

Detecting_memory_leaks_with_HookLogger

It is one of the most time consuming task for developers to find memory leaks in their code. If your application happens to panic on exit with some memory leak, then sometimes it is a hard work to find the wrong piece of code with code analysis. HookLogger is a great emulator tool by Symbian that helps finding the leak fast.

Contents

- [1 How does HookLogger work?](#)
- [2 Installation](#)
- [3 Installation - tamhan's way](#)
- [4 Usage](#)
- [5 making sense of el dump](#)
- [6 Internal Links](#)
- [7 External Links](#)

How does HookLogger work?

This tool replaces EUSER.DLL with its own to enable logging of memory allocations, process and thread creation, leaves, etc. HookLogger application collects these logs and presents them to the user in an easily readable form.

Installation

HookLogger can be [downloaded here](#). It has a convenient installer.

Notes:

- HookLogger's installation path must not contain any spaces. If it does (e.g. installing to the default location under Program Files\Common Files\Symbian\tools), you will have to edit two lines at the end of HookEUSER.pl. Replace

```
Line Number: 74 my $cmd = "copy $hooks_src";
```

with

```
my $cmd = "copy \"$hooks_src\"";
```

and

```
Line Number : 77 $cmd = "$Bin/AttachDll $euser $hooks $hooked_euser";
```

with

```
$cmd = "\"$Bin/AttachDll\" $euser $hooks $hooked_euser";
```

- S60 platform's installation path must not contain any spaces. This is not a severe issue as almost all tools in the Symbian toolchain have this limitation in any case.

Detecting_memory_leaks_with_HookLogger

Of course, you can also replace EUSER.DLL manually.

You might need to add following line to project .mmp files if HookLogger call stack display is not correct:

```
OPTION CW -frame on
```

.

Installation - tamhan's way

I have stumbled across a later version of HookLogger which no longer supports the command above. My command sequence for activating the critter is as follows:

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Dokumente und Einstellungen\TAMHAN)cd ..

C:\Dokumente und Einstellungen)cd ..

C:\)cd Programme

C:\Programme)cd HookLogger

C:\Programme\HookLogger)HookEUser winscw
The "HookUser" utility has been replaced by "SetupHooks",
which supports "devices" directly. Here's the syntax
:
** usage:
SetupHooks ((DEVICE)) [--remove|r] [--platform=((PLATFORM))] [--build=((BUILD))]
** where:
((DEVICE)) specifies the device name to attach hooks to (as listed by 'devices'),
or DEFAULT for the default device, or EPOCROOT to attach to %EPOCROOT%\epoc32
(as used by legacy Symbian tools)
((PLATFORM)) defaults to WINSCW
((BUILD)) defaults to UDEB
--remove detaches the hooks, restoring the original EUSER.DLL

C:\Programme\HookLogger)SetupHooks
** usage:
SetupHooks ((DEVICE)) [--remove|r] [--platform=((PLATFORM))] [--build=((BUILD))]
** where:
((DEVICE)) specifies the device name to attach hooks to (as listed by 'devices'),
or DEFAULT for the default device, or EPOCROOT to attach to %EPOCROOT%\epoc32
(as used by legacy Symbian tools)
((PLATFORM)) defaults to WINSCW
((BUILD)) defaults to UDEB
--remove detaches the hooks, restoring the original EUSER.DLL

C:\Programme\HookLogger)devices
S60_5th_Edition_SDK_v0.9:com.nokia.s60 - default

C:\Programme\HookLogger)SetupHooks S60_5th_Edition_SDK_v0.9:com.nokia.s60
Setting up hooks in "C:\S60\devices\S60_5th_Edition_SDK_v0.9\epoc32\release\WINS
CW\UDEB"
1 Datei(en) kopiert.
Setup euser.dll to load EUserParasite_EKA2.dll, original preserved as euser.orig
.dll.
Run SetupHooks with --r to restore it
```

Detecting_memory_leaks_with_HookLogger

```
C:\Programme\HookLogger)
```

I had to use ((and)) to replace the ampersands. Cutting a long story short, the process involves first running devices to obtain a list of targets, then assign it to a specific target.

Usage

1. There is a script provided with HookLogger to do that. Open a command line in HookLoggers's installation folder and type:

```
HookEUSER winscw
```

After this one might get this kind of error

```
C:\Program Files\Common Files\Symbian\tools>HookEUSER winscw Target path is
C:\Symbian\9.1\S60_3rd_MR\epoc32/release/winscw/UDEB
```

```
1 file(s) copied.
```

'C:/Program' is not recognized as an internal or external command, operable program or batch file. Failed running "C:/Program Files/Common Files/Symbian/tools/AttachDll euser.dll EUserParasite_eka2.dll euser.dll.HOOKED" at HookeUSER.pl line 78. at line 78 one can set the path of the sdk that he might be working on

```
my $path = "C:/Symbian/9.1/S60_3rd_MR/Epoc32/release/$platform/$release";
```

or you can set the epocroot path Note, this perl script requires EPOCROOT environment variable to be set. E.g.:

```
set EPOCROOT=S:\Symbian\9.2\S60_3rd_FP1\
```

2. Start HookLogger from Start menu.

3. Select 'Filters' tab and press 'Browse for EXE' button and find your leaking executable you want to analyze. In the combo box select 'Include only checked'.

4. Start the emulator.

5. Start your application in the emulator, reproduce the memory leak, then close it.

6. Check HookLogger's 'Heap' tab then press 'List all allocs' button.

You will see the list of leaked data. With 'Alloc details' you get all the information you need: The call stack, when the leaked memory was allocated!

Note!

Detecting_memory_leaks_with_HookLogger

Remember to restore old EUSER.DLL after using HookLogger, since the modified one is slow and uses lot's of resources!

```
HookEUSER -r winscw
```

making sense of el dump

Most important: if you use this to change down a ALLOC panic, you must mark the offending memory section before hitting the details button. The program will then get a dump like this one:

```
Ordinal350 + 36 - euser.dll! 60037034() Expo + 11900 - EUserParasite_EKA2.dll! 1000dd6c() Expo + 12679  
- EUserParasite_EKA2.dll! 1000e077() Ordinal364 + 32 - euser.dll! 60036ff0() Expo + 11900 -  
EUserParasite_EKA2.dll! 1000dd6c() Expo + 13873 - EUserParasite_EKA2.dll! 1000e521() Ordinal365 + 18  
- euser.dll! 6000f482() Expo + 11900 - EUserParasite_EKA2.dll! 1000dd6c() Expo + 14073 -  
EUserParasite_EKA2.dll! 1000e5e9() >>>>Ordinal122 + 571 - efsrv.dll! 6600208b() Ordinal122 + 52 -  
efsrv.dll! 66001e84() Ordinal176 + 353 - efsrv.dll! 66004bb1() Ordinal176 + 211 - efsrv.dll! 66004b23()  
Ordinal176 + 277 - efsrv.dll! 66004b65() Ordinal76 + 164 - efsrv.dll! 66005034()  
CLocaNoteListView::ConstructL(class TRect const &,class CCoeControl const *,class  
MEikCommandObserver *) <C:\LocaNote\src\LocaNoteListView.cpp #176> + 35 - LocaNote.exe!  
10882195() CLocaNoteListView::NewLC(class TRect const &,class CCoeControl const *,class  
MEikCommandObserver *) <C:\LocaNote\src\LocaNoteListView.cpp #122> + 16 - LocaNote.exe!  
10882037() CLocaNoteListView::NewL(class TRect const &,class CCoeControl const *,class  
MEikCommandObserver *) <C:\LocaNote\src\LocaNoteListView.cpp #98> + 14 - LocaNote.exe! 10881fdc()  
CLocaNoteListViewView::CreateContainerL(void) <C:\LocaNote\src\LocaNoteListViewView.cpp #277> +  
37 - LocaNote.exe! 10881820() CLocaNoteListViewView::DoActivateL(class TVwsViewId const &,class  
TUid,class TDesC8 const &) <C:\LocaNote\src\LocaNoteListViewView.cpp #184> + 8 - LocaNote.exe!  
108813da() Ordinal372 + 67 - avkon.dll! 6c11bd3() Ordinal1330 + 462 - avkon.dll! 6c8748e() Ordinal1330 +  
75 - avkon.dll! 6c8730b() Ordinal1490 + 39 - euser.dll! 60001487() Ordinal1656 + 1200 - euser.dll!  
600020c0() Ordinal1656 + 727 - euser.dll! 60001ee7() Ordinal1656 + 221 - euser.dll! 60001ced()  
Ordinal1656 + 10 - euser.dll! 60001c1a() Ordinal588 + 126 - cone.dll! 4060a4de() Ordinal322 + 503 -  
eikcore.dll! 40b55077() E32Main(void) <C:\LocaNote\src\LocaNoteApplication.cpp #55> + 29 -  
LocaNote.exe! 10883a24() __E32Startup <L:\src\cedar\generic\base\32\euser\epoc\win32\uc_exe.cpp #83>  
+ 5 - LocaNote.exe! 1088ef08() Ordinal274 + 1759 - ekern.exe! 496947f() Ordinal454 + 407 - ekern.exe!  
4967987() kernel32.dll! 7c80b713()
```

It must be read from the bottom to the top. Read, in this case, the offending code is in ConstructL, which is the last user generated on the top

Internal Links

- [Memory Management](#)
- [Debugging techniques](#)
- [Cleanup stack support for non-CBase classes](#)
- [Leave](#)

External Links

- [HookLogger video representation](#)
 - [Tracing Memory Leaks in Symbian](#)
-