

(2007년 7월 23일자 오타자 현황...한빛미디어)

2007-07-23 1

=====

오타자 확인(2007년 7월 23일자. 1쇄... 2쇄부터는 모두 반영되었습니다)

=====

-. 40페이지 [그림 1-10]에서
어셈블리 -> 어셈블러

-. 42페이지 [그림 1-11]에서
어셈블리 -> 어셈블러

-. 86페이지 [표 3-2]에서
LONG64 UINT Unsigned INT typedef unsigned int UINT
-> LONG64 Unsigned INT typedef unsigned int UINT

-. 147페이지.
#define _ttoi_wtoi -> #define _ttoi _wtoi
#define _ttoiatoi -> #define _ttoi atoi

-. 175페이지 끝에서 7번째 줄.
그렇다면 커널 오브젝트가 Windows 커널에 종속적이라고 표현할 수 있는 이유
-> 그렇다면 커널 오브젝트가 Windows 운영체제(커널)에 종속적이라고 표현할 수 있는 이유

-. 176페이지 '핸들의 종속 관계' 제목 바로 아래 줄.
지금까지 커널 오브젝트는 프로세스에 종속적이지 않고 커널 오브젝트에 종속적이라고
-> 지금까지 커널 오브젝트는 프로세스에 종속적이지 않고 운영체제에 종속적이라고

-. 186페이지 26행.

ste = CreateProcess(NULL, command,
-> CreateProcess(NULL, command,

"ste ="를 삭제해 주세요..

-. 188페이지 '커널 오브젝트와 Usage Count' 바로 다음 줄.
자식 프로세스의 종료코드는 어디에 저장되겠는가? 바로 커널 오브젝트에 저장된다.
-> 자식프로세스의 종료코드는 어디에 저장되겠는가? 바로 자식 프로세스의 커널 오브젝트
에 저장된다.

-. 195페이지 [그림 6-12]의 아래 그림에서.
UC = 2
-> UC = 1

-. 227페이지 [그림 7-5] 바로 아래 줄.
그렇다면 반대로 Signaled 상태에서 Non-Signaled 상태로 변경된 시점은 언제이겠는가?
아마도 종료된 프로세스가 다시 실행을 재개하면 Signaled 상태가 될 것이다.
-> 그렇다면 반대로 Signaled 상태에서 Non-Signaled 상태로 변경된 시점은 언제이겠는가?
아마도 종료된 프로세스가 다시 실행을 재개하면 Non-Signaled 상태가 될 것이다.

-. 240페이지. 세 번째 문단 세 번째 줄.
만약에 MSND이 없다면
-> 만약에 MSDN이 없다면

-. 326페이지 [그림 10-4]에서
굵은 글씨체로 "함수 fct1 호출 시" 라는 문장이 두 개 있는데, 두 번째 문장을 "함수 fct2 호
출 시"로 변경해야 함.

-. 430페이지 'Note' 바로 다음 줄.
뮤텍스는 열쇠에 비유되므로 누군가가 열쇠를 취득했을 때 Signaled 상태가 되거나,
취득한 열쇠를 반환했을 때 Signaled 상태가 될 것이다.

-> 뮤텍스는 열쇠에 비유되므로 누군가가 열쇠를 취득했을 때 Non-Signaled 상태가 되고, 취득한 열쇠를 반환했을 때 Signaled 상태가 될 것이다.

- . 431페이지 위에서 두 번째 줄.
반면에 뮤텍스를 반환할 때에는 다음 함수를 이용해서 반환하게 된다.
물론 뮤텍스는 다시 Non-Signaled 상태가 된다.
-> 반면에 뮤텍스를 반환할 때에는 다음 함수를 이용해서 반환하게 된다.
물론 뮤텍스는 다시 Signaled 상태가 된다.

- . 573페이지 33행.
DefinedException != 0x00<<28;
-> DefinedException |= 0x00<<28;

36행.
DefinedException != 0x00<<16;
-> DefinedException |= 0x00<<16;

===예제파일일도 ! 로 잘못되어 있음. 정정 요망===

- . 574페이지 39행 설명부분에서.
임의로 0x80 을 지정하였다.
-> 임의로 0x08 을 지정하였다.

- . 604페이지 44행.
소스코드 44행에 다음을 넣어주세요. 43, 45번째 줄과 줄을 맞춰서 넣어주세요
44. CloseHandle(hFile);

- . 741페이지 [테스트 3단계]에서.
이번에는 이 상태에서(테스트2단계까지 진행한 상태) 예제(DLL을 사용하는 예제SwapDll.cpp)
의
확장자를 .cpp로 다시 변경하자.
-> 이번에는 이 상태에서(테스트2단계까지 진행한 상태) 예제(DLL을 사용하는 예제DllTest.c)
의
확장자를 .cpp로 다시 변경하자.

1	41	(1장, p41, 밑에서 5번째 줄) 이해할 수 있는 바이어리 코드로 -> 이해할 수 있는 바이너리 코드로
1	41	(41P) 표1-1 아래로 4번째 줄에 바이어리 코드 -> 바이너리 코드
1	42	(42p 밑에서 8재줄) 명령어C->명령어C
1	46	(p.46 어드레스 버스 설명 단락 마지막 "메모리는 0x1024번지에 존재하는 4바 터 버스를 통해서 전달받게 된다." => "메모리는 0x1024번지에 존재하는 4바 터 버스를 통해서 전달하게 된다."

3	86	<p>(86p 표 3-2 밑에서 6번째)</p> <p>86p [표 3-2] 밑에서 6번째 "LONG64" d</p> <p>UNIT -> Unsigned INT</p> <p>Unsigned INT typedef unsigned int UN > typedef unsigned int UNIT</p>
3	86	<p>(86P)</p> <p>86페이지 Windows 자료형 표에서</p> <p>첫번째 필드 12번째 행에 LONG64가 이</p> <p>LONG64 -> UINT</p> <p>그리고 세번째 필드 11번째 행에 LONG</p> <p>typedef signed _int64 -> typedef si</p>
3	91	<p>(90p~91p소스 18, 20 행)</p> <p>_tprintf(_T("Position %d, %d n"), (UINT) _tprintf(_T("distance : %d n"), CalDistance((UINT)&val1, --> 총 3개의 %d를 %u로 변경</p> <p>_tprintf(_T("Position %u, %u n"), (UINT) _tprintf(_T("distance : %u n"), CalDistance((UINT)&val1</p>
3	91	<p>(91페이지 실행결과)</p> <p>distance : 12</p> <p>이 한줄을 다음과 같이 두 줄로 삽입되</p> <p>Position 1245024, 1245012 distance : 12</p>
3	99	<p>(99페이지 소스코드 10라인)</p> <p>int main(int argc, TCHAR * argv[])</p> <p>--></p> <p>int _tmain(int argc, TCHAR * argv[])</p>
5	138	<p>(p.138 그림 5-7)</p> <p>Running 상태 프로세스와 Ready 상태 : 는 MSWord의 화살표 방향이 반대가 도</p>
5	146	<p>(146p, 맨 아래에서 두번째.)</p> <p>gettchar -> getwchar</p>

5	153	(153page 하단 여기서 잠깐 마지막줄) 바람 -> 바람
6	176	(6장 176page 위에서 6번째) <구문> 지금까지 커널 오브젝트는 프로세스에 널 오브젝트에 종속적이라고 설명하였 <변경 내용> -> 커널 오브젝트에 종속적이라고 설명 위의 내용을 -> 운영체제에 종속적이라고 설명하였 으로 변경 되어야 됩니다
6	181	(181.쪽 예제 6-4 파일이름과 주석) 총 두 군데에 변경이 되어야 합니다 CloseHandleProb1.cpp -> KernelObjPr
6	183	(P.183) 예제 6-5는 CloseHandleProb2.cpp가 0 라 KernerlObjProb.cpp 입니다. 제목과 주석이 잘못되었습니다.
6	186	((186p 예제6-6 OperationStateParen 26행에서 ste = CreateProcess(NULL, command, CREATE_NEW_CONSOLE, f &si, &pi); 'ste =' 를 삭제합니다.
6	192	(192p. 여기서 잠깐 마지막에서 두번째 TermiateProcess -> TerminateProcess
6	195	(195쪽 [그림 6-12]) 계산기 kernel object UC = 2 ==> 계산기 kernel object UC = 1

6	202	<p>(202p 14번째줄 for문)</p> <p>이곳은 오타자가 아니고요 편집과정에 금 들어진것 입니다. 현재 코드가 아래처럼 되어 있습니다.</p> <pre>if(argc > 2) { for(int i=0; i<argc; i++) _tcscpy(cmdTokenList[i-1], argv[i]) CmdProcessing(argc-1); // 예를 한 위치로 밀어야 합니다. } // 예를 위의 {}와 동일한 위치 -></pre> <p>여기서 내용은 바꾸지 않고요 여백만 다 if(argc > 2) { for(int i=0; i<argc; i++) _tcscpy(cmdTokenList[i-1], argv[i]) CmdProcessing(argc-1); }</p>
7	221	<p>(221page, 소스 35번째라인)</p> <p>35행의 소스코드 시작 위치를 36행에</p>
8	252	<p>(252쪽)</p> <p>MainSender -> MainReceiver</p> <p>To Hanbit) 이전에 한번 등록된 내용입니</p>
8	252	<p>(p.252 1번째줄)</p> <p>MailSender는 변경할 필요가 없으므로 -> MailReceiver는 변경할 필요가 없으.</p>
8	282	<p>(282쪽 [예제8-10]에서 32라인)</p> <p>소스코드 32라인 삭제</p>
9	316	<p>(316쪽 밑에서 2번째줄)</p> <p>[그림 9-3]r과 -> [그림 9-3]과</p>
10	321	<p>(321쪽 위에서 5째줄)</p> <p>ATPCS(ARM Procedure Call Standard) -> ATPCS(ARM-Thumb Procedure Call Sta</p>
10	322	<p>(322page 밑에서 5번째줄)</p> <p>(First In, Fist Out)</p> <p>-></p> <p>(Last In, First Out)</p>
10	324	<p>(p.324 밑에서 9번째 줄)</p> <p>무엇인가를 선택할 때마다 더해 넣아가 를 선택할 때마다 더해 나아가는</p>

10	329	<p>(p329 [여섯 번째 단계] 4줄)</p> <p>네 번째 그리도 다섯 번째</p> <p>-> 네 번째 그리고 다섯 번째</p>
10	345	<p>(p345 표10-1)</p> <p>'32 bit' 항목의 첫 번째 라인, 'Stack clear Faller -> Caller 로 변경</p>
11	366	<p>(366p, 14번째 줄)</p> <p>일반적인 프로그램은</p> <p>-> 일반적인 프로그램은</p>
13	419	<p>(419쪽 크리티컬 섹션(Critical Section 두번째 줄)</p> <p>섹션 -> 섹션</p>
13	422	<p>(422쪽 03라인)</p> <p>프로그램 설명 : 생성 가능한 쓰레드의</p> <p>-></p> <p>프로그램 설명 : 임계영역의 동기화</p>
13	430	<p>(13장, p430, 밑에서 2번째 줄)</p> <p>EnterCirticalSection 함수와</p> <p>-> EnterCriticalSection 함수와</p>
13	433	<p>(433쪽, 48줄)</p> <p>// unnamed mutex 를 // nonamed mutex 로 변경해야 됨</p>
13	434	<p>(13장, p434, 밑에서 3번째 줄)</p> <p><21행 설명라인></p> <p>뮤텍스를 얻기 위해 WaitForSignleObjec 고 있다.</p> <p>-></p> <p>뮤텍스를 얻기 위해 WaitForSingleObjec 고 있다.</p> <p><밑에서 3번째 줄></p> <p>생각해 봐도, WaitForSignleObject 함수</p> <p>-> 생각해 봐도, WaitForSingleObject 함수</p>

13	443	<p>(p.443 첫 번째 줄)</p> <p>"음 그림 프로세스 B가~" -> "음 그림 프</p>
14	461	<p>(461쪽 24라인)</p> <p>FALSE, // auto reset mode로 생성</p> <p>-></p> <p>TRUE, // auto reset mode로 생성</p>
14	471	<p>(469p, 471p, SetWaitableTimer의 2번 명)</p> <p>469p 중간 부분</p> <p>② pDueTime : 알람이~~~ 10억 분의 1 위로 시간을 설정한다.</p> <p>이를</p> <p>② pDueTime : 알람이~~~ 1000만 분의 (100 Nanoseconds)단위로 시간을 설정</p> <p>471p 33행에 대한 소스 해설</p> <p>33행 : 두 번째 ~~ . 따라서 100000000 초 후에)</p> <p>이를</p> <p>33행 : 두 번째 ~~ . 따라서 1000000000 초 후에)</p>
14	473	<p>(473페이지 밑에서 4째줄)</p> <p>SetWaitableTimer -> CancelWaitableTi</p>
16	502	<p>(502쪽 10번째줄)</p> <p>에서는 -> 해서는</p>
16	508	<p>(16장 508페이지)</p> <p>여기서 잠깐 위에서 8번째 줄</p> <p>모두 틀리지 않은-> 모두 틀리지 않은</p>
16	511	<p>(511p 밑에서 9번째줄)</p> <p>발이다. 한 번은 값을 참조하기위해서,</p> <p>-></p> <p>말이다. 한 번은 값을 참조하기위해서,</p>

16	522	<p>(522p~523p)</p> <p>- 522페이지 마지막 줄 하위 8바이트를 통해서 -> 하위 12바이트</p> <p>- 523 페이지 첫 문장 마지막 8비트가 00000001이므로 -> 마 가 000000000001이므로</p> <p>- 523 페이지 두번째 문장 00000010, 00000011, 00000100번지까 > 000000000010, 000000000011, 0000</p>
17	531	<p>(531p 예외처리의 필요성 밑에 줄.)</p> <p>이어서 얘기하자면</p> <p>-></p> <p>이어서 말 하자면</p>
17	537	<p>(소스코드 17, 27 라인)</p> <p>17번 라인 _tscanf(_T("%d / %d", &a, &b)); ==> _tscanf(_T("%d / %d"), &a, &b);</p> <p>27번 라인 _tprintf(_T("result: %d n", a/b)); ==> _tprintf(_T("result: %d n"), a/b);</p>
17	558	<p>(558p 소스 43번째줄)</p> <p>43번째 라인의 }을 bold 처리!</p>
18	606	<p>(606페이지 위에서 2번째줄)</p> <p>lpFilePath ==> lpFilePart</p>
18	632	<p>(p632 두번째줄)</p> <p>가장 기본이 되는 입(출력 및 파일 생성 을 기억하자</p> <p>-></p> <p>가장 기본이 되는 입/출력 및 파일 생성 을 기억하자</p>
19	659	<p>(659쪽 17번째 줄)</p> <p>38번째 줄에서 -> 36번째 줄에서</p>
20	675	<p>(675p 밑에서 12번째줄 마지막 문장)</p> <p>RESEERVE 상</p> <p>을</p> <p>RESERVE 상</p>

20	681	(681p, 86번째 라인) _tprintf(_T("VirtualAlloc failed %d n")) 를 _tprintf(_T("VirtualAlloc failed n"));
20	694	(694 page, source code line 35) sizeof(TCHAR) * 10 -> sizeof(TCHAR) *
20	709	(709 page, source code line 61) for(int j = 1; j < num-1; j++) --> for(int j = 1; j < num; j++)
21	745	(745 page, 소스코드 25 line) GetProcAddress(hinstLib, _T("swap")); --> GetProcAddress(hinstLib, "swap");