「ハング時PC位置検索ツール」説明書

2013/05/19

森昭人

**●概要**

これは3DS用のプログラムが止まってしまった時に表示されるアドレス情報から止まった関数の位置を検索するツールです。主にDLL内で止まった時のオフセットを自動で計算できるようにするために作成しました。

「xy\_tool\PC\_DLL\_Search」フォルダにコミットしています。

**●説明**

3DSプログラムには開発実機上でハングアップした際には「プログラム例外ハンドラ」という機能で情報を取得する事ができます。

そこでKujiraでは止まった瞬間のPCカウンタやレジスタ・スタック・DLL割り付け情報がエラーウインドウで表示されるようにしました。

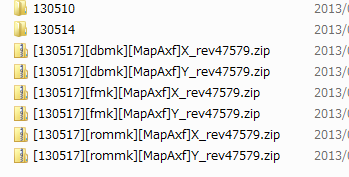
この「PCカウンタ」と「DLL割り付け情報」を入力して、実際にどの関数の中で止まったのかmapファイルを検索できるようにしたのがこのツールです。

**●使い方**

**１．mapファイルをツールのフォルダに展開する。**

ROM焼き日に作成したROMがハングアップしてしまった場合、mapファイルをサーバーからコピーしてくる必要があります。

X:\rom\mapaxfフォルダにROMを焼いた日ごとにmapファイルがバックアップしてありますので、コピーしてきてください。



『「作成年月日」「作成ビルド」「MapAxf」バージョン\_SVNリビジョン番号.zip』というファイル名で保存されています。

解凍してこのツールのフォルダにコピーしてください。

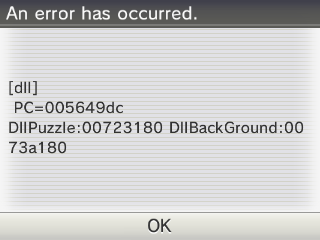
**２．ツールの起動**

ツールを起動した時に現れるのがこの画面です。



このツールを起動しているという事は、プログラムがROMでハングアップしてしまったという事だと思います。

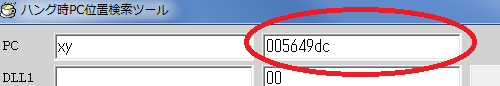
エラーウインドウが表示された時の画像も存在しているはずです。



この画像をもとにツールに情報を入力していきます。

**３．PC位置の入力**

「PC=005649dc」という部分を入力します。

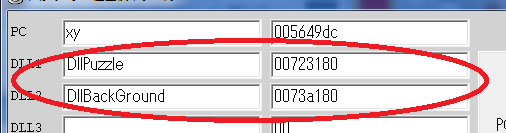


**４．DLL情報の入力**

「DllPuzzle:00723180」

「DllBackGround:0073a180」

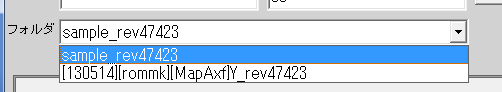
という情報を下の欄に入力します。



**５．フォルダの指定**

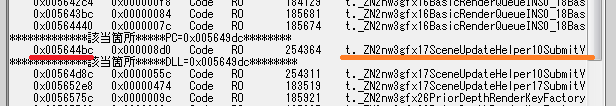
１．でツールのフォルダにコピーしてきた参照するmapファイルのフォルダを選択します。

作成したバージョン・SVNのリビジョンが指定されないと関数のアドレスは正しく取得できません。



**６．検索ボタンの実行**

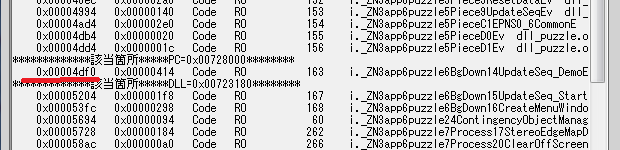
検索に成功すると実行結果が下のテキストボックスに表示されます。



赤字の部分がPC位置に一番近い関数の先頭アドレスを示しています。

オレンジ色の部分が実際のC++のクラス名をマングリングした文字列になります。（ツールでは右にスクロールできます）

上の結果は静的領域のプログラムアドレスを直接指しています。



この結果の場合はDLLの中の関数が見つかった時の表示です。

ＤＬＬの場合は「動的割り付け」なわけですからmapファイルには本当のアドレスではなく、スタート位置からのオフセット情報しかありません。

４．で入力した動的割り付けアドレスを足した上で算出される場所が表示されます。

例えば「00728000」という箇所を検索した結果が上の画像ですが、アドレス情報には「0x00004df0」というアドレスが表示されています。

このアドレスはDllBackGroundが割り付けられた「0x00723180」という数値と足すと「0x00727f70」というアドレスになり、検索該当箇所に近い場所を指していることが分かります。

**※使い方の補足【実践編】**

PC位置を入力して検索しても見つからない事があります。

その場合もエラー画面の他の数値を入力してみましょう。

レジスタやスタックにあるコードアドレスは止まった箇所のプログラムに近い可能性が高いためです。

特に「00123456」という数値のように上位が「０」２つになっている数値を探してみましょう。

例）

|  |  |
| --- | --- |
| 3番目に表示される画像 | この画像のPCアドレス（0x000001b4）を探しても関数は見つかりません。アドレスの数値が小さすぎると検索に失敗します。  XY.mapを見るとコード領域は0x00100000からスタートしているのが分かります。 |

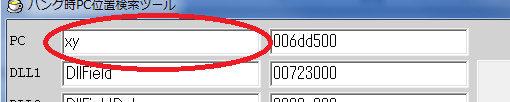
|  |  |
| --- | --- |
| 最初に表示される画像 | レジスタ情報のプリントの中の数値に「08d0cfe0」や「8d040e0」というのがありますが、この数値は大きすぎるため関数の検索には失敗します。  （ちなみにこの数値はコードアドレスではなくヒープ等のワークアドレスを示しています。） |

|  |  |
| --- | --- |
| ２番目に表示される画像 | スタックを表示した内容の中に一つだけコード領域を指している数値が存在しています。  ８番目の「00819d28」です。  この数値は検索するとＭＡＰファイルの中の関数にヒットしました。 |

Kujiraプロジェクトの静的領域プログラムサイズはだいたい0x0072000で終了しているため、

DllFieldが0x00721600あたりから割り付けられている事が多いようです。

**●追記１**



「xy」と書いている部分は上の説明では何も使用していませんが、これは静的領域のmapファイル名を示しています。別プロジェクトでも使用できるようにするために書き込み可能な場所に配置しています。次のプロジェクトの静的領域名が「xy2」であればこの欄を書き換えるだけで対応できます。

**●追記２**

　・検索した条件はログファイルに保存されるため、次回立ち上げた時に前回の検索条件として　内容が復帰します。

　・「develop.zip」は開発環境一式です。ツールを使用するだけなら解凍する必要はありません。

●プログラム的な工夫

・このツールはmapファイルのアドレスと関数定義部分を読み取って探索するツールですが、そのアドレス定義部分は

　最大３万行程存在するため、正直に検索すると5分ぐらい返ってこないという事が発生していました。

　そこれで探索スピードを上げるためにいくつか工夫しました。

　・ファイルを開いた後、実際の関数定義の部分までを破棄する。

　→ＨＳＰのテキストデータ処理はファイルが大きくなるほど、後ろの行になるほど重くなる傾向があったため、読み込まない場所を

　　破棄する処理はスピードアップにつながりました。

　・探索するアドレスの近くに来るまでは１０００行飛ばしで読み込んでいく。

　　→１万行目にあるかもしれないアドレス値を１行目から読み込んでいくのはとても時間がかかってしまうため、

　　　１０００行ずつ飛ばしながら読み込んで指定のアドレウを超えた時点で、１回分読み込む場所を戻して探索を始めるようにしました。

　　　この処理はそのまま探索が１０００倍速くなりました。