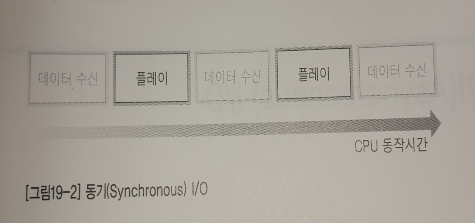
19장. 비동기 I/O 와 APC

Session 01 비동기 (Asynchronous) I/O

| 비동기 I/O의 이해



CPU가 동작하지않는 '데이터 수신' (I/O 영역)이 빈번하게 발생하게 되면 발생되는 빈도만큼 CPU가 동작하지

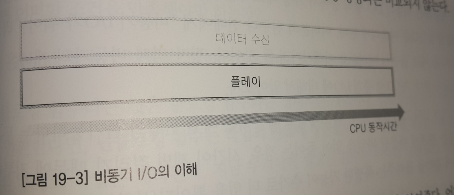
않고 성능의 저하로 이어지게 된다.

기본적으로 ANSI 표준함수를 기반으로 프로그램을 작성하게 되면 I/O 작업을 하게 될 시에 작업이

완료될 때까지 호출된 함수가 블로킹 상태 (함수 호출이 완료되지 않은 상태)에 놓이게 되고 작업이

완료 될 때까지 블로킹 되게 된다.

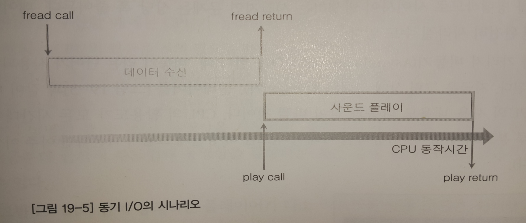
이 함수들을 활용한 입/출력 연산을 가리켜 동기 (Synchronous) I/O 라 표현한다.



이러한 CPU 효율의 저하를 막기위해 비동기 I/O가 있으며, 이러한 작업은 데이터 송/수신과 동시에 작업을 병행하게 되어 작업의 효율을 높이게 된다.

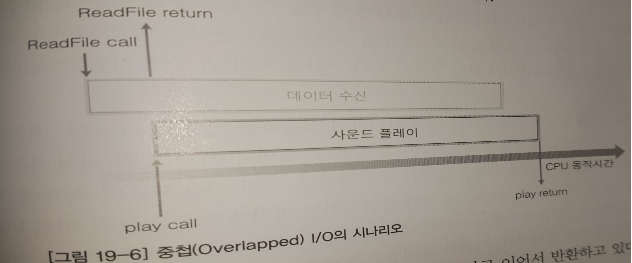
| 중첩 (Overlapped) I/O

Windows 에서 제공하는 비동기 I/O 방식 중 가장 대표적인 것이 중첩 I/O 이다.



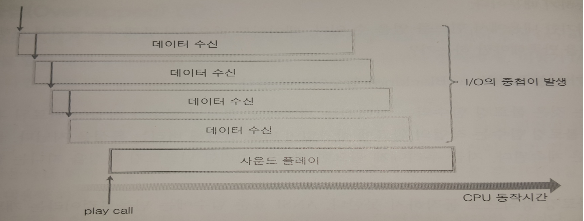
ANSI 표준함수인 fread 함수로 호출시 위와 같은 형태로 작업을 수행하게 된다. (블로킹 상태)

이러한 문제를 막기위해서는 다음과 같은 형태의 작업이 필요하다



Windows의 시스템 함수인 ReadFile에서는 I/O 작업을 넌블러킹 방식으로 작업을 수행한다.

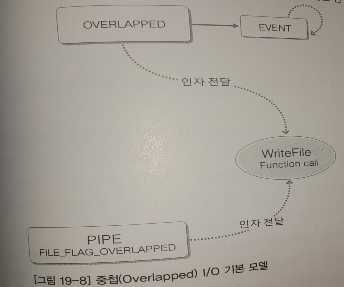
이러한 방식은 다음과 같은 방식으로 응용이 가능하다.



이러한 작업의 형태를 중첩 (Overlapped) I/O 라고 부른다.

| 중첩 (Overlapped) I/O 예제

파이프 통신을 사용하여 다음과 같은 형태로 구현된다.



[예제 namedpipe\_saynch\_server.cpp]

위 예제는 Overlapped를 통해 WriteFile 함수를 수행하며 이에 비동기로 작업을 수행한다.

작업에 대한 결과는 GetOverlappedResult 함수로 확인할 수 있으며, 비동기 작업이므로 I/O가 완료되지 않아도 바로

다음작업으로 작업 수행이 가능하다.

|  |
| --- |
| BOOL GetOverlappedResult(  HANDLE hFile,  LPOVERLAPPED lpOverlapped,  LPDWORD lpNumberOfBytesTransferred,  BOOL bWait  ); |

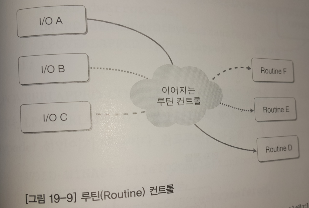
hFile - 확인할 파이프 핸들을 전달

lpOverlapped - WriteFile / ReadFile 호출 시 마지막 인자로 전달했던 OVERLAPPED 구조체 주소값 전달

lpNumberOfBytesTransferred - 실제 전송이 완료된 바이트의 크기를 얻기위한 변수 주소 전달

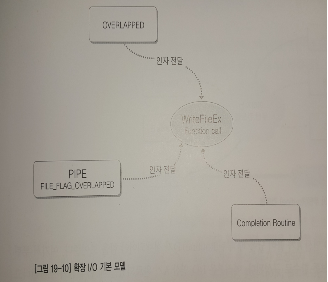
bWait - I/O 연산이 끝나지 않은 상황에서의 처리를 결정 (TRUE 완료 대기 FALSE 다음작업 수행)

| 완료 루틴 (Completion Routine) 기반 확장 I/O



위 방식과 같이 특정 연산이 끝났을때 루틴이 호출되는것을 완료 루틴이라고 정의한다.

이러한 방식을 확장하여 사용하는 방식은 다음과 같다.



특정 작업에 따라 루틴이 달라지는 방식을 확장 I/O 모델이라고 한다.

확장 I/O를 사용하는 두 함수는 다음과 같다.

|  |
| --- |
| BOOL WriteFileEx(  HANDLE hFile,  LPCVOID lpBuffer,  DWORD nNumberOfBytesToWrite,  LPOVERLAPPED lpOverlapped,  LPOVERLAPPED\_COMPLETION\_ROUTINE lpCompletionRoutine  );    BOOL ReadFileEx(  HANDLE hFile,  LPVOID lpBuffer,  DWORD nNumberOfBytesToRead,  LPOVERLAPPED lpOverlapped,  LPOVERLAPPED\_COMPLETION\_ROUTINE lpCompletionRoutine  );    VOID (WINAPI \*LPOVERLAPPED\_COMPLETION\_ROUTINE)(  DWORD dwErrorCode,  DWORD dwNumberOfBytesTransfered,  LPOVERLAPPED lpOverlapped  );    VOID WINAPI FileIOCompletionRoutine(  DWORD errorCode, DWORD numOfBytesTransfered, LPOVERLAPPED overlapped) |

hFile - 데이터 송/수신 대상자

lpBuffer - 데이터 송/수신 버퍼

nNumberOfBytesToRead - 데이터 송/수신 버퍼 크기

lpOverlapped - OVERLAPPED 구조체 변수 포인터 전달

lpCompletionRoutine - I/O 연산 완료시 자동으로 호출되는 완료루틴 지정

완료 루틴에 대한 함수 정의는 아래의 FileIOCompletionRoutine 와 같이 정의한다.

위 함수에 대한 사용예는 다음과 같다.

[예제 completion\_routine\_file.cpp]

| 알람 가능한 상태 (Alertable State)

SleepEx 함수는 완료루틴을 실행하라는 신호를 주는 용도로 사용된다.

|  |
| --- |
| DWORD SleepEx(  DWORD dwMilliseconds,  BOOL bAlertable  ); |

dwMilliseconds - ms 단위, 정지값

bAlertable - FALSE 전달시 일반 Sleep 함수와 동일, TRUE 전달시 이 함수가 호출된 스레드를 알림가능상태로

변경함

[예제 nonBlocking\_fileIO.cpp]

| 타이머에서의 완료루틴

SetWaitableTimer 함수를 통해 SleepEx에 타이머를 부여, 이를 통해 완료루틴을 수행할 수 있다.

|  |
| --- |
| BOOL SetWaitableTimer(  HANDLE hTimer,  const LARGE\_INTEGER \* lpDueTime,  LONG lPeriod,  PTIMERAPCROUTINE pfnCompletionRoutine,  LPVOID lpArgToCompletionRoutine,  BOOL fResume  );    typedef  VOID  (APIENTRY \*PTIMERAPCROUTINE)(  LPVOID lpArgToCompletionRoutine,  DWORD dwTimerLowValue,  DWORD dwTimerHighValue  ); |

pfncompletionroutine - 완료루틴을 지정

lpArgToCompletionRoutine - 타이머의 완료루틴 첫번째 천달인자로 그대로 전달

[예제 CompletionRoutinePeriodicTimer.cpp]

Session 02 APC (Asynchronous Procedure Call)

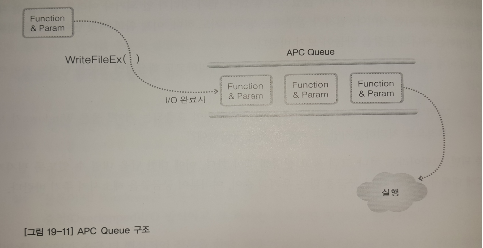
| APC의 구조

APC는 크게 두가지로 나뉜다.

User\_mode APC 그리고 Kernel\_mode APC이며

Kernel\_mode APC 는 다시 Nomal kernel\_mode APC / Special kernel\_mode APC로 나뉘게 된다.

Kernel\_mode APC는 해당 서적에서 다루지 않으므로 User 모드만 다룬다.



모든 스레드는 자신만의 APC Queue를 가지고 있고 위 그림과 같이 구성된다.

완료 루틴을 구성하는 도중 WriteFileEx 함수가 호출되며, 이때 인자로 전달된 함수(완료 루틴) 포인터와

매개변수 정보가 스레드의 APC Queue에 저장되고 있다.

APC Queue는 스레드별로 독립적이며, 각각의 스레드당 다른 APC Queue를 가지고 있다.

이러한 Queue는 비동기적으로 호출되어야할 함수들과 매개변수 정보들을 저장하게 되지만 바로 호출되지는

않으며, 스레드가 알림 가능상태 (Alertable State)에 놓이게 되어야 호출되게 된다.

| APC Queue의 접근

|  |
| --- |
| DWORD QueueUserAPC(  PAPCFUNC pfnAPC,  HANDLE hThread,  ULONG\_PTR dwData  ); |

pfnAPC - 비동기로 호출될 함수를 지정

hThread - 비동기 함수 정보를 추가할 APC Queue를 지정 (스레드 핸들을 지정)

dwData - APC Queue에 등록된 함수 호출시 전달할 인자를 지정

[예제 APCQueue.cpp]