

// 【HSP3】 libxml2を使ってHTMLタグを列挙してみるサンプル & hscallbk.dllをcdecl
規約に対応してみる

// libxml2.dll は <http://xmlsoft.org/sources/win32/> この辺から探してくる。

// iconv.dll と zlib1.dll も内部的に使っているようなので、どこからか拾ってくる。

// 呼び出し規約については <http://prasinoblog2.fc2.com/blog-entry-627.html> を参
考にしました。

// 逆アセンブルにxdawinを使用してhscallbkが書き出したコードを調査しました。

```
#uselib "kernel32.dll"
```

```
#func _VirtualProtect "VirtualProtect" var,int,int,var
```

```
#define xdim(%1,%2) dim %1,%2: _VirtualProtect %1,%2*4,$40,AZSD
```

// xdimで動かないときはxdim2で。ただし開放処理がないよ。

```
#define xdim2(%1,%2) VirtualAlloc 0, %2*4, 0x2000, 0x40: \
```

```
VirtualAlloc stat, %2*4, 0x1000, 0x40: \
```

```
dupptr %1, stat, %2*4, 4
```

```
#include "hscallbk.as"
```

```
#uselib ""
```

```
#func startElementDummyProc "" int, int, int
```

```
#func endElementDummyProc "" int, int
```

```
#uselib "libxml2.dll"
```

```
#func htmlCreatePushParserCtxt "htmlCreatePushParserCtxt" int, int, int, int,  
int, int
```

```
#func htmlDefaultSAXHandlerInit "htmlDefaultSAXHandlerInit"
```

```
#func htmlCtxtUseOptions "htmlCtxtUseOptions" int, int
```

```
#func htmlParseChunk "htmlParseChunk" int, int, int, int
```

```
#func htmlFreeParserCtxt "htmlFreeParserCtxt" int
```

```
#define XML_CHAR_ENCODING_UTF8 1
```

```
#define HTML_PARSE_RECOVER 1
```

```
#define HTML_PARSE_NOERROR 32
```

```
#define HTML_PARSE_NOWARNING 64
```

```
// hscallbk
```

```
setcallbk startElementHandler, startElementDummyProc,
```

```
*_startElementHandler
```

```
setcallbk endElementHandler, endElementDummyProc,
```

```
*_endElementHandler
```

```
// cdecl規約の呼び出しに無理やり対応させる（動かないケースもあるか
```

も・・・?)

```
dupptr startElementHandlerPtr, varptr(startElementHandler), 40, 2  
// RET xxxx -> RET 0 に直す(cdeclの場合、呼び出し元が ADD esp, xxxx するため)
```

```
wpoke startElementHandlerPtr, 18, 0
```

```
// 同じく
```

```
dupptr endElementHandlerPtr, varptr(endElementHandler), 40, 2
```

```
wpoke endElementHandlerPtr, 18, 0
```

```
// libxml2
```

```
data = "<html><head><title>aaa</title></head><body onLoad=\"test()  
\"><p>test</p></body></html>"
```

```
dim saxHandlerStruct, 128/4
```

```
htmlDefaultSAXHandlerInit
```

```
saxHandlerStruct.14 = varptr(startElementHandler)
```

```
saxHandlerStruct.15 = varptr(endElementHandler)
```

```
htmlCreatePushParserCtxt varptr(saxHandlerStruct), 0, 0, 0, 0,  
XML_CHAR_ENCODING_UTF8
```

```
parserContext = stat
```

```
htmlCtxtUseOptions parserContext, HTML_PARSE_RECOVER |  
HTML_PARSE_NOERROR | HTML_PARSE_NOWARNING  
;mes stat
```

```
htmlParseChunk parserContext, varptr(data), strlen(data), 1  
;mes stat
```

```
htmlFreeParserCtxt parserContext
```

```
stop
```

```
*_startElementHandler
```

```
dupptr name, callbkarg(1), 1, 2
```

```
mes "<"+name+">"
```

```
return
```

```
*_endElementHandler
```

```
dupptr name, callbkarg(1), 1, 2
```

```
mes "</"+name+">"
```

```
return
```

