////////////////////////// fill the blank //////////////////////////////////  
        ::SendMessage(HWND_BROADCAST, nRegSendMsg, 0, 0);  
        //////////////////////////////////////////////////////////////////////  
    }  
}
```