開発者必読！

“もしも”のときのデバッグ技法

.NETアプリケーション障害解析

株式会社NTTデータ
飯山 教史
IIYAMA, Takashi

第2回 WinDbgによる解析

Level

1 2 3 **4** 5

Technology Tools

- ❑ Visual Basic
- ❑ Visual C#
- ❑ Visual C++
- ❑ SQL Server
- ❑ Oracle
- ❑ Access
- ❑ ASP.NET
- ☑ Other:
  WinDbg（Debugging Tools for Windows）

Samples

user.dmpを解析するツール

先月のワトソンログの解析に引き続き、今回は同様の解析をuser.dmpで行なってみる。user.dmpもワトソン博士によって記録されるファイルで、その情報を元に故障原因を解析できる。しかしuser.dmpはワトソンログとは異なり、通常のテキストエディタでは開くことができないバイナリファイルとして保存されるため、その解析には特別なツール“WinDbg”が必要となる。

WinDbgはコマンドを利用して各種情報を表示するため、ワトソンログに比べると敷居が高いと感じる方もいるだろう。しかし、最低限必要な情報であれば機械的にコマンドを実行するだけで解析できる。

今回は「WinDbgを利用した解析手法」とともに、少々厄介な「WinDbgのデバッグ環境を構築する方法」を紹介する。

デバッグ環境の構築方法

WinDbgはMicrosoftが提供する、OS自体のデバッグを実行するためのツールである。Microsoftの以下のWebサイトで“Debugging Tools for Windows”として公開されている。

http://www.microsoft.com/japan/whdc/devtools/debugging/

ここからWinDbgをダウンロードしてインストールする。しかしインストールしただけではWinDbgは利用できない。WinDbgを利用するためには、

①シンボルサーバーの構築
②シンボルファイルパスの設定
③user.dmpの読み込み

といった作業が必要になる。

WinDbgでuser.dmpを解析するにはまず“シンボル情報”が必要になる。シンボル情報は、モジュール内に存在する関数や変数とそのオフセットとのペ

アリング情報である。Windowsでは、シンボル情報を、「.PDB」の拡張子を持つファイルに記録している。OS標準のモジュールとしてMicrosoftからPDBファイルが提供されているが、システム用に開発されたアプリケーションでは個別に設定しないとPDBファイルが作成されない。そのため、プログラム開発時にはリリース後のデバッグ作業のためにソースコードとともにそのアプリケーションのPDBファイルもあわせて管理しなければならない[注1]。

シンボル情報はOSのバージョンやSP（Service Pack）によっても異なるため、解析には毎回user.dmpと同じバージョンのシンボルファイルが必要となる。このペアリングを自動的に行なうのが、Microsoftが提供する「シンボルサーバー」である。WinDbgはシンボルサーバーを利用して、解析するダンプファイルと、利用するシンボル情報のマッピングを行なう。シンボルサーバーには、以下の2種類の利用方法がある[注2]。

①シンボルファイルをローカルの環境にすべてダウンロードし、マッピング用のファイルサーバーを構築する
②WinDbgから直接MicrosoftのWebサイト（シンボルサーバー）へアクセスして必要なシンボルをダウンロードして利用する

今回は②の直接アクセスする方法を利用して解析を進める。

WinDbgをシンボルサーバーに接続するためには、WinDbgのメニューから［File］-［Symbol File Path］と選択し、表示される「Symbol path」に、

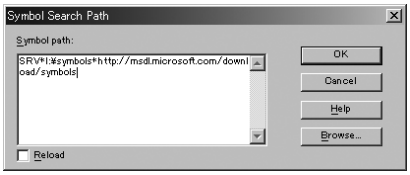
```
SRV*<シンボルファイルを格納するフォルダへのパス>*http://msdl.microsoft.com/download/symbols
```

と記述する。たとえば、「I:¥symbols」へ格納するなら、図1のように、

```
SRV*I:¥symbols*http://msdl.microsoft.com/download/symbols
```

となる[注3]。これでuser.dmpを解析する下準備が完了した。

図1：シンボルファイルパスの設定



WinDbgでuser.dmpを解析する

では実際にuser.dmpを解析してみる。解析対象は先月のプログラムと同じ障害時に作成されたuser.dmpである。まず、メニューから［File］-［Open Crash Dump］と選択し、解析するuser.dmpを指定する。ファイルを開くとWinDbgの画面にさまざまな内容の情報が表示される（図2）。

はじめに注目してもらいたいのは下から2番目にある情報である。今回のuser.dmpでは「StackInfo+0x101d:」と表示されている。これは、今回の障害がstackInfo.exeの中で発生したことを示している。「+0x101d」は障害が発生したコードのアプリケーション内（Stack

図2：WinDbgでuser.dmpの内容を表示

```
Command
Symbol search path is: SRV*I:\symbols*http://msdl.microsoft.com/download/symbols

Microsoft (R) Windows Debugger  Version 6.3.0017.0
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [                \debug\user.dmp]
User Dump File: Only application data is available

Windows 2000 Version 2195 UP Free x86 compatible
Product: WinNt
Debug session time: Sun Mar 20 14:20:02 2005
System Uptime: 2 days 22:55:47.835
Process Uptime: not available
WARNING: \\bofsrv\symbols\nt50.sp4.jp is not accessible, ignoring
Symbol search path is: SRV*          *http://msdl.microsoft.com/download/symbols
Executable search path is:
...
(650.6bc): Access violation - code c0000005 (!!! second chance !!!)
eax=00000007 ebx=7ffdf000 ecx=00000001 edx=00340000 esi=00407030 edi=00000000
eip=0040101d esp=0012ff74 ebp=0012ffc0 iopl=0         nv up ei pl nz na pe cy
cs=001b  ss=0023  ds=0023  es=0023  fs=0038  gs=0000             efl=00000203
*** WARNING: Unable to verify checksum for stackInfo.exe
*** ERROR: Module load completed but symbols could not be loaded for stackInfo.exe
stackInfo+0x101d:
0040101d f3a5            rep  movsd ds:00407030=65686568 es:00000000=????????
```

注1）デバッグバージョンではPDBはデフォルトで作成されるが、リリースバージョンでは作成されない。リリース後のサポートを考慮し、必ずPDBファイルを作成しておくことをお勧めする。作成方法は以下のURLで紹介されている。
・MSDNライブラリ「デバッグ構成とリリース構成」
http://www.microsoft.com/japan/msdn/library/default.asp?url=/japan/msdn/library/ja/vsdebug/html/_core_Debug_Build_Versus_Release_Build.asp
注2）シンボルサーバーの使用方法については、以下のURLを参照。
http://www.microsoft.com/japan/whdc/devtools/debugging/symbols.mspx
注3）http://msdl.microsoft.com/download/symbolsは直接参照できない。WinDbgからのアクセスのみを対象としている。