

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, Takeshi

新連載

ワトソンログの解析

Level

1 2 3 **4** 5

Technology Tools

- ☐ Visual Basic
- ☐ Visual C#
- ☐ Visual C++
- ☐ SQL Server
- ☐ Oracle
- ☐ Access 2002
- ☐ ASP.NET
- ☒ Other:

Samples



開発者を悩ませるモノ

Windowsプラットフォームで実装されたシステムでのトラブルはとどまるところを知らない。特にシステムでアプリケーションを実装しているプログラマを悩ませているのは“ワトソン博士”ではないだろうか? Windows上のアプリケーションで例外が発生して異常終了すると(アプリケーション例外の場合。システムトラップなど、OSごと強制終了する場合もある)、ワトソン博士は画面上にポップアップを表示し、アプリケーションを強制終了する。開発者にとってこのポップアップは辛いデバッグの開始を告げる前兆として忌み嫌われているようだ。

しかし、この表示と同時にワトソン博士はエラー状態をログやクラッシュダンプファイルに記録している。本来ならこれらの情報を基にアプリケーションが障害を起こした原因を解析しなければならないが、解析するにはいくつかのハードルがあるため、無視され

ることが多い。

本連載が読者の皆様にとってこれらのハードルを効率的にクリアするためのきっかけとなり、ひとりでも多くの開発者がワトソン博士を有効に利用できるようになれば、と考えている。



ワトソン博士とは?

そもそもワトソン博士とは何だろうか? ワトソン博士はWindows OSに組み込まれているデバッガである。デバッガとは、バグを検出/修正するために利用されるツールであり、Windowsはさまざまなツール/ソフトウェアをデバッガとして利用できる^[注1]。新規にソフトウェアがデバッガとして登録されていなければ、NT系OS(2000/XP/Server 2003)ではデフォルトでワトソン博士が起動するようになっている^[注2]。これは、アプリケーション実行時に「例外」と呼ばれる事象が発生す

注1) たとえばVisual Studio .NET 2003をインストールすれば、それがデバッガとなる。

ると、プログラム側で対応コードが書かれていない場合にはワトソン博士がその後処理を行なうことを意味している。

.NETへ移行したばかりの開発者にはなじみは薄いと思われるが、アプリケーションでは例外が発生する。例外の定義は難しいが、強いて言うなら「それ以上の実行が困難であると判断されたプログラムの後処理用のシグナル」と定義できる。例外は0除算などの「ハードウェア例外」とOSやアプリケーションがコードの中でエラー処理のために自分で発生させる「ソフトウェア例外」の2種類が存在する。アプリケーションを開発するプログラマは、アプリケーションが実行中に発生する可能性がある例外を処理する「ハンドラ」と呼ばれるコードを記述し、例外に対処しなければならない。

しかしシステム開発の現場ではこの例外処理用のハンドラを記述していないコードが多く存在する。そのため、発生した例外は開発者の書いたプログラム内では処理されず、例外はハンドラを求めて呼び出し元の関数を順番にたどるが、最後までハンドラが見つからない場合にはシステムが提供する「UnhandledExceptionFilter」関数に受け渡される。この関数内で皆さんがよく見る例外発生を告げるアプリケーションエラーのダイアログが表示され、デフォルトでデバグとして登録されているワトソン博士を起動するのである。

ワトソン博士は設定により「ワトソンログファイル (drwtsn32.log)」と「クラッシュダンプファイル (user.dmp)」を記録する^[注3]。コマンドプロンプトから「drwtsn32」と入力すると、ワトソ

ン博士の設定画面が表示される (図1)^[注4]。クラッシュダンプファイルの解析方法の説明は次回から始める。今回はワトソンログを利用した最低限の解析に必要なOS/CPUの知識について簡単にまとめた。

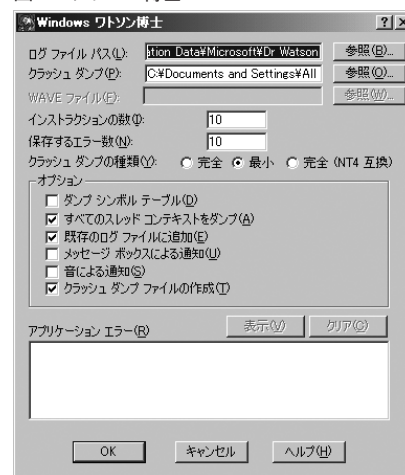
OSのアプリケーション管理

Windowsでは、アプリケーションを「プロセス」と呼ばれる単位で管理している。たとえば、Windows上でExcelやWordといったアプリケーションを起動すると、OSはそれぞれのアプリケーションにプロセスIDを付与して個別なプロセスとして管理／実行している。

プロセスは独立して存在し^[注5]、他のプロセスの情報を利用／編集することは原則としていない。そのためユーザーは複数のアプリケーションを同時に利用しても安定性を失うことなく操作できるのである。

「プロセス」は非常に抽象的で「何？」と問われて正確に回答できる開発者は少ない。ワトソン博士を利用して故障部位^[注6]を調査するために利用される情報は例外が発生したプロセス内の情報に限られている。それは、障害

図1：ワトソン博士



時に複数のプロセスが実行されていても、ワトソン博士の仕様で例外を引き起こしたコードを持つプロセスの情報だけを記録するからだ。

故障部位とは「例外を引き起こしたコードの命令アドレス^[注7]」である。システム状態が異常となりそのアドレスの命令を実行したときに例外が発生するので、ワトソン博士を利用して行なう「故障部位の特定」とは、実は「どのプロセスのどの命令アドレスで例外が発生したのか」を調査することと同じである。

プロセスとは別に、プログラムの実行単位として忘れてはならないのが「スレッド」である。スレッドはプログ

注2) どのプログラムがデバグとして登録されているかは、以下のレジストリで確認可能。

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AeDebug

注3) 設定については以下のURLを参照。

http://www.microsoft.com/windows2000/ja/professional/help/default.asp?url=/windows2000/ja/professional/help/app_dr_watson.htm

注4) デフォルトではオフになっている「ダンプシンボルテーブル」は必ずオンにしてシンボル情報を記録したほうが良い。

注5) ここで言う独立とは、他のアプリケーションにメモリ上のデータが改ざんされないなどのように、お互いに干渉しないようOSがプロセスを管理していることを示している。

注6) 故障原因とは異なることに注意。

注7) アドレスとは、操作されるデータがメモリ内のどこに格納されているかを特定する場合に利用されるメモリ上の番地である。命令アドレスとは、アドレスの内「実行されるコードが格納されているメモリ上の場所」のことである。