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5回

実行中の.NETアプリケーションの解析

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）
  - SOS.DLL

Samples

.NETアプリケーションの中身

今まではクラッシュが発生した後にワトソン博士が作成したログやuser.dmpを解析する方法を説明してきた。クラッシュの原因を調査するには、クラッシュが発生した瞬間のメモリ状態を解析する必要があるからである。

しかし、メモリリークが発生しているときには、どのクラスがリークしているのかなど、状況によってはアプリケーションの動作中にメモリの状態を調査したい場合がある。そこで今回は実行中の.NETアプリケーションにWinDbgをアタッチし、SOS.DLLのコマンドを利用してメモリの内容を確認する方法を紹介する。

“.NETアプリケーションの状態”とは？

実際の調査手順を説明する前に、.NETアプリケーションの構成を簡単に説明する。ご存知のように.NETではメモリ管理機能“ガベージコレクション”が提供されている。プログラムでインスタンス化（newされたクラス）されたすべてのリソースが「マネージヒープ」と呼ばれるヒープ領域に確保され、アプリケーションから利用されなくなると自動的にそのリソースを解放してくれるのである[注1]。

プログラムの状態を調べるには主にこのリソースの状態を調べる必要がある。それはマネージヒープ内のオブジェクトの状態によって、そのときのプログラムの振る舞いが決められるからだ。もちろん、メソッドに渡されたパラメータはスタックに詰まれるので、スタックの状態もチェックする必要がある。しかし、渡される値のほとんどがクラスの参照であり、その参照先はマネージヒープ上のオブジェクトであることを考えれば、やはり最終的には

注1）ガベージコレクションの詳細は以下のURLを参照。
http://www.microsoft.com/japan/msdn/net/mag00/GCI.asp
http://www.microsoft.com/japan/msdn/net/mag00/GCI2.asp

リスト1：SimpleClassをインスタンス化するコード

```
private void button1_Click(object sender, System.EventArgs e)
{
   int i = 0;
   string str = "Hello";
   SimpleClass simple = new SimpleClass();
   simple.method_1(ref i, ref str);
   MessageBox.Show("Simple Class in GC.¥n");
}
```

リスト2：SimpleClass

```
public class SimpleClass
{
   // フィールド
   public int k = 200;
   public string str = "This is simple class.";
   // メソッド
   public void method_1(ref int α, ref string strHello)
   {
      α=1;
      strHello = "Bye";
   }
   public void method_2(ref long l)
   {
      l = 200;
   }
}
```

マネージヒープからその内部にあるリソースの状態をチェックする必要がある。

それでは、具体的にどのようなアプリケーションの状態を確認すればよいのだろうか？　いろいろと確認しなければならない情報はあるが、これまでどおり故障原因ではなく“その部位を特定する”といったレベルで留めるのであれば、以下の2点を確認できれば十分である。

①インスタンス化されているクラスは何か？
②クラスの状態は？

①と②がわかれば、任意のタイミングでクラスのフィールドの値が確認できるようになり、それらの値によるアプリケーションの挙動の違いを確認できるようになるからだ。また、①がわかれば、メモリリークを解析するときには現在のマネージヒープに存在するクラスのメモリ内のサイズと数がわかるので、メモリリークの解析もできるようになる[注2]。

本稿では、アプリケーション実行中の状態を確認するツールとして、これまでどおりWinDbgを利用する。WinDbgにはプロセスにアタッチしてプロセスの実行を停止し、コマンドを実行してアプリケーションの状態を確認する機能が提供されている。今回はそれらのコマンドの中から上記の①と②を確認する方法を紹介する。

注2）メモリリークの解析については次号で説明する。

プロセスにアタッチする

はじめにWinDbgでプロセスにアタッチする方法を説明する。いつものようにSOS.dllを利用できる状態でWinDbgを起動する[注3]。

続いて、デバッグ対象のアプリケーションも起動し、アプリケーションの状況を確認したいところまで処理を進め、その状態でWinDbgの［File］-［Attach to a Process］を選択する。すると、現在マシン上で実行しているプログラムの一覧が表示されるので、デバッグ対象のプロセスを選択してアプリケーションの進行をストップする。その後、Commandウィンドウから、

```
.load sos
```

を実行してSOS.dllのコマンドを利用できるようにして、デバッグを開始する。

▶何がメモリにあるのか？

今回は実行中のアプリケーションで利用されているクラスの状態を見るため、サンプルを用意した。

リスト1はSimpleClassをインスタンス化するコード、リスト2はSimpleClassの実装である。

実行中のマネージヒープの状態を見るため、SimpleClassがインスタンス化

注3）SOS.dllを利用するための設定については、本誌7月号の本連載を参照してもらいたい。