Windows デバッグ講座

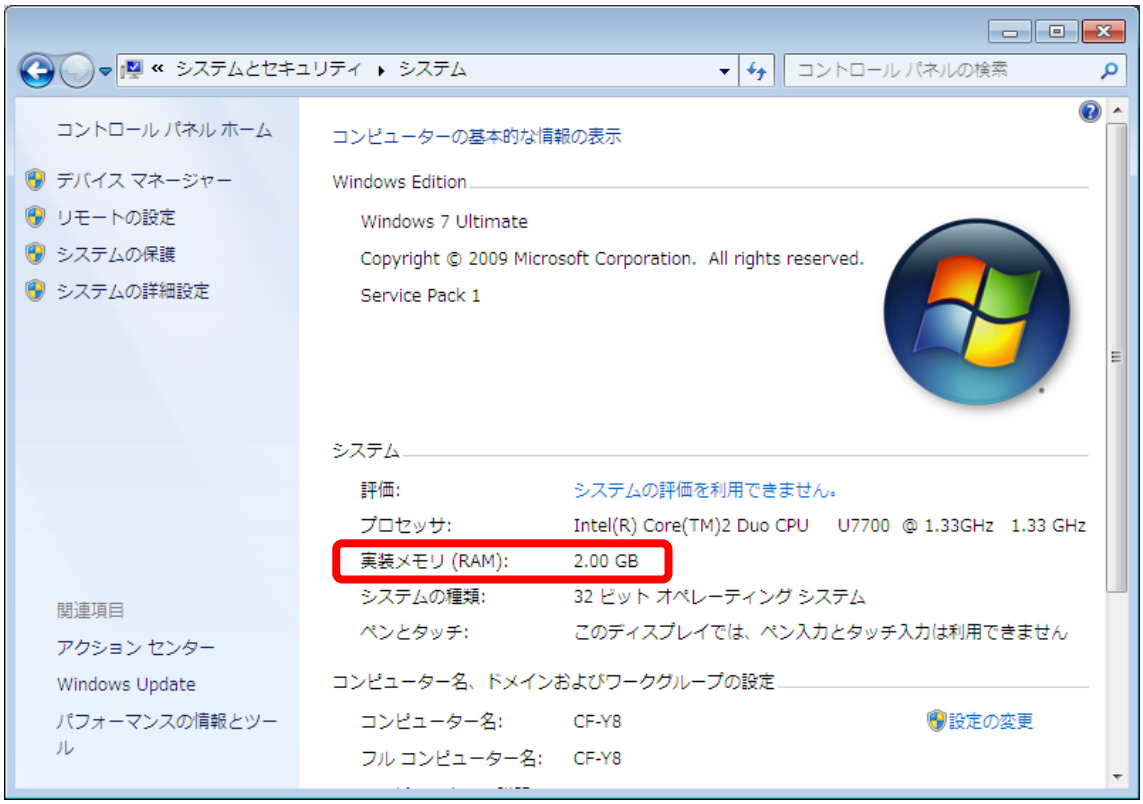
株式会社風太

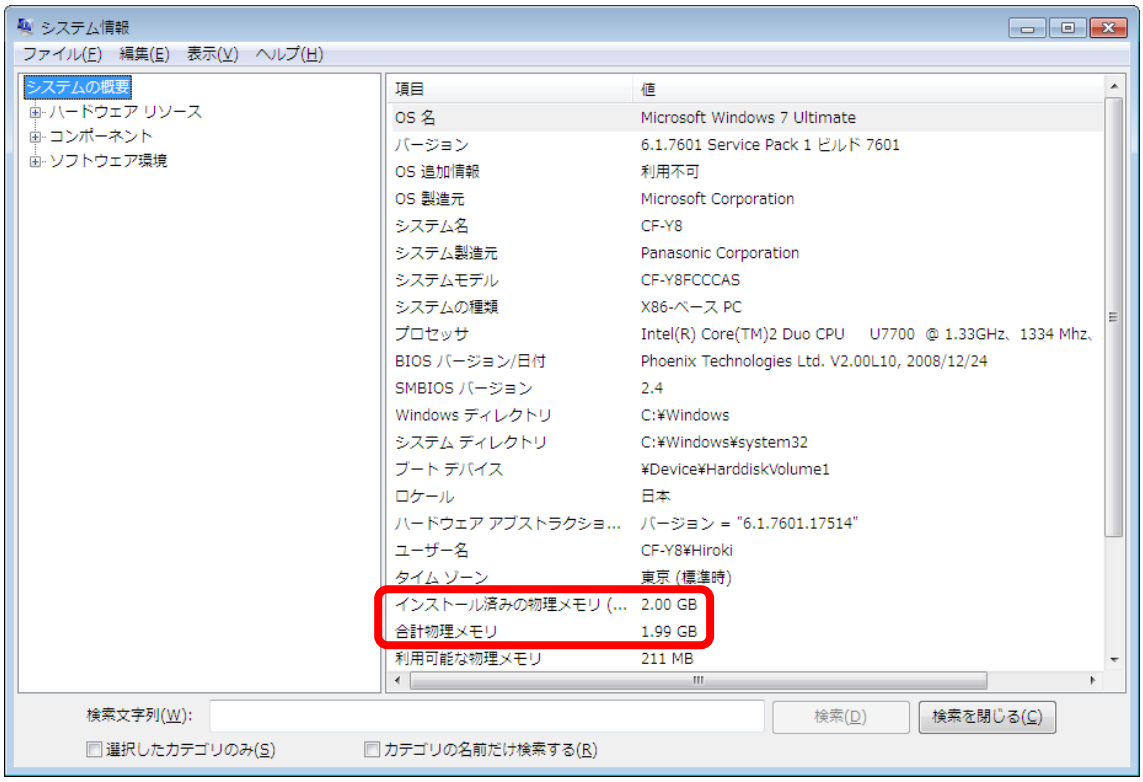
萩原 宏紀

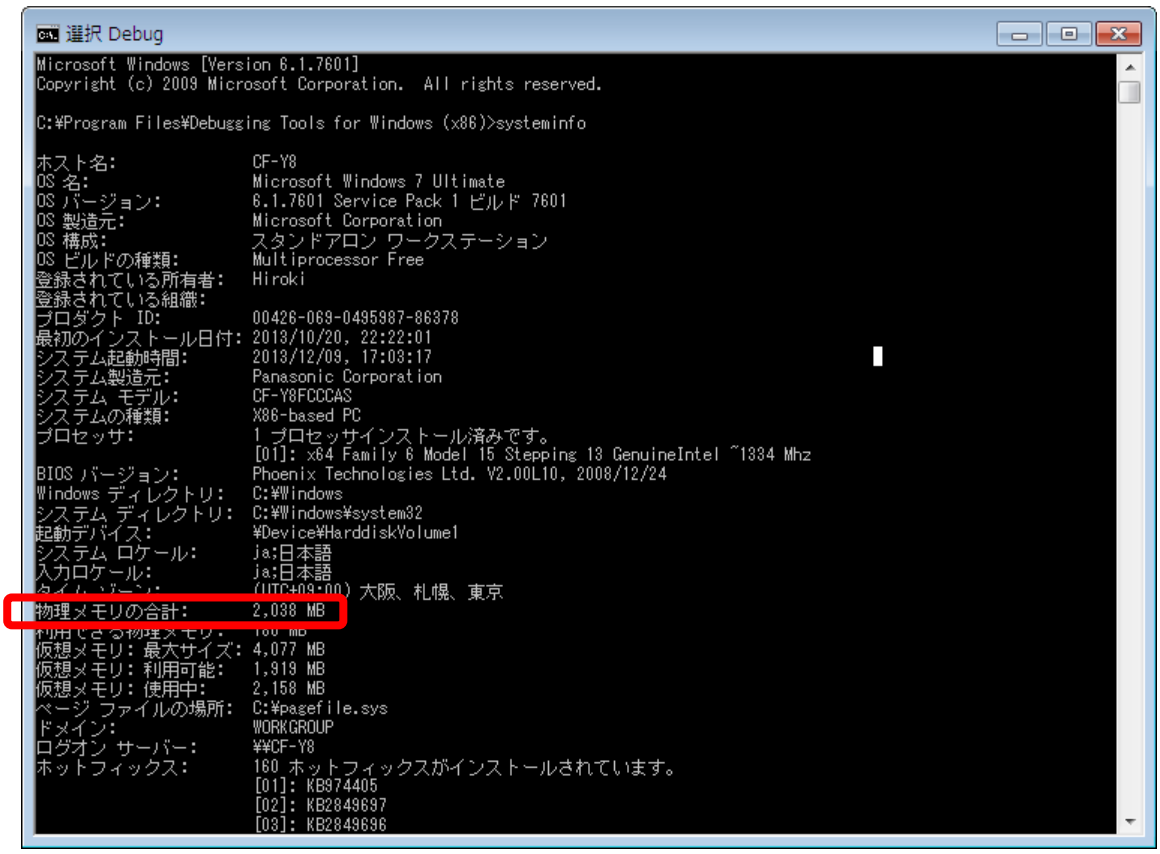
第２回「メモリ空間（その１）」

第１回では、ユーザーモードデバッガ cdb.exe によって、プロセスにアタッチおよびデタッチするところまでをお話ししました。第２回以降では、いよいよデバッガを使ってプロセスの内部を調査していきます。そのためには、メモリ空間についての知識が必要となってくるのですが、果たしてメモリ空間とは何でしょう？

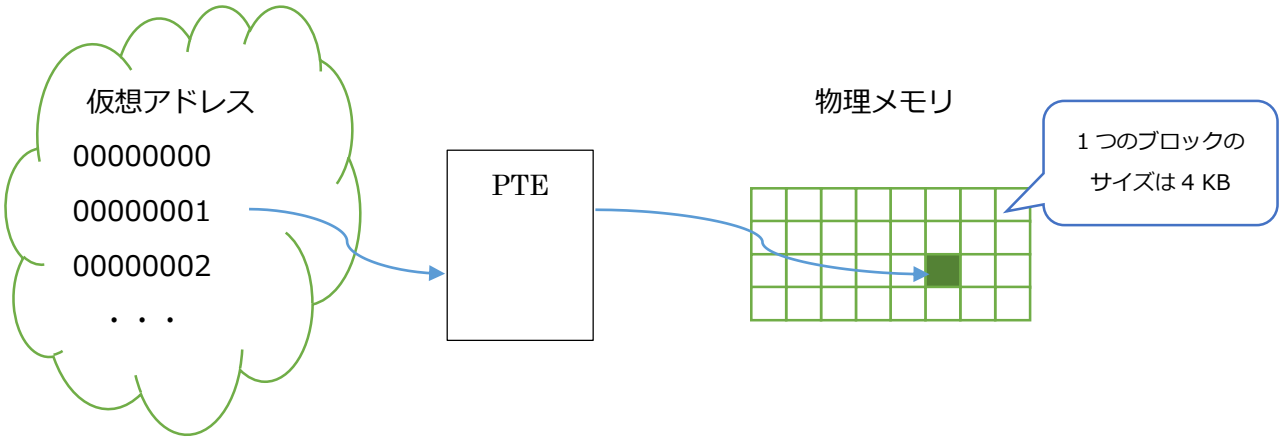
正確にはメモリアドレス空間と呼んだ方が正しいのですが、実際にパソコンに搭載されているメモリ（RAM）には容量が決まっています。ちなみに Windows 上で、搭載されている物理メモリを確認する方法はいくつかあります。コントロールパネルの [システムとセキュリティ] から [システム] を選択しても良いですし、[すべてのプログラム] から [アクセサリ] – [システム ツール] – [システム情報]を選択しても良いです。（コマンドプロンプトから ”msinfo32” でも同じ。）テキストベースのシステム情報はコマンドプロンプトから、“systeminfo” と入力して確認できます。




システム情報
ファイル(F) 編集(E) 表示(V) ヘルプ(H)
システムの概要
ハードウェア リソース
コンポーネント
ソフトウェア環境
項目
値
OS 名 Microsoft Windows 7 Ultimate
バージョン 6.1.7601 Service Pack 1 ビルド 7601
OS 追加情報 利用不可
OS 製造元 Microsoft Corporation
システム名 CF-Y8
システム製造元 Panasonic Corporation
システムモデル CF-Y8FCCCAS
システムの種類 X86-ベース PC
プロセッサ Intel(R) Core(TM)2 Duo CPU U7700 @ 1.33GHz、1334 Mhz、
BIOS バージョン/日付 Phoenix Technologies Ltd. V2.00L10, 2008/12/24
SMBIOS バージョン 2.4
Windows ディレクトリ C:¥Windows
システム ディレクトリ C:¥Windows¥system32
ブート デバイス ¥Device¥HarddiskVolume1
ロケール 日本
ハードウェア アブストラクショ... バージョン = "6.1.7601.17514"
ユーザー名 CF-Y8¥Hiroki
タイム ゾーン 東京 (標準時)
インストール済みの物理メモリ (... 2.00 GB
合計物理メモリ 1.99 GB
利用可能な物理メモリ 211 MB
検索文字列(W):
検索(D)
検索を閉じる(C)
選択したカテゴリのみ(S)
カテゴリの名前だけ検索する(R)


選択 Debug
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:¥Program Files¥Debugging Tools for Windows (x86)>systeminfo
ホスト名: CF-Y8
OS 名: Microsoft Windows 7 Ultimate
OS バージョン: 6.1.7601 Service Pack 1 ビルド 7601
OS 製造元: Microsoft Corporation
OS 構成: スタンドアロン ワークステーション
OS ビルドの種類: Multiprocessor Free
登録されている所有者: Hiroki
登録されている組織:
プロダクト ID: 00426-069-0495987-86378
最初のインストール日付: 2013/10/20, 22:22:01
システム起動時間: 2013/12/09, 17:03:17
システム製造元: Panasonic Corporation
システム モデル: CF-Y8FCCCAS
システムの種類: X86-based PC
プロセッサ: 1 プロセッサインストール済みです。
[01]: x64 Family 6 Model 15 Stepping 13 GenuineIntel ~1334 Mhz
BIOS バージョン: Phoenix Technologies Ltd. V2.00L10, 2008/12/24
Windows ディレクトリ: C:¥Windows
システム ディレクトリ: C:¥Windows¥system32
起動デバイス: ¥Device¥HarddiskVolume1
システム ロケール: ja;日本語
入力ロケール: ja;日本語
タイム ゾーン: (UTC+09:00) 大阪、札幌、東京
物理メモリの合計: 2,038 MB
仮想メモリ: 最大サイズ: 4,077 MB
仮想メモリ: 利用可能: 1,919 MB
仮想メモリ: 使用中: 2,158 MB
ページ ファイルの場所: C:¥pagefile.sys
ドメイン: WORKGROUP
ログオン サーバー: ¥¥CF-Y8
ホットフィックス: 160 ホットフィックスがインストールされています。
[01]: KB974405
[02]: KB2849697
[03]: KB2849696

今、私が使っているこのパソコンでは、2 GB の物理メモリを搭載しています。このメモリ上に OS がロードされ、アプリケーションプログラムがロードされているわけです。実際、昔のマイコン時代は、この物理メモリ上にプログラムが直接ロードされていました。もちろん、シングルタスクでしかプログラムを実行できなかった時代の話です。しかし、マルチタスクを実現させる上で、たくさんのアプリケーションプログラムを同時にメモリ上に読み込ませていたのでは、メモリがいくらあっても足りません。そこで、仮想メモリという考えが現れました。仮想メモリとは、物理メモリのどこに記録されるかという事は関係なく、仮想的にメモリアドレスを定義し、少ない物理メモリであっても、大きなメモリ空間を仮想的に利用する事が出来るという仕組みです。物理メモリと仮想メモリのアドレスマッピングは、OS が自動で行います。具体的には、ページテーブルエントリ（PTE）というメモリ領域にそのマッピング情報が配置されます。通常、メモリはページ単位（4 KB）で管理されるので、そのページのアドレス変換を OS が、ページテーブルエントリの情報に基づいて行います。

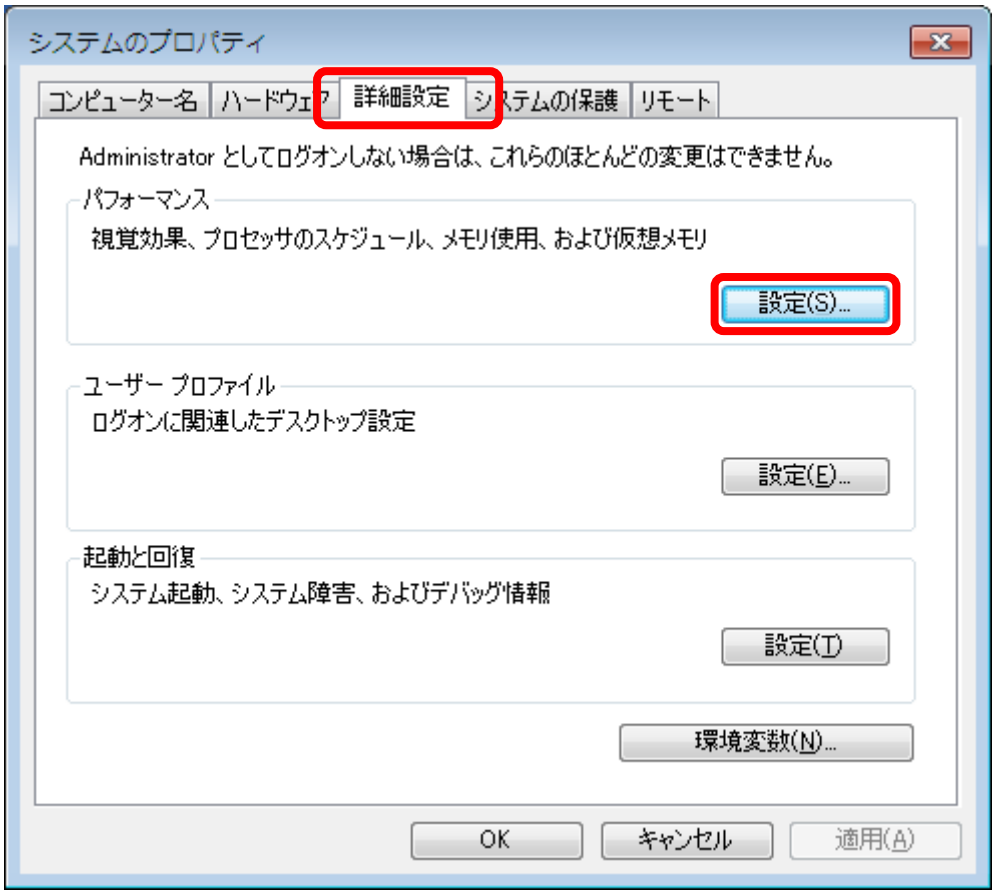


もう少し正確に言うと、ページディレクトリテーブルというのが存在して、そこにページテーブルのベースアドレスが格納されるのですが、ここでは割愛します。詳しくは、以下のサイトをご参照下さい。

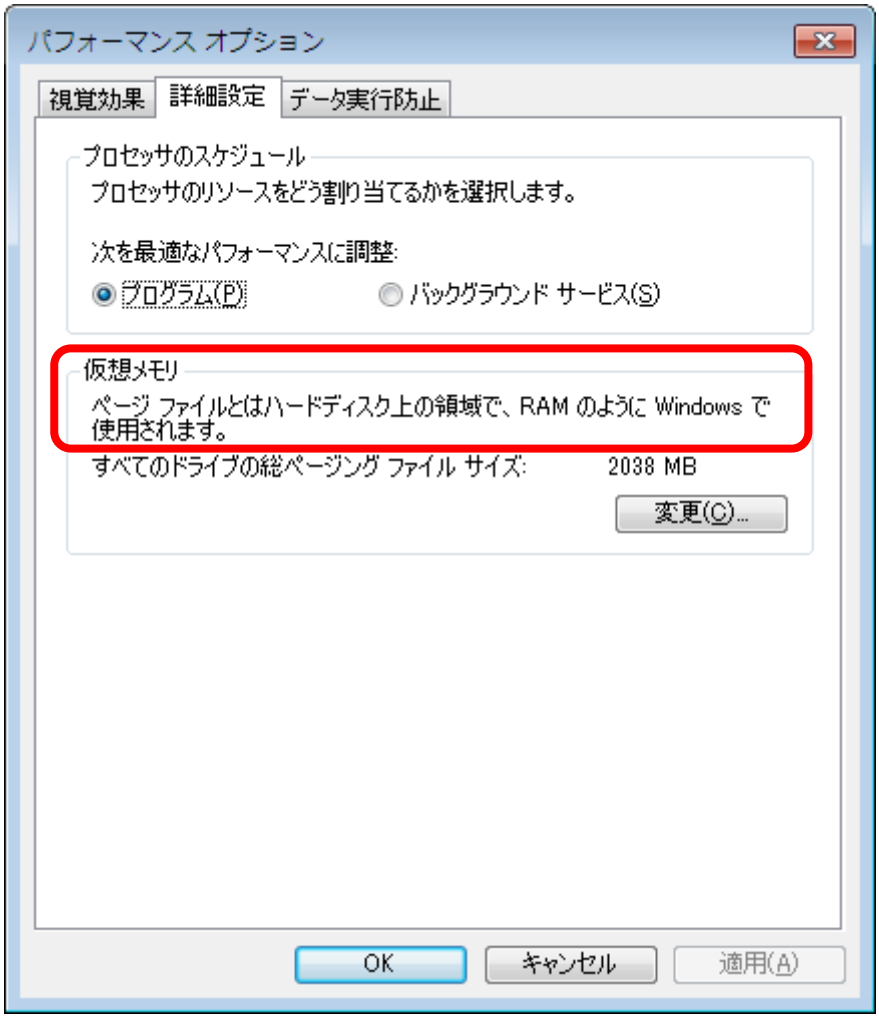
Windows のメモリー管理を x86 の仕組みから読み解く
http://ascii.jp/elem/000/000/649/649680/

なお、物理メモリ上を専有しているデータが使われなくなったら、一時的にスワッピングという行為によって、ハードディスクに退避しておき、必要となったら自動で読み込ませる事ができます。これは、ページファイルというファイルを利用するのですが、Windows の

ユーザーインターフェースには、ちょっと混乱を招きかねない説明があります。コントロールパネルの [システムとセキュリティ] – [システム] より [設定の変更] をクリックして、[システムのプロパティ] を開いてみて下さい。そこから、[詳細設定] タブをクリックし、[パフォーマンス] の [設定(S)…] ボタンをクリックします。

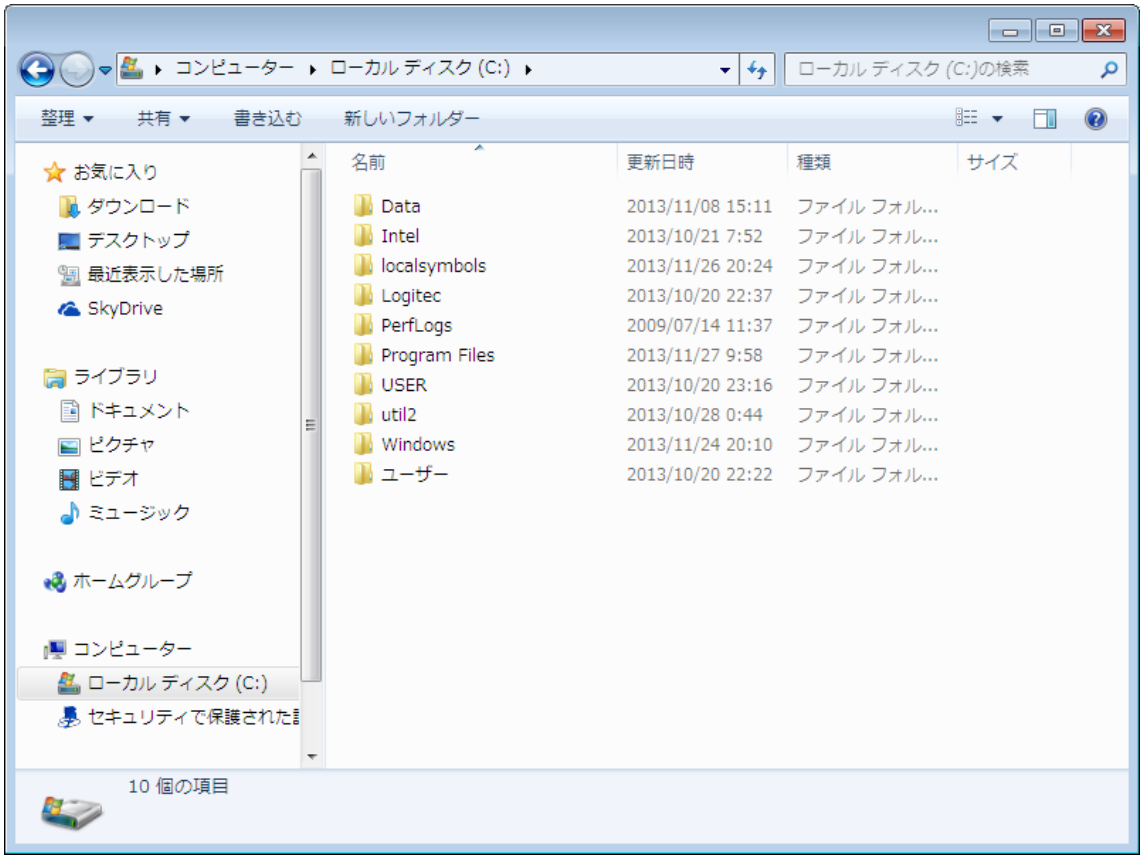


すると、次の画面が表示されます。

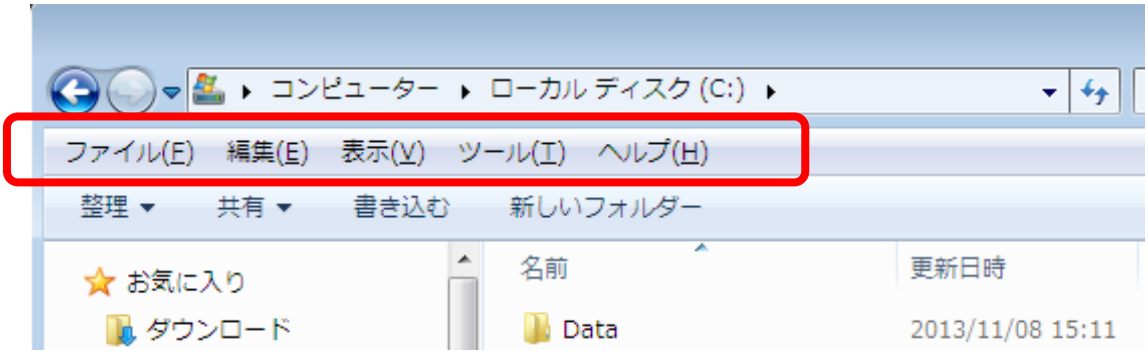


「仮想メモリ」と書かれている中に、「ページ ファイルとはハードディスク上の領域で、RAM のように Windows で使用されます。」との説明がありますよね。何のこっちゃ？という感じですね。

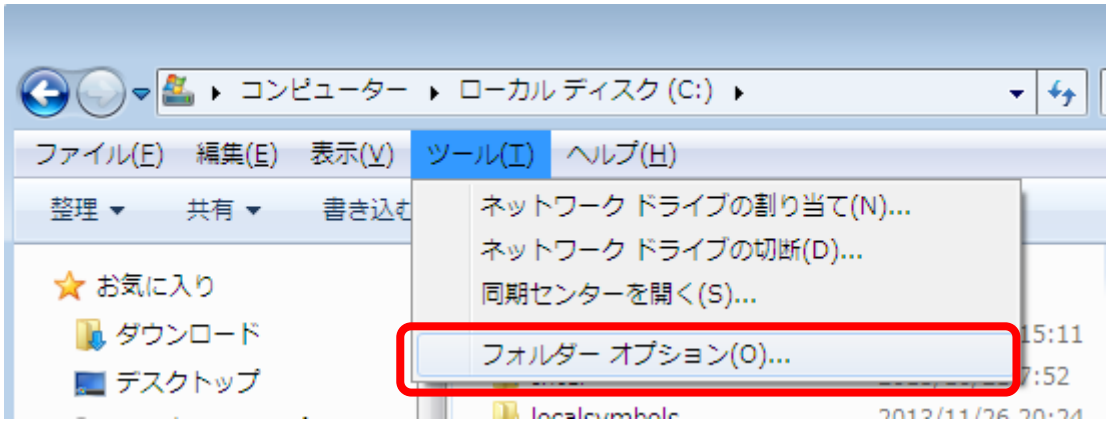
では、エクスプローラーを開いてみて下さい。[スタート] メニューから [コンピューター] をクリックして、[ローカル ディスク (C:)] をクリックして下さい。すると、次の様な画面になるかと思います。



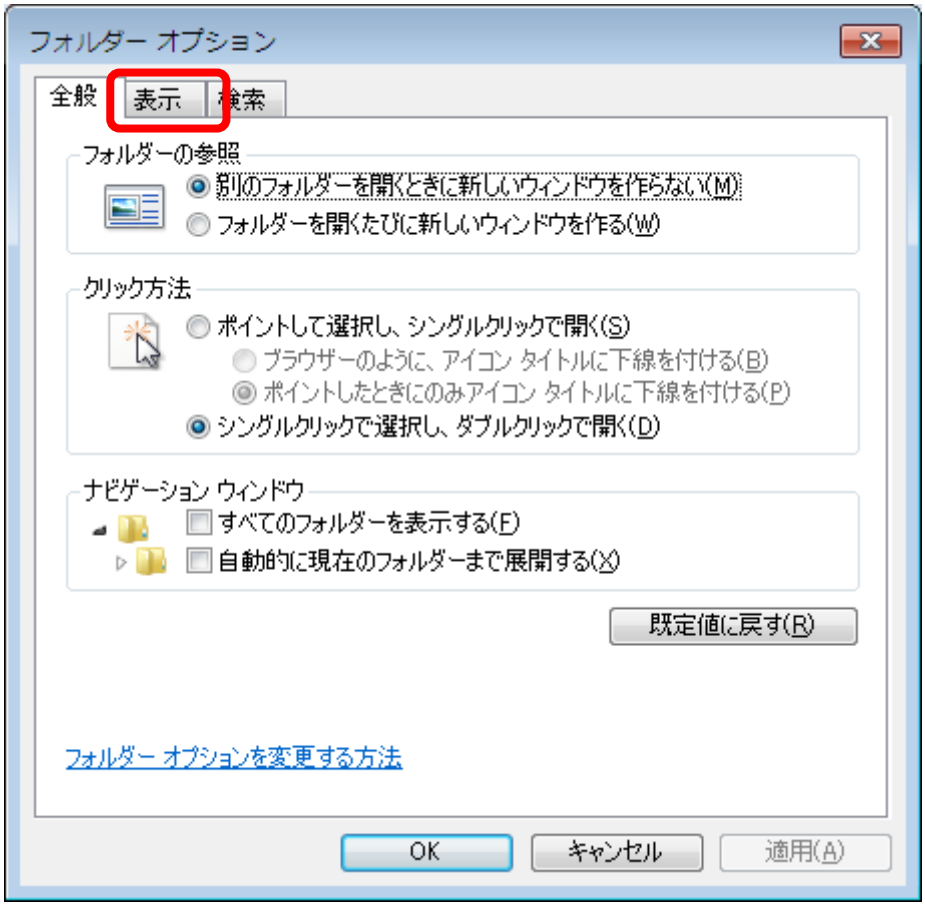
ここで、[Alt] キーを一回押してみて下さい。



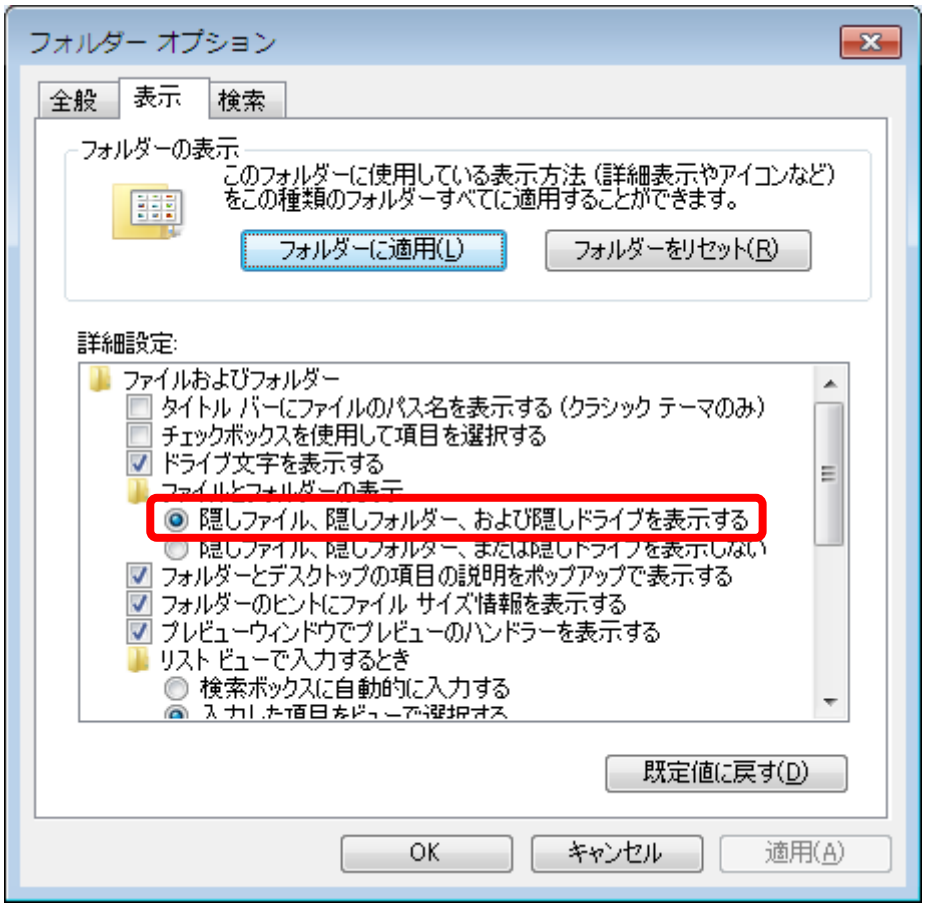
Windows XP の頃に表示されていたメニューバーが出現しました。では、[ツール(T)] をクリックして、[フォルダー オプション(O…)] をクリックして下さい。



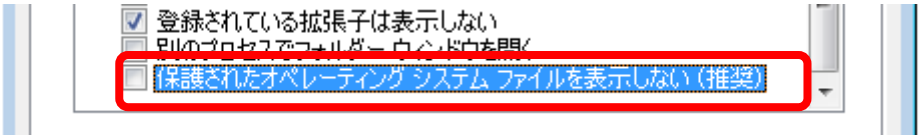
[フォルダー オプション(O…)] をクリックすると、次のダイアログが表示されます。[表示] タブをクリックして下さい。



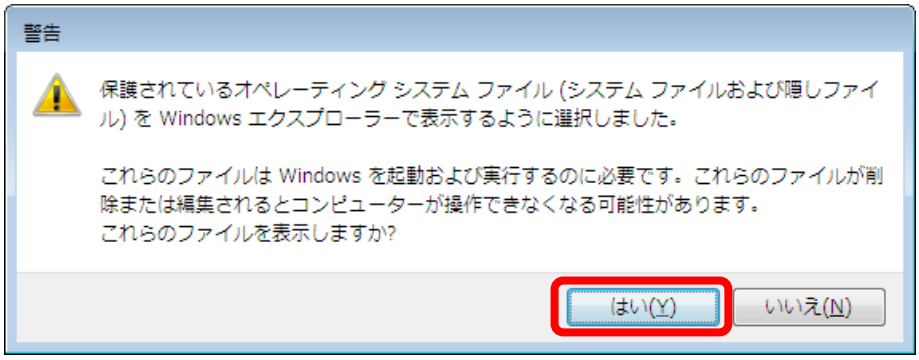
[表示] タブを開いたら、[ファイルとフォルダーの表示] 配下の [隠しファイル、隠しフォルダー、および隠しドライブを表示する] にチェックをします。



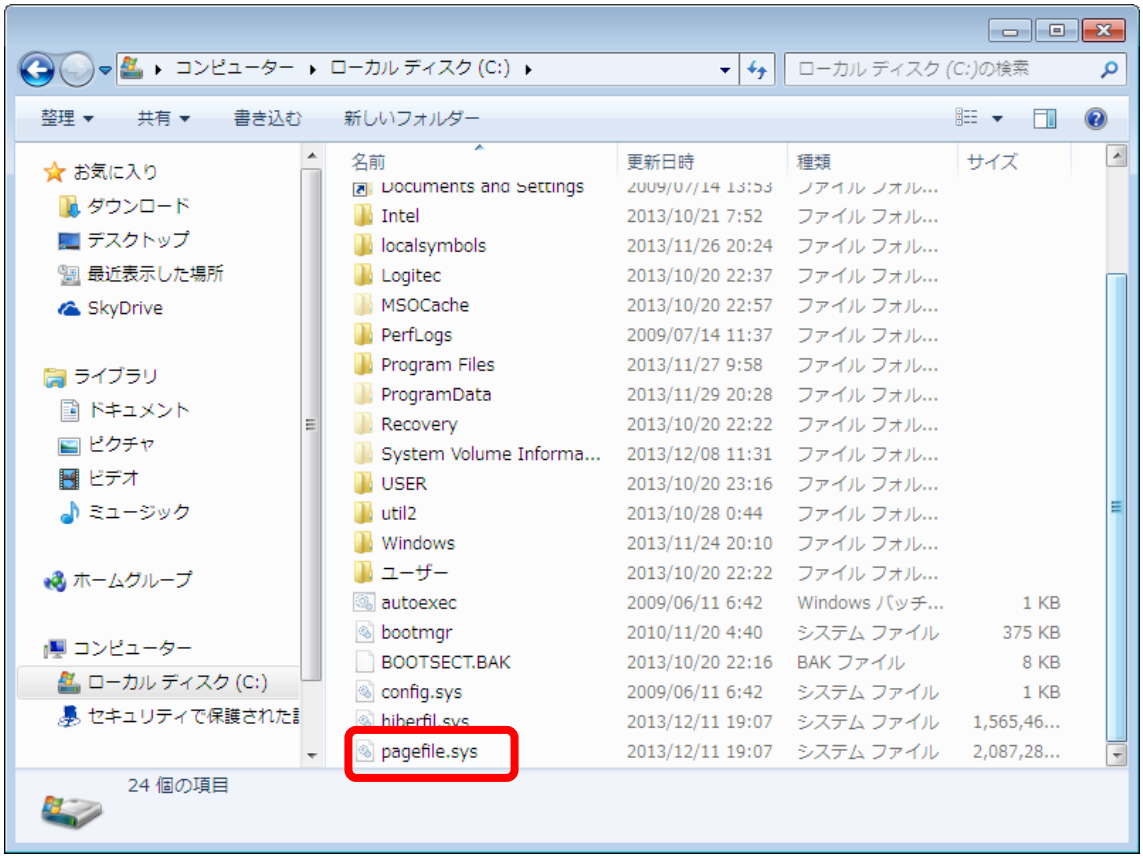
続いて、スクロールバーで [詳細設定:] を最後までスクロールさせ、[保護されたオペレーティング システム ファイルを表示しない（推奨）] のチェックを外します。



すると、次の警告が表示されます。



ここでは [はい(Y)]ボタンをクリックします。すると、隠しフォルダーや隠しファイルが表示され、c ドライブ直下には、pagefile.sys というファイルが表示されます。これが、ページファイルの実体です。



このページファイルには、物理メモリからスワップされたページが格納され、必要に応じて物理メモリ上に再読み込みされます。先ほどの「仮想メモリ」と書かれている中に、「ページ ファイルとはハードディスク上の領域で、RAM のように Windows で使用されます。」という説明がありましたが、言い換えると、ページファイル（pagefile.sys）はハード

ディスク上の１ファイルであり、そのファイルに物理メモリ上のページを退避させるので、ページファイルは、あたかも、使用頻度の少ない物理メモリの代わりの様に使われている仮想的なメモリ、ゆえに仮想メモリという箇所に説明が書かれていたわけです。しかし、仮想メモリ＝ページファイルではありません。ややこしいですね。

さて、話が少し脱線してしまいました。当初、「仮想メモリとは、物理メモリのどこに記録されるかという事は関係なく、仮想的にメモリアドレスを定義し、少ない物理メモリであっても、大きなメモリ空間を仮想的に利用する事が出来るという仕組み」というお話をしていましたが、何故かページファイルの説明になってしまいました。ただ、何の全く脈絡もなくページファイルの説明をしたわけではありません。ページファイルは、仮想メモリという仕組みを構成する１要素であり、仮にページファイルが無かったとすると、物理メモリからのスワップが行われませんので、物理メモリ上の古いページは不要になると完全に消去される事になります。そして再び必要になった場合、１から目的のファイルを読み込み直さなければなりません。しかし、ページファイルに退避した場合、仮想アドレスに基づいてページファイルに格納された古いページを参照するだけですので、少しアクセスの遅いメモリという感じで利用できるわけです。ゆえにページファイルは、仮想メモリの考え方である、大きなメモリ空間を仮想的に利用するために役立ちます。

ここでそろそろ話をメモリ空間に戻します。メモリ空間（メモリアドレス空間）とは、仮想メモリの仕組みによって定義された、仮想的なメモリアドレスで作られる空間を指します。つまり、アドレスの許す限りの空間を定義できるわけです。ここで、32 ビットのアドレス空間とは、32 ビットですので、2 進数でいうと 32 桁

0000 0000 0000 0000 0000 0000 0000 0000

から

1111 1111 1111 1111 1111 1111 1111 1111

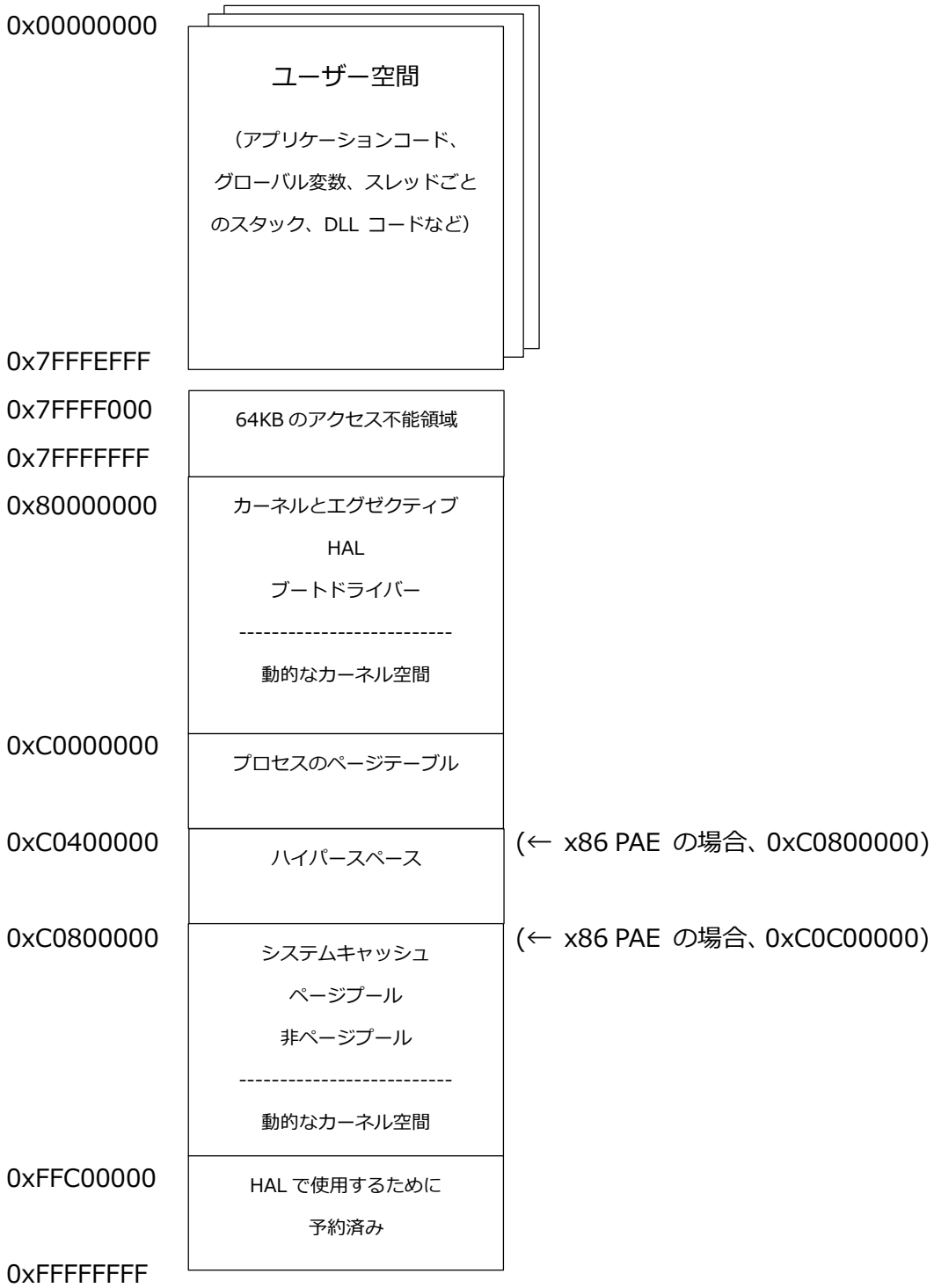
までが指定できます。16 進数に直すと、

0x00000000

から

0xFFFFFFFF

となります。つまり、x86 では、これだけの仮想的なメモリ空間を使う事が出来るわけです。ただし、どこでも自由に使えるわけではなく、Windows ではその領域がきちんと定義されています。例えば Windows 7 （x86 版）では、次の様に定義されています。



何だか、色々とわからない単語が出てきましたね。PAE というのは、Physical Address Extension の略で、例えば、サーバーマシンなどで 4 GB 以上の物理メモリを搭載していた場合に、4 GB を超える部分を有効に利用するための仕組みです。bcdedit コマンドのオ

プションに /set pae forceenable を指定する事で、PAE を有効にする事ができます。ただし、PAE を有効にしたからといって、必ずしも 4 GB 以上の物理メモリを利用できるというわけではなく、PAE 対応の API（AWE）を利用したアプリケーションでのみメリットを享受できます。この PAE スイッチを ON にして再起動すると、要はプロセスのページテーブルの領域が増えるわけです。つまり、ページテーブルエントリの領域が増える＝物理メモリへアクセス出来る数が増えるわけですね。ただし、その分、ページプールや非ページプールなどの領域が圧迫されてしまいます。

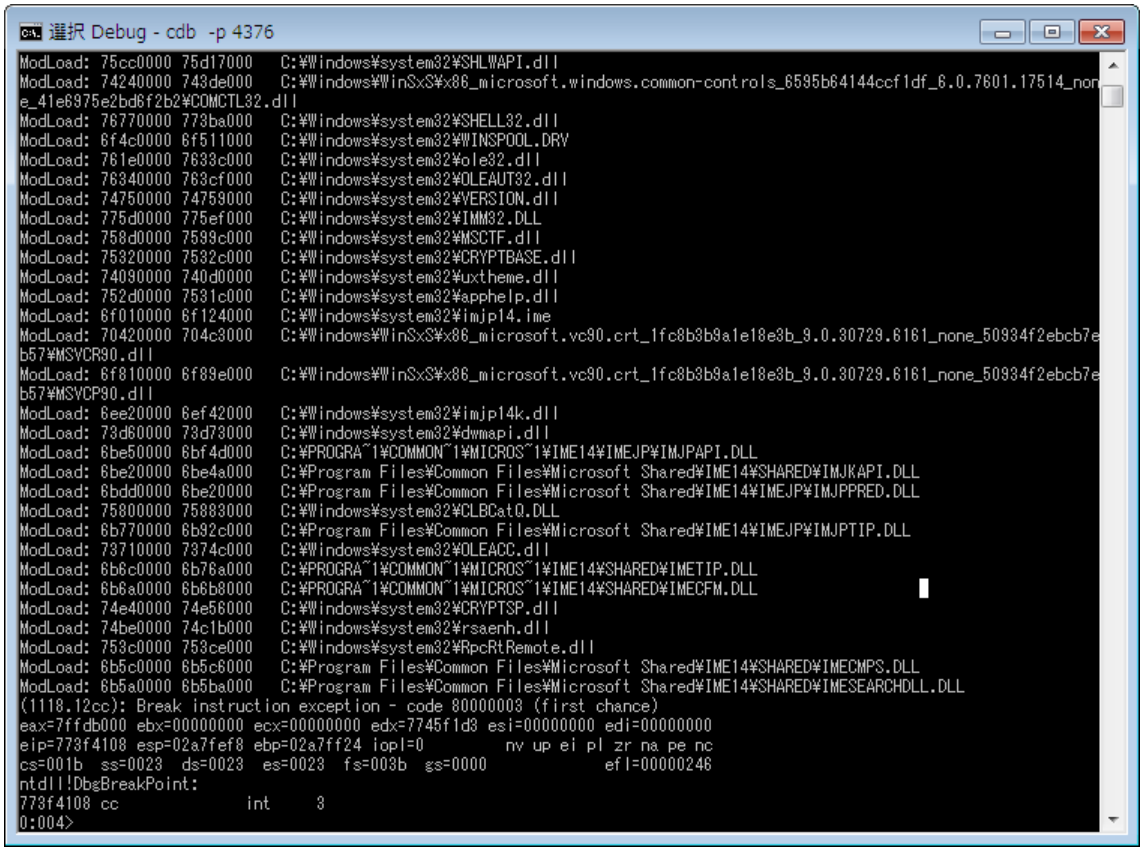
HAL とは Hardware Abstract Layer の略で、アプリケーションプログラムとハードウェアの間に介在し、ハードウェアの違いを吸収するソフトウェアです。実モジュールとしては、hal.dll というカーネルモードの DLL です。ntoskrnl.exe と hal.dll は、共に OS のコアイメージとして構成されており、HAL は ntoskrnl.exe とドライバーをハードウェアに接続する役割を果たしています。

ページプール、非ページプールは共に、カーネルモードで使用されるメモリ領域です。PagedPool 、Non-PagedPool などとも言います。

ここでは、それくらいを理解しておけば OK です。

あと、3GB スイッチというスイッチを有効にすると、ユーザー空間を 3 GB（0xBFFFFFFF まで） に拡張する事が出来ます。これは、広大なユーザー空間を必要とする、SQL Server などに有効です。ただし、カーネル空間を圧迫するため、メモリを多用する様なグラフィックドライバーなどが搭載されている場合、要注意です。

さて、長い前置きでしたが、それではデバッガで実際にプロセスにアタッチしてメモリ空間を見てみましょう。Notepad.exe を起動して、tlist で pID を調べ、cdb でアタッチですね。第 1 回で行った通りです。



では、”!peb”と入力してみて下さい。次の様に出力されるかと思います。

```
0:004> !peb
PEB at 7ffdf000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            Yes
    ImageBaseAddress:         00150000
    Ldr                       77497880
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 002e1ee8 . 002f9648
    Ldr.InLoadOrderModuleList:           002e1e58 . 002f9638
    Ldr.InMemoryOrderModuleList:         002e1e60 . 002f9640
            Base TimeStamp                     Module
          150000 4a5bc60f Jul 14 08:41:03 2009 C:¥Windows¥system32¥notepad.exe
        773c0000 521ea91c Aug 29 10:51:24 2013 C:¥Windows¥SYSTEM32¥ntdll.dll
        75f50000 51fb10c5 Aug 02 10:52:05 2013 C:¥Windows¥system32¥kernel32.dll
        756d0000 51fb10c6 Aug 02 10:52:06 2013 C:¥Windows¥system32¥KERNELBASE.dll
        765a0000 521ea86a Aug 29 10:48:26 2013 C:¥Windows¥system32¥ADVAPI32.dll
```

75c10000 4eeaf722 Dec 16 16:45:38 2011 C:¥Windows¥system32¥msvcrt.dll

77550000 4a5bdb04 Jul 14 10:10:28 2009 C:¥Windows¥SYSTEM32¥sechost.dll

76130000 51db96a4 Jul 09 13:50:44 2013 C:¥Windows¥system32¥RPCRT4.dll

760c0000 524ccf2f Oct 03 10:58:07 2013 C:¥Windows¥system32¥GDI32.dll

759a0000 4ce7ba26 Nov 20 21:08:06 2010 C:¥Windows¥system32¥USER32.dll

76110000 51b0158a Jun 06 13:52:26 2013 C:¥Windows¥system32¥LPK.dll

75730000 50adaddf Nov 22 13:45:19 2012 C:¥Windows¥system32¥USP10.dll

76040000 4ce7b82d Nov 20 20:59:41 2010 C:¥Windows¥system32¥COMDLG32.dll

75cc0000 4ce7b9e2 Nov 20 21:06:58 2010 C:¥Windows¥system32¥SHLWAPI.dll

74240000 4ce7b71c Nov 20 20:55:08 2010 C:¥Windows¥WinSxS¥x86_microsoft.windows.common-controls_6595b64144ccf1df_
6.0.7601.17514_none_41e6975e2bd6f2b2¥COMCTL32.dll

76770000 51f1d731 Jul 26 10:56:01 2013 C:¥Windows¥system32¥SHELL32.dll

6f4c0000 4ce7ba4b Nov 20 21:08:43 2010 C:¥Windows¥system32¥WINSPOOL.DRV

761e0000 4ce7b96f Nov 20 21:05:03 2010 C:¥Windows¥system32¥ole32.dll

76340000 4e58702a Aug 27 13:18:50 2011 C:¥Windows¥system32¥OLEAUT32.dll

74750000 4a5bdb2b Jul 14 10:11:07 2009 C:¥Windows¥system32¥VERSION.dll

775d0000 4ce7b845 Nov 20 21:00:05 2010 C:¥Windows¥system32¥IMM32.DLL

758d0000 4a5bda69 Jul 14 10:07:53 2009 C:¥Windows¥system32¥MSCTF.dll

75320000 4a5bbf41 Jul 14 08:12:01 2009 C:¥Windows¥system32¥CRYPTBASE.dll

74090000 4a5bdb38 Jul 14 10:11:20 2009 C:¥Windows¥system32¥uxtheme.dll

752d0000 4ce7b73e Nov 20 20:55:42 2010 C:¥Windows¥system32¥apphelp.dll

6f010000 4b580e8a Jan 21 17:21:30 2010 C:¥Windows¥system32¥imjp14.ime

70420000 4dace5b9 Apr 19 10:30:33 2011 C:¥Windows¥WinSxS¥x86_microsoft.vc90.crt_1fc8b3b9a1e18e3b_9.0.30729.6161_
none_50934f2ebcb7eb57¥MSVCR90.dll

6f810000 4dace5bd Apr 19 10:30:37 2011 C:¥Windows¥WinSxS¥x86_microsoft.vc90.crt_1fc8b3b9a1e18e3b_9.0.30729.6161_
none_50934f2ebcb7eb57¥MSVCP90.dll

6ee20000 50908d79 Oct 31 11:31:21 2012 C:¥Windows¥system32¥imjp14k.dll

73d60000 4a5bda07 Jul 14 10:06:15 2009 C:¥Windows¥system32¥dwmapi.dll

6be50000 50908d90 Oct 31 11:31:44 2012 C:¥PROGRA~1¥COMMON~1¥MICROS~1¥IME14¥IMEJP¥IMJPAPI.DLL

6be20000 50908d68 Oct 31 11:31:04 2012 C:¥Program Files¥Common Files¥Microsoft Shared¥IME14¥SHARED¥IMJKAPI.DLL

6bdd0000 4f442941 Feb 22 08:31:13 2012 C:¥Program Files¥Common Files¥Microsoft Shared¥IME14¥IMEJP¥IMJPPRED.DLL

75800000 4a5bd9b1 Jul 14 10:04:49 2009 C:¥Windows¥system32¥CLBCatQ.DLL

6b770000 506be31b Oct 03 16:02:51 2012 C:¥Program Files¥Common Files¥Microsoft Shared¥IME14¥IMEJP¥IMJPTIP.DLL

73710000 4e587028 Aug 27 13:18:48 2011 C:¥Windows¥system32¥OLEACC.dll

6b6c0000 50908d45 Oct 31 11:30:29 2012

C:¥PROGRA~1¥COMMON~1¥MICROS~1¥IME14¥SHARED¥IMETIP.DLL

6b6a0000 50908d17 Oct 31 11:29:43 2012 C:¥PROGRA~1¥COMMON~1¥MICROS~1¥IME14¥SHARED¥IMECFM.DLL

74e40000 4a5bda3d Jul 14 10:07:09 2009 C:¥Windows¥system32¥CRYPTSP.dll

74be0000 4a5bdae0 Jul 14 10:09:52 2009 C:¥Windows¥system32¥rsaenh.dll

753c0000 4ce7992f Nov 20 18:47:27 2010 C:¥Windows¥system32¥RpcRtRemote.dll

6b5c0000 4b580e0d Jan 21 17:19:25 2010 C:¥Program Files¥Common Files¥Microsoft Shared¥IME14¥SHARED¥IMECMPS.DLL

6b5a0000 4b580e49 Jan 21 17:20:25 2010 C:¥Program Files¥Common Files¥Microsoft Shared¥IME14¥SHARED¥IMESEARCHDLL.

DLL

SubSystemData: 00000000

ProcessHeap: 002e0000

ProcessParameters: 002e1488

CurrentDirectory: 'C:¥Users¥Hiroki¥'

WindowTitle: 'C:¥Windows¥system32¥notepad.exe'

ImageFile: 'C:¥Windows¥system32¥notepad.exe'

CommandLine: '"C:¥Windows¥system32¥notepad.exe" '

DllPath:

'C:¥Windows¥system32;;C:¥Windows¥system32;C:¥Windows¥system;C:¥Windows;.;C:¥Program Files¥Common Files

¥Microsoft Shared¥Windows Live;C:¥Windows¥system32;C:¥Windows;C:¥Windows¥System32¥Wbem;C:¥Windows¥System32¥WindowsPowerS

hell¥v1.0¥;C:¥Program Files¥Windows Live¥Shared;C:¥Program Files¥QuickTime¥QTSystem¥;C:¥Program Files¥Microsoft Windows

Performance Toolkit¥;C:¥Program Files¥Windows Kits¥8.1¥Windows Performance Toolkit¥;C:¥Program Files¥Microsoft SQL Serve

r¥110¥Tools¥Binn¥'

Environment: 002e07f0

=::=::¥

ALLUSERSPROFILE=C:¥ProgramData

APPDATA=C:¥Users¥Hiroki¥AppData¥Roaming

asl.log=Destination=file

CLASSPATH=.;C:¥Program Files¥QuickTime¥QTSystem¥QTJava.zip

CommonProgramFiles=C:¥Program Files¥Common Files

COMPUTERNAME=CF-Y8

ComSpec=C:¥Windows¥system32¥cmd.exe

FP_NO_HOST_CHECK=NO

HOMEDRIVE=C:

HOMEPATH=¥Users¥Hiroki

LOCALAPPDATA=C:¥Users¥Hiroki¥AppData¥Local

LOGONSERVER=¥¥CF-Y8
NUMBER_OF_PROCESSORS=2
OS=Windows_NT
Path=C:¥Program Files¥Common Files¥Microsoft Shared¥Windows Live;C:¥Windows¥system32;C:¥Windows;C:¥Windows¥System32¥Wbem;C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥;C:¥Program Files¥Windows Live¥Shared;C:¥Program Files¥QuickTime¥QTSystem¥;C:¥Program Files¥Microsoft Windows Performance Toolkit¥;C:¥Program Files¥Windows Kits¥8.1¥Windows Performance Toolkit¥;C:¥Program Files¥Microsoft SQL Server¥110¥Tools¥Binn¥
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 15 Stepping 13, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=0f0d
ProgramData=C:¥ProgramData
ProgramFiles=C:¥Program Files
PSModulePath=C:¥Windows¥system32¥WindowsPowerShell¥v1.0¥Modules¥
PUBLIC=C:¥Users¥Public
QTJAVA=C:¥Program Files¥QuickTime¥QTSystem¥QTJava.zip
SESSIONNAME=Console
SystemDrive=C:
SystemRoot=C:¥Windows
TEMP=C:¥Users¥Hiroki¥AppData¥Local¥Temp
TMP=C:¥Users¥Hiroki¥AppData¥Local¥Temp
USERDOMAIN=CF-Y8
USERNAME=Hiroki
USERPROFILE=C:¥Users¥Hiroki
VS120COMNTOOLS=C:¥Program Files¥Microsoft Visual Studio 12.0¥Common7¥Tools¥
windir=C:¥Windows
_NT_SYMBOL_PATH=SRV*c:¥localsymbols*http://msdl.microsoft.com/download/symbols

この、”!peb” というコマンドは、Process Environment Block(PEB) を表示せよというコマンドです。この notepad.exe というプロセスに関する様々な情報がこの PEB（プロセス環境ブロック）に格納されています。

さて、

Base TimeStamp Module
150000 4a5bc60f Jul 14 08:41:03 2009 C:¥Windows¥system32¥notepad.exe
773c0000 521ea91c Aug 29 10:51:24 2013 C:¥Windows¥SYSTEM32¥ntdll.dll

```
75f50000 51fb10c5 Aug 02 10:52:05 2013 C:¥Windows¥system32¥kernel32.dll
756d0000 51fb10c6 Aug 02 10:52:06 2013 C:¥Windows¥system32¥KERNELBASE.dll
                        ・・・
```

という箇所に着目して下さい。これは、0x00150000 というアドレスから notepad.exe のモジュールが読み込まれている事を示しています。0x773c0000 というアドレスからは ntdll.dll、0x75f50000 からは kernel32.dll、0x756d0000 からは KERNELBASE.dll が読み込まれていると書かれています。

では、どんな事が書かれているのか、実際に中身を見てみましょう。ここでは、dc コマンドを使います。

```
0:004> dc 150000
00150000  00905a4d 00000003 00000004 0000ffff  MZ..............
00150010  000000b8 00000000 00000040 00000000  ........@.......
00150020  00000000 00000000 00000000 00000000  ................
00150030  00000000 00000000 00000000 000000e0  ................
00150040  0eba1f0e cd09b400 4c01b821 685421cd  ........!..L.!Th
00150050  70207369 72676f72 63206d61 6f6e6e61  is program canno
00150060  65622074 6e757220 206e6920 20534f44  t be run in DOS
00150070  65646f6d 0a0d0d2e 00000024 00000000  mode....$.......
```

dd コマンドでもダンプの出力は出来ますが、右側にキャラクタが表示されますので、私は dc コマンドを愛用しています。よく見ると、”This is program cannot be run is DOS mode” などと書かれているのがわかります。何となく、プログラムの冒頭の様な気がしますね。もう少し後ろも見てみましょう。今度は d コマンドを打つだけで、続きが表示されます。

```
0:004> d
00150080  62c2beb2 31acdff6 31acdff6 31acdff6  ...b...1...1...1
00150090  3139a7ff 31acdff5 313fa7ff 31acdfeb  ..91...1..?1...1
001500a0  31addff6 31acdf00 312fa7ff 31acdfe9  ...1...1../1...1
001500b0  3128a7ff 31acdff4 3138a7ff 31acdff7  ..(1...1..81...1
001500c0  313da7ff 31acdff7 68636952 31acdff6  ..=1...1Rich...1
001500d0  00000000 00000000 00000000 00000000  ................
001500e0  00004550 0004014c 4a5bc60f 00000000  PE..L.....[J....
001500f0  00000000 010200e0 0009010b 0000a800  ................
```

ここは何だかわかりませんね。では、さらに下を見ていきましょう。今度は [Enter] キーを押すだけで、勝手に d コマンドを繰り返してくれます。([Enter] キーは、直前のコマンドを繰り返します。)

```
0:004>
00150100   00022400 00000000 00003689 00001000   .$.......6......
00150110   0000c000 00150000 00001000 00000200   ................
00150120   00010006 00010006 00010006 00000000   ................
00150130   00030000 00000400 00039741 81400002   ........A.....@.
00150140   00040000 00011000 00100000 00001000   ................
00150150   00000000 00000010 00000000 00000000   ................
00150160   0000a048 0000012c 0000f000 0001f160   H...,.......`...
00150170   00000000 00000000 00000000 00000000   ................
0:004>
00150180   0002f000 00000e34 0000b62c 00000038   ....4...,...8...
00150190   00000000 00000000 00000000 00000000   ................
001501a0   00000000 00000000 00006d58 00000040   ........Xm..@...
001501b0   00000278 00000128 00001000 00000400   x...(...........
001501c0   00000000 00000000 00000000 00000000   ................
001501d0   00000000 00000000 7865742e 00000074   ........text...
001501e0   0000a68c 00001000 0000a800 00000400   ................
001501f0   00000000 00000000 00000000 60000020   ............ ..`
0:004>
00150200   7461642e 00000061 00002164 0000c000   .data...d!......
00150210   00001000 0000ac00 00000000 00000000   ................
00150220   00000000 c0000040 7273722e 00000063   ....@....rsrc...
00150230   0001f160 0000f000 0001f200 0000bc00   `...............
00150240   00000000 00000000 00000000 40000040   ............@..@
00150250   6c65722e 0000636f 00000e34 0002f000   .reloc..4.......
00150260   00001000 0002ae00 00000000 00000000   ................
00150270   00000000 42000040 4a5bd97e 00000080   ....@..B~.[J....
```

text とか data といった単語も見えますね。

もっと先に進みましょう。

・・・(中略)

```
0:004>
00152c80   5ffc4d8b 33c38b5e d5e85bcd c9ffffe8   .M._^..3.[......
00152c90   83000cc2 e175fff8 8d57e6eb fff83885   ......u...W..8..
```

```
00152ca0   ff5650ff 1510d415 68f88b00 00000104   .PV........h....
00152cb0   fcf4858d 8350ffff 1a74ffff f864858d   ......P...t...d.
00152cc0   ff50ffff 15135815 15ff5700 001510d8   ..P..X...W......
00152cd0   f7e3e95f ff56ffff 15135815 90f1eb00   _.....V..X......
00152ce0   00500069 0069006f 0074006e 00690053   i.P.o.i.n.t.S.i.
00152cf0   0065007a 90900000 0066006c 00610046   z.e.....l.f.F.a.
0:004>
00152d00   00650063 0061004e 0065006d 90900000   c.e.N.a.m.e.....
00152d10   0066006c 00690050 00630074 00410068   l.f.P.i.t.c.h.A.
00152d20   0064006e 00610046 0069006d 0079006c   n.d.F.a.m.i.l.y.
00152d30   90900000 0066006c 00750051 006c0061   ....l.f.Q.u.a.l.
00152d40   00740069 00000079 0066006c 006c0043   i.t.y...l.f.C.l.
00152d50   00700069 00720050 00630065 00730069   i.p.P.r.e.c.i.s.
00152d60   006f0069 0000006e 0066006c 0075004f   i.o.n...l.f.O.u.
00152d70   00500074 00650072 00690063 00690073   t.P.r.e.c.i.s.i.
0:004>
00152d80   006e006f 90900000 0066006c 00680043   o.n.....l.f.C.h.
00152d90   00720061 00650053 00000074 0066006c   a.r.S.e.t...l.f.
00152da0   00740053 00690072 0065006b 0075004f   S.t.r.i.k.e.O.u.
00152db0   00000074 0066006c 006e0055 00650064   t...l.f.U.n.d.e.
00152dc0   006c0072 006e0069 00000065 0066006c   r.l.i.n.e...l.f.
00152dd0   00740049 006c0061 00630069 90900000   I.t.a.l.i.c.....
00152de0   0066006c 00650057 00670069 00740068   l.f.W.e.i.g.h.t.
00152df0   90900000 0066006c 0072004f 00650069   ....l.f.O.r.i.e.
0:004>
00152e00   0074006e 00740061 006f0069 0000006e   n.t.a.t.i.o.n...
00152e10   0066006c 00730045 00610063 00650070   l.f.E.s.c.a.p.e.
00152e20   0065006d 0074006e 90900000 006f0053   m.e.n.t.....S.o.
00152e30   00740066 00610077 00650072 004d005c   f.t.w.a.r.e.¥.M.
00152e40   00630069 006f0072 006f0073 00740066   i.c.r.o.s.o.f.t.
00152e50   004e005c 0074006f 00700065 00640061   ¥.N.o.t.e.p.a.d.
00152e60   90900000 00570069 006e0069 006f0064   ....i.W.i.n.d.o.
00152e70   00500077 0073006f 00590044 90900000   w.P.o.s.D.Y.....
0:004>
00152e80   00570069 006e0069 006f0064 00500077   i.W.i.n.d.o.w.P.
00152e90   0073006f 00580044 90900000 00570069   o.s.D.X.....i.W.
00152ea0   006e0069 006f0064 00500077 0073006f   i.n.d.o.w.P.o.s.
00152eb0   00000059 00570069 006e0069 006f0064   Y...i.W.i.n.d.o.
00152ec0   00500077 0073006f 00000058 004d0069   w.P.o.s.X...i.M.
00152ed0   00720061 00690067 0052006e 00670069   a.r.g.i.n.R.i.g.
00152ee0   00740068 90900000 004d0069 00720061   h.t.....i.M.a.r.
00152ef0   00690067 004c006e 00660065 00000074   g.i.n.L.e.f.t...
```

```
0:004>
00152f00   004d0069 00720061 00690067 0042006e   i.M.a.r.g.i.n.B.
00152f10   0074006f 006f0074 0000006d 004d0069   o.t.t.o.m...i.M.
00152f20   00720061 00690067 0054006e 0070006f   a.r.g.i.n.T.o.p.
00152f30   90900000 007a0073 00720054 00690061   ....s.z.T.r.a.i.
00152f40   0065006c 00000072 007a0073 00650048   l.e.r...s.z.H.e.
00152f50   00640061 00720065 90900000 00740053   a.d.e.r.....S.t.
00152f60   00740061 00730075 00610042 00000072   a.t.u.s.B.a.r...
00152f70   00570066 00610072 00000070 00640045   f.W.r.a.p...E.d.
0:004>
00152f80   00740069 90900000 8b909090 ec8b55ff   i.t..........U..
00152f90   8d026a51 6a50fc45 0400680d 05c70000   Qj..E.Pj.h......
00152fa0   0015c184 00154240 c18c05c7 000c0015   ....@B..........
00152fb0   15ff0000 00151114 fc7d8366 51850f31   ........f.}.1..Q
00152fc0   b800001a 000003e8 15c170a3 c178a300   ..........p....x.
00152fd0   05c70015 0015c150 0000a006 0002eeb8   ....P...........
00152fe0   c174a300 6ca30015 c90015c1 909090c3   ..t....l........
00152ff0   90909090 cd00a190 6c680015 ff0015c0   ..........hl....
```

何やら API 名の様なものも出てきました。

ところで、ここでいったんデタッチしてみます。qd ではなく、q で終了して下さい。そうすると、アタッチしていた notepad.exe も終了します。そして、再びメモ帳を起動して下さい。tlist で pID を確認すると、前回とは違う ID となっているはずです。pID は、OS によってランダムに決定されますので、都度、tlist で確認する必要があります。さて、デバッガでアタッチして !peb コマンドを入力してみましょう。

```
0:007> !peb
PEB at 7ffdf000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            Yes
    ImageBaseAddress:         009f0000
    Ldr                       77347880
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 00191ee8 . 001a9bc8
    Ldr.InLoadOrderModuleList:           00191e58 . 001a9bb8
    Ldr.InMemoryOrderModuleList:         00191e60 . 001a9bc0
            Base TimeStamp                     Module
        9f0000 4a5bc60f Jul 14 08:41:03 2009 C:¥Windows¥system32¥notepad.exe
       773c0000 521ea91c Aug 29 10:51:24 2013 C:¥Windows¥SYSTEM32¥ntdll.dll
```

75f50000 51fb10c5 Aug 02 10:52:05 2013 C:¥Windows¥system32¥kernel32.dll
756d0000 51fb10c6 Aug 02 10:52:06 2013 C:¥Windows¥system32¥KERNELBASE.dll
・・・

あれ？よく見ると、アドレスが違いますね。

（前回）

150000 4a5bc60f Jul 14 08:41:03 2009 C:¥Windows¥system32¥notepad.exe
773c0000 521ea91c Aug 29 10:51:24 2013 C:¥Windows¥SYSTEM32¥ntdll.dll
75f50000 51fb10c5 Aug 02 10:52:05 2013 C:¥Windows¥system32¥kernel32.dll
756d0000 51fb10c6 Aug 02 10:52:06 2013 C:¥Windows¥system32¥KERNELBASE.dll

（今回）

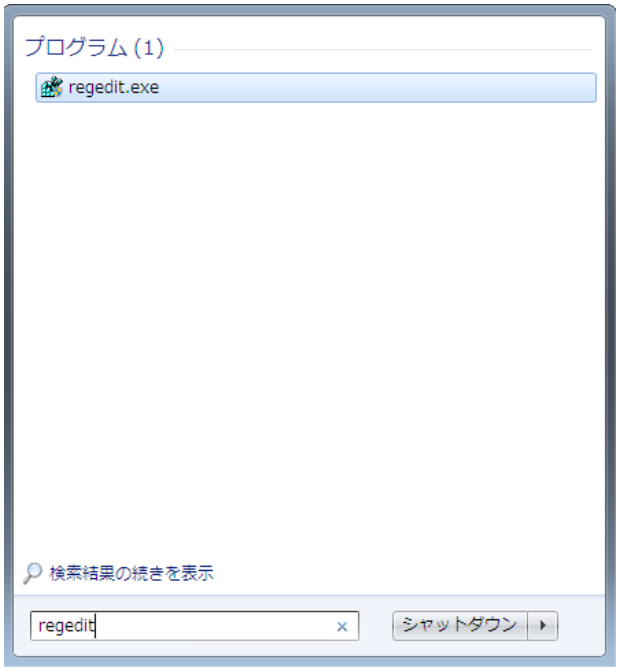
9f0000 4a5bc60f Jul 14 08:41:03 2009 C:¥Windows¥system32¥notepad.exe
773c0000 521ea91c Aug 29 10:51:24 2013 C:¥Windows¥SYSTEM32¥ntdll.dll
75f50000 51fb10c5 Aug 02 10:52:05 2013 C:¥Windows¥system32¥kernel32.dll
756d0000 51fb10c6 Aug 02 10:52:06 2013 C:¥Windows¥system32¥KERNELBASE.dll

notepad.exe の読み込まれたアドレスが異なっています。これは、Windows Vista 以降導入された ASLR (Address Space Layout Randomization) によるものです。この仕組みについては、

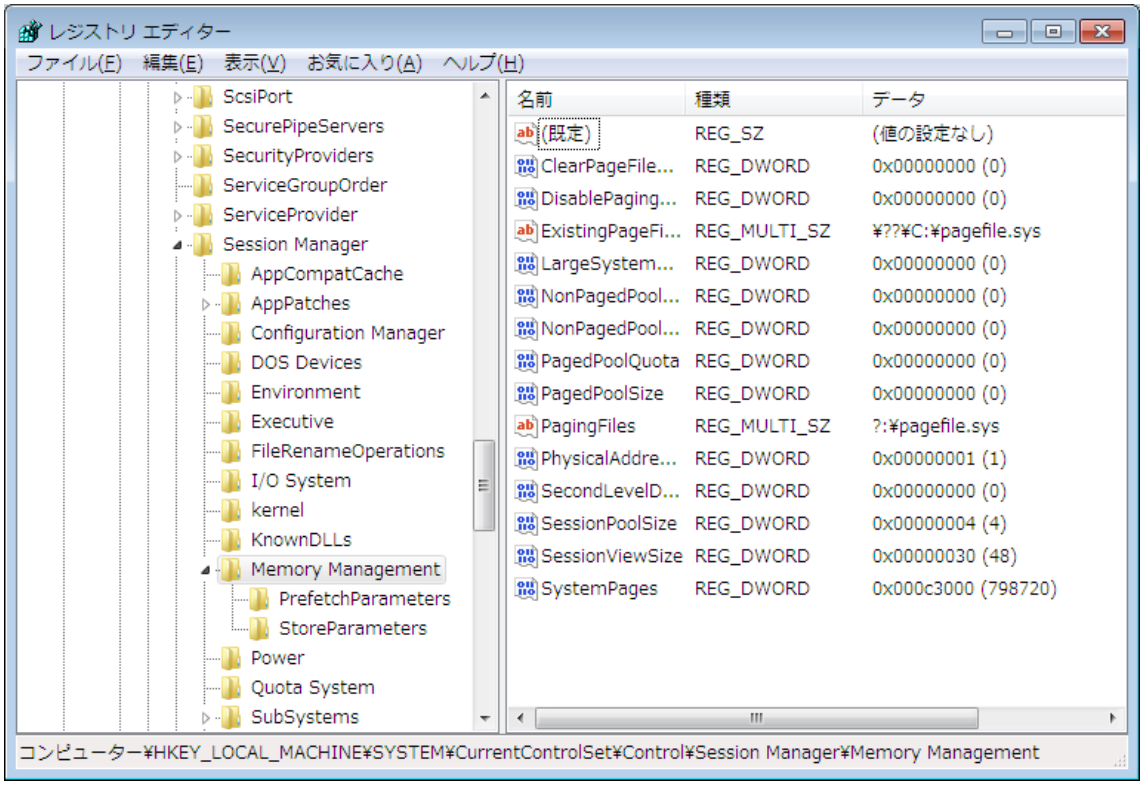
HKLM¥SYSTEM¥CurrentControlSet¥Control¥Session Manager¥Memory Management¥MoveImages

で設定できる様です。早速、レジストリエディタを使ってこの値を見てみましょう。

スタートメニューから [プログラムとファイルの検索] の箇所に “regedit” と入力します。



では、レジストリエディターで該当箇所を探してみましょう。



あれ？MoveImages という値が無い様です。

ちょっと調べてみると、ASLR について、Microsoft より次の技術情報が掲載されていまし

た。

An update is available for the ASLR feature in Windows 7 or in Windows Server 2008 R2
http://support.microsoft.com/kb/2639308/en-us

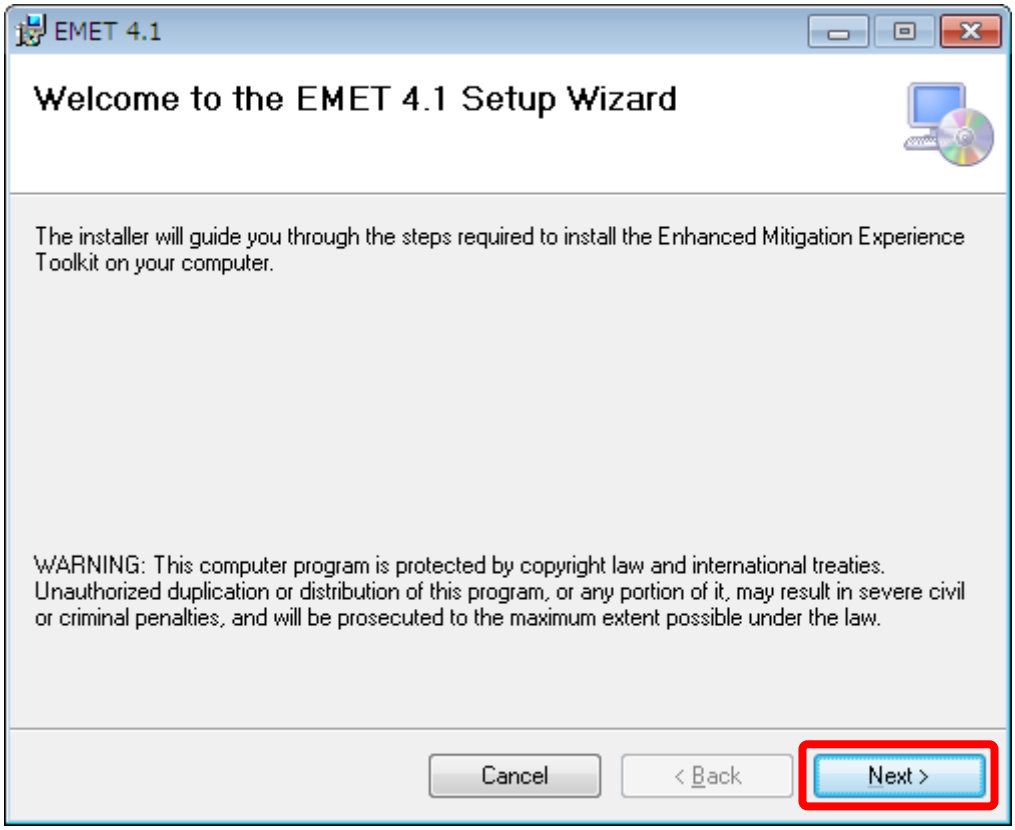
どうも、この更新プログラムを適用すると、ASLR を強制的に有効に出来る様です。このサイトからも更新プログラムはダウンロードできますが、Windows Update でも自動ダウンロードの対象となっているらしく、私のパソコンでは既に適用済みとなっていました。

そこで、もう少し調べてみると、どうも Microsoft が提供している EMET(Enhanced Mitigation Experience Toolkit) なるものがある様です。早速調べてみると、次のサイトに行きつきました。

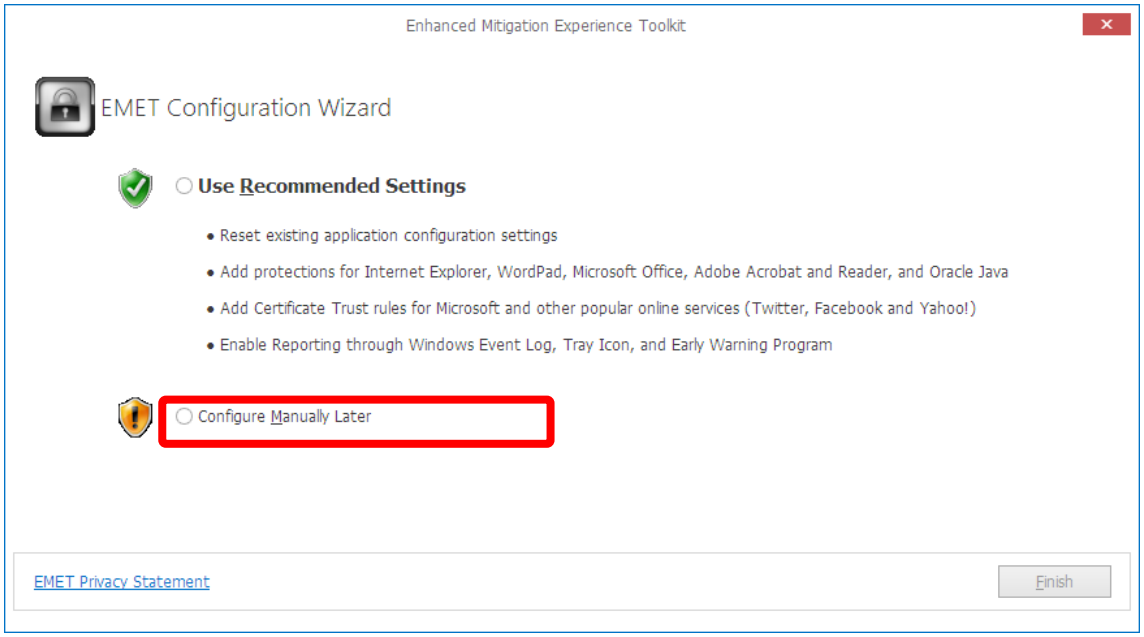
Enhanced Mitigation Experience Toolkit
http://technet.microsoft.com/ja-jp/security/jj653751

EMET 4.1 をダウンロードしてみます。

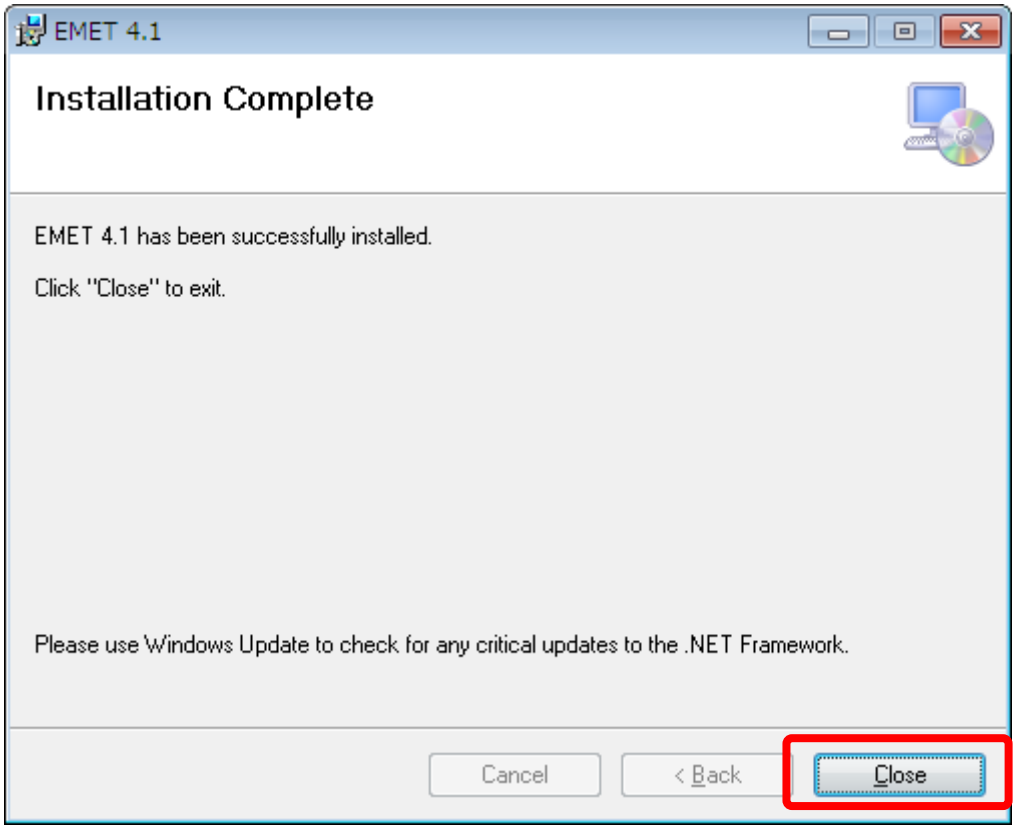
ダウンロードして実行してみると、セットアップが開始します。



セットアップ中、次の画面が出ますが、ここでは、”Configure Manually Later” を選択しておきます。

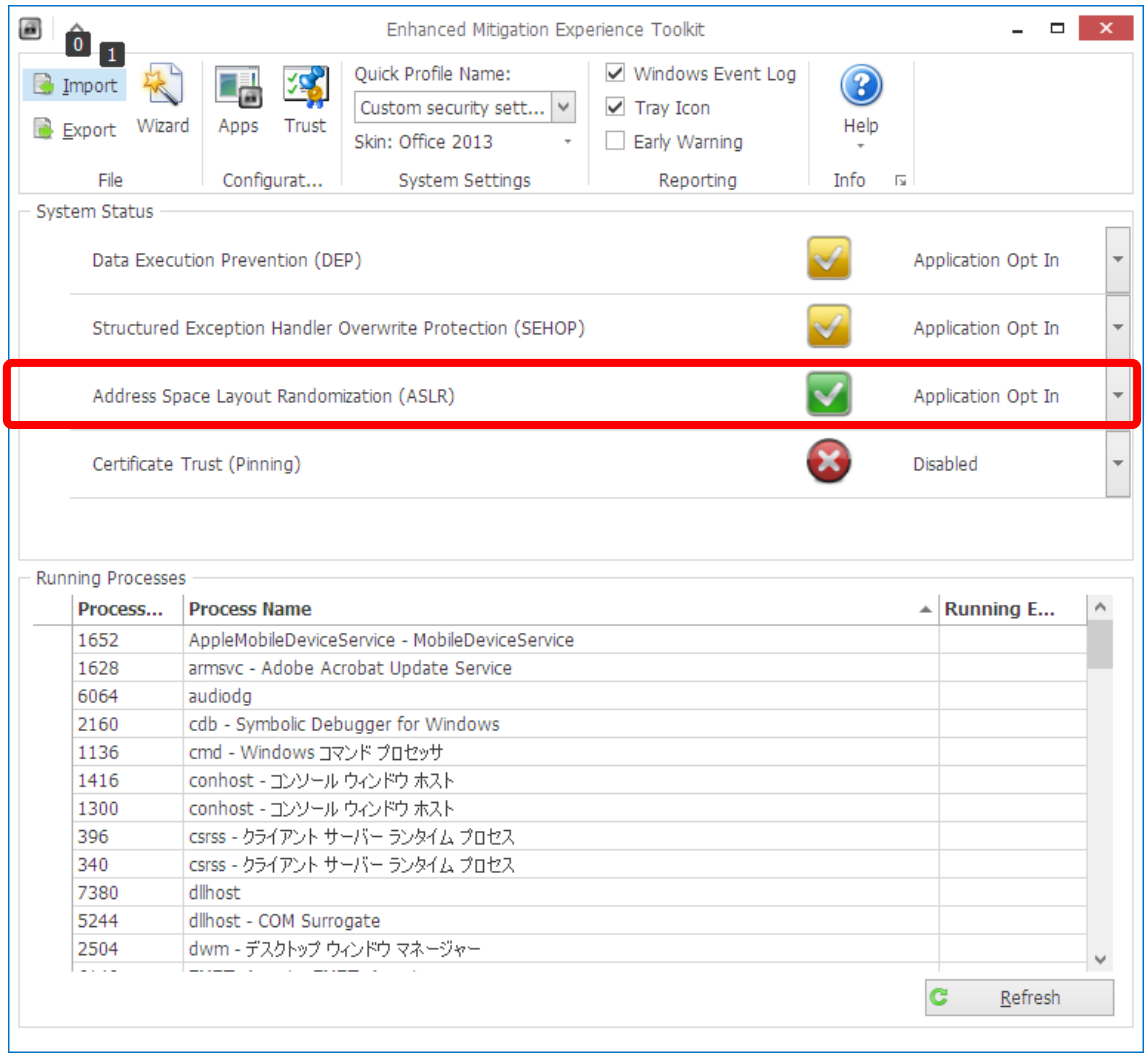


さらに進めていくと、次の様に、インストールが完了します。

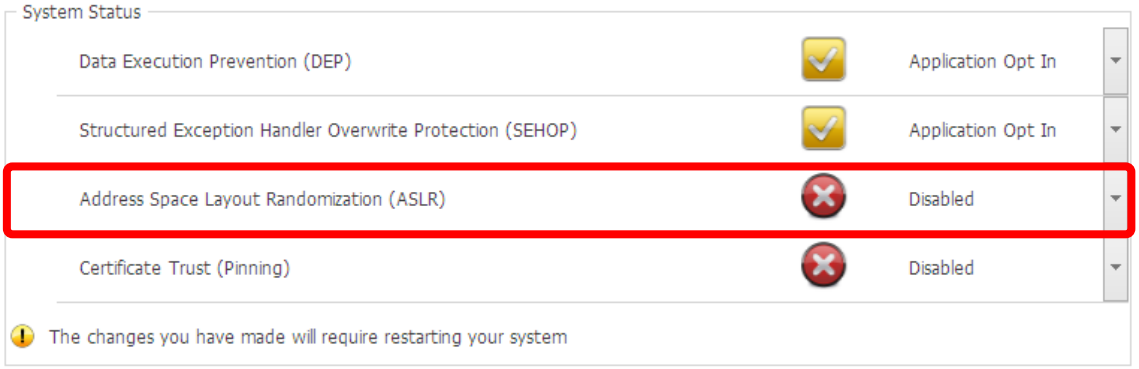


それでは、EMET を起動してみます。

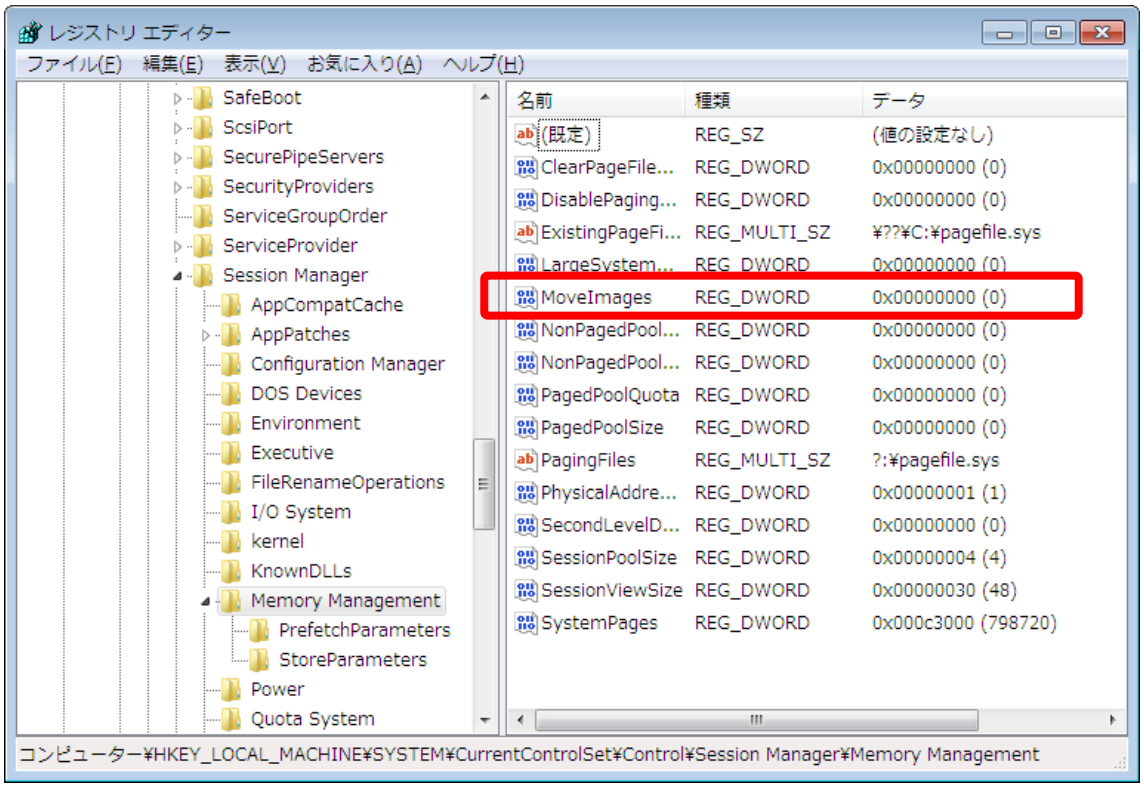
ASLR の有効・無効を設定出来る様です。今は有効になっていますね。



これを無効にします。右側の下矢印マークをクリックして Disabled に変更できます。



この時、レジストリエディタで [F5] キーを押してリフレッシュしてみると、MoveImages が作成されています。



ちなみに、EMET 上で、”Application Opt In” に変更すると、MoveImages は削除されます。

それでは、システムを再起動して、ASLR を無効にした状態で、2 回メモ帳を起動し、アタッチしてみます。

```
C:¥Program Files¥Debugging Tools for Windows (x86)>tlist
   0 System Process
   4 System
    ・・・
3832 notepad.exe          無題 - メモ帳
3752 tlist.exe

C:¥Program Files¥Debugging Tools for Windows (x86)>cdb -p 3832

Microsoft (R) Windows Debugger Version 6.12.0002.633 X86
Copyright (c) Microsoft Corporation. All rights reserved.
```

```
    ・・・
0:007> !peb
PEB at 7ffdf000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            Yes
    ImageBaseAddress:         01000000
    Ldr                       77f97880
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 00151ee8 . 00169648
    Ldr.InLoadOrderModuleList:           00151e58 . 00169638
    Ldr.InMemoryOrderModuleList:         00151e60 . 00169640
            Base TimeStamp                     Module
         1000000 4a5bc60f Jul 14 08:41:03 2009 C:¥Windows¥system32¥notepad.exe
        77ec0000 521ea91c Aug 29 10:51:24 2013 C:¥Windows¥SYSTEM32¥ntdll.dll
        77de0000 51fb10c5 Aug 02 10:52:05 2013 C:¥Windows¥system32¥kernel32.dll
         dce0000 51fb10c6 Aug 02 10:52:06 2013 C:¥Windows¥system32¥KERNELBASE.dll
    ・・・
0:007> q
quit:

C:¥Program Files¥Debugging Tools for Windows (x86)>tlist
   0 System Process
   4 System
    ・・・
3240 notepad.exe         無題 - メモ帳
3376 tlist.exe

C:¥Program Files¥Debugging Tools for Windows (x86)>cdb -p 3240

Microsoft (R) Windows Debugger Version 6.12.0002.633 X86
Copyright (c) Microsoft Corporation. All rights reserved.
    ・・・
0:007> !peb
PEB at 7ffdd000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            Yes
    ImageBaseAddress:         01000000
    Ldr                       77f97880
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 001a1ee8 . 001b9648
```

```
Ldr.InLoadOrderModuleList:              001a1e58 . 001b9638
Ldr.InMemoryOrderModuleList:            001a1e60 . 001b9640
            Base TimeStamp                     Module
        1000000 4a5bc60f Jul 14 08:41:03 2009 C:¥Windows¥system32¥notepad.exe
       77ec0000 521ea91c Aug 29 10:51:24 2013 C:¥Windows¥SYSTEM32¥ntdll.dll
       77de0000 51fb10c5 Aug 02 10:52:05 2013 C:¥Windows¥system32¥kernel32.dll
        dce0000 51fb10c6 Aug 02 10:52:06 2013 C:¥Windows¥system32¥KERNELBASE.dll
・・・
```

今度は、notepad.exe が同じアドレスにロードされている事がわかります。

（１回目：pID=3832）

```
    1000000 4a5bc60f Jul 14 08:41:03 2009 C:¥Windows¥system32¥notepad.exe
   77ec0000 521ea91c Aug 29 10:51:24 2013 C:¥Windows¥SYSTEM32¥ntdll.dll
   77de0000 51fb10c5 Aug 02 10:52:05 2013 C:¥Windows¥system32¥kernel32.dll
    dce0000 51fb10c6 Aug 02 10:52:06 2013 C:¥Windows¥system32¥KERNELBASE.dll
```

（２回目：pID=3240）

```
    1000000 4a5bc60f Jul 14 08:41:03 2009 C:¥Windows¥system32¥notepad.exe
   77ec0000 521ea91c Aug 29 10:51:24 2013 C:¥Windows¥SYSTEM32¥ntdll.dll
   77de0000 51fb10c5 Aug 02 10:52:05 2013 C:¥Windows¥system32¥kernel32.dll
    dce0000 51fb10c6 Aug 02 10:52:06 2013 C:¥Windows¥system32¥KERNELBASE.dll
```

つまり、ASLR が有効になっていると、メモリ空間のどのアドレスに実行ファイルが読み込まれるのかは固定でなくなるので、アドレスを指定してメモリ上にロードされているモジュールの一部を書き換えて不正を働かせる、といった攻撃が困難になります。なお、プロセスの再起動だけでは、その実行ファイルのアドレスだけが変更され、読み込まれるその他の DLL は同じアドレスとなりますが、システムごと再起動すると、再起動ごとに DLL も違う位置にロードされます。

なお、私の環境では、ASLR を無効にすると、Internet Explorer 11 が起動できなくなりました。有効にすると起動できますので、ASLR は無効にしない方が賢明という事ですね。

さて、長くなりましたので、いったんここで休憩します。次回第 3 回で続きを説明していきます。

では、今回のまとめです。

【まとめ】
・物理メモリと仮想メモリ
・ページファイル
・メモリ空間
・!peb コマンド
・dc コマンド
・ASLR
・EMET

今回は解説中心で少々わかりづらかったかもしれませんが、いかがでしたでしょうか？解説中心とはいえども、実際に手を動かす内容として、システム情報の確認方法、ページファイルの確認方法、モジュールがロードされるアドレスの確認方法、メモリ上に書かれている内容の確認方法、ASLR の ON/OFF などを行っていただきました。

次回は、今回のメモリ空間の続きで、複数のプロセスの話や、スタックやヒープといった、アプリケーションが利用するメモリ領域について、デバッガを使って確認していく予定です。