Altreonic

Open Event Tracer

25.03.2011

Table of Contents

Revision history

Bernhard Sputh	Creation	25.03.2011

1. Introduction

This document gives a quick introduction to what the Open Event Tracer Installation provides.

2. Provided Examples

This installation provides two examples:

- SemaphoreTracing_SP
- SemaphoreTracing_MP

2.1 SemaphoreTracing_SP

Demonstrates the displaying of traces generated by a single Node (SP). Figure 1 gives a screen shot of Open Event Tracer displaying the trace file provided with this example.

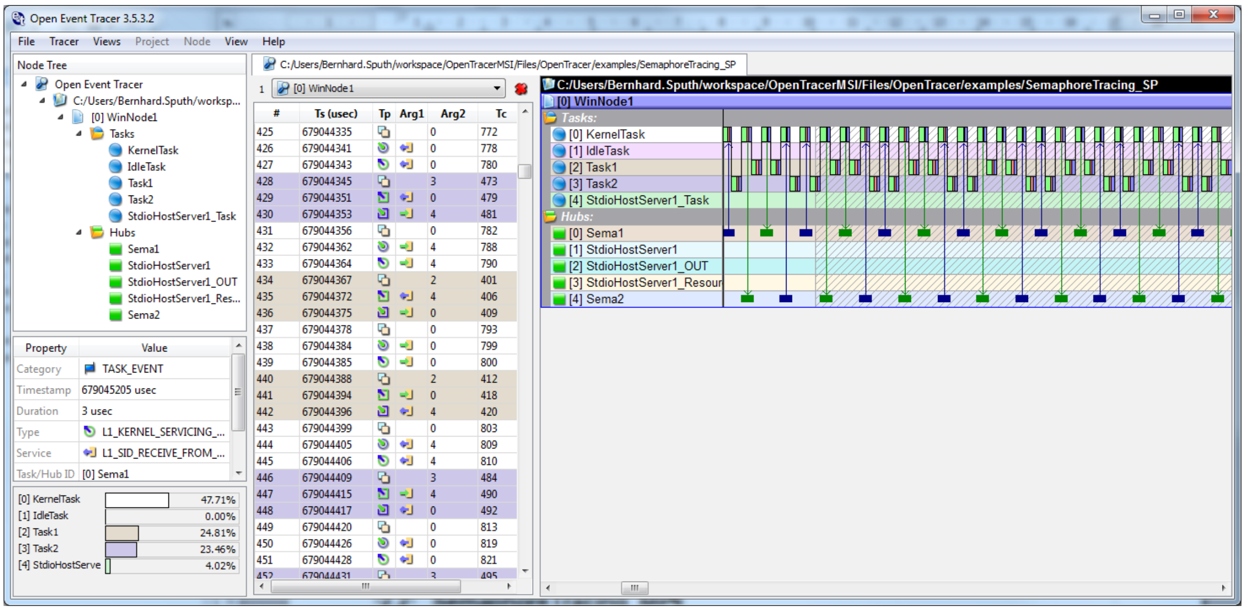
In this example a single Node executes a semaphore loop while collecting trace information. A semaphore loop consists of two tasks: Task1 and Task2, which signal and test two Semaphores: Sema1 and Sema2. The system contains additionally a StdioHostServer1_Task, which is uses to print messages onto the screen, and to write the trace information onto the disk.

An example trace file has been placed in the directory:
“${OpenTracerInstallDir} \examples\SemaphoreTracing_SP\”

The Source code is available below this directory as well, and a pre-built binary that will generate trace-files has been placed in the directory `Output\bin'. Please note that in order to generate a valid trace file, the file OpenComRTOS_Node0.entities must be in the same directory as the executable. Because this file contains header information generated by our code generators.

To take a look at the trace follow these steps:

1. Start Open Event Tracer using the start menu
2. In the main menu go to: File → `Open Node File'
3. In the Open File dialogue select the trace file you want to open.



#	Ts (usec)	Tp	Arg1	Arg2	Tc
425	679044335	⧉		0	772
426	679044341	◎	⇐	0	778
427	679044343	↻	⇐	0	780
428	679044345	⧉		3	473
429	679044351	◩	⇐	0	479
430	679044353	◪	⇒	4	481
431	679044356	⧉		0	782
432	679044362	◎	⇒	4	788
433	679044364	↻	⇒	4	790
434	679044367	⧉		2	401
435	679044372	◩	⇐	4	406
436	679044375	◪	⇒	0	409
437	679044378	⧉		0	793
438	679044384	◎	⇒	0	799
439	679044385	↻	⇒	0	800
440	679044388	⧉		2	412
441	679044394	◩	⇒	0	418
442	679044396	◪	⇐	4	420
443	679044399	⧉		0	803
444	679044405	◎	⇐	4	809
445	679044406	↻	⇐	4	810
446	679044409	⧉		3	484
447	679044415	◩	⇒	4	490
448	679044417	◪	⇐	0	492
449	679044420	⧉		0	813
450	679044426	◎	⇐	0	819
451	679044428	↻	⇐	0	821
452	679044431	⧉		3	495

Property	Value
Category	TASK_EVENT
Timestamp	679045205 usec
Duration	3 usec
Type	L1_KERNEL_SERVICING_...
Service	L1_SID_RECEIVE_FROM_...
Task/Hub ID	[0] Sema1

[0] KernelTask	47.71%
[1] IdleTask	0.00%
[2] Task1	24.81%
[3] Task2	23.46%
[4] StdioHostServe	4.02%

Figure 1: Screen shot of Open Event Tracer displaying the SemaphoreTracing_SP trace.

2.2 SemaphoreTracing_MP

Demonstrates the ability of OpenTracer to link multiple traces to form a single one. Thus one can see what has happened in the system. Figure 2 gives a screen shot of Open Event Tracer displaying the two trace files provided with this example.

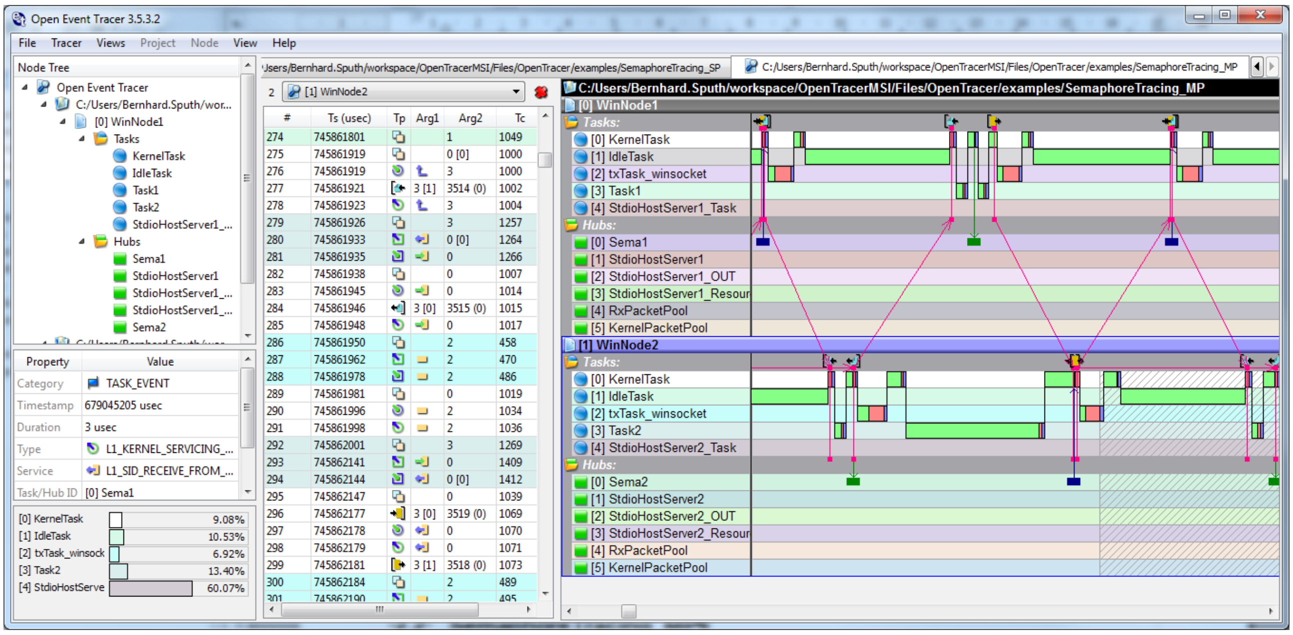
In this example two Nodes execute a semaphore loop, distributed among them, while collecting trace information. A semaphore loop consists of two tasks: Task1 (WinNode1) and Task2 (WinNode1), which signal and test two Semaphores: Sema1 (WinNode1) and Sema2 (WinNode2). Additionally, each node has a StdioHostServer-Task mapped to them, these are uses to print messages onto the screen, and to write the trace information onto the disk.

Two example trace file have been placed in the directory:
“${OpenTracerInstallDir} \examples\SemaphoreTracing_MP\”

The Source code of the example is available below this directory as well, and a pre-built binaries that will generate trace-files has been placed in the directory `Output\bin'. Please note that in order to generate a valid trace file, the files OpenComRTOS_Node0.entities and OpenComRTOS_Node1.entities must be in the same directory as the executables. Because this files contain header information generated by our code generators.

To take a look at the trace follow these steps:

1. Start Open Event Tracer using the start menu
2. In the main menu go to: File → `Open Node File'
3. In the Open File dialogue select the two trace files



Open Event Tracer 3.5.3.2

Node Tree
- Open Event Tracer
  - C:/Users/Bernhard.Sputh/wor...
    - [0] WinNode1
      - Tasks
        - KernelTask
        - IdleTask
        - Task1
        - Task2
        - StdioHostServer1_...
      - Hubs
        - Sema1
        - StdioHostServer1
        - StdioHostServer1_...
        - StdioHostServer1_...
        - Sema2

Property	Value
Category	TASK_EVENT
Timestamp	679045205 usec
Duration	3 usec
Type	L1_KERNEL_SERVICING_...
Service	L1_SID_RECEIVE_FROM_...
Task/Hub ID	[0] Sema1

[0] KernelTask	9.08%
[1] IdleTask	10.53%
[2] txTask_winsock	6.92%
[3] Task2	13.40%
[4] StdioHostServe	60.07%

Users/Bernhard.Sputh/workspace/OpenTracerMSI/Files/OpenTracer/examples/SemaphoreTracing_SP
C:/Users/Bernhard.Sputh/workspace/OpenTracerMSI/Files/OpenTracer/examples/SemaphoreTracing_MP

2 [1] WinNode2

#	Ts (usec)	Tp	Arg1	Arg2	Tc
274	745861801			1	1049
275	745861919			0 [0]	1000
276	745861919			3	1000
277	745861921		3 [1]	3514 (0)	1002
278	745861923			3	1004
279	745861926			3	1257
280	745861933			0 [0]	1264
281	745861935			0	1266
282	745861938			0	1007
283	745861945			0	1014
284	745861946		3 [0]	3515 (0)	1015
285	745861948			0	1017
286	745861950			2	458
287	745861962			2	470
288	745861978			2	486
289	745861981			0	1019
290	745861996			2	1034
291	745861998			2	1036
292	745862001			3	1269
293	745862141			0	1409
294	745862144			0 [0]	1412
295	745862147			0	1039
296	745862177		3 [0]	3519 (0)	1069
297	745862178			0	1070
298	745862179			0	1071
299	745862181		3 [1]	3518 (0)	1073
300	745862184			2	489
301	745862190			2	495

Figure 2: Screen shot of Open Event Tracer displaying the SemaphoreTracing_MP trace.

3. Trace-File Format

Open Event Tracer reads in so called trace-files (extension .trace), see Listing 1 for an example of such a file. These files are plain XML, and consists of the following sections:

- <node> – Gives basic information about the Node (ID, Name)
  - <tasks> – Gives information about the Tasks that have been mapped onto this Node (ID, Name)
  - <hubs> – Gives information about the Hubs that have been mapped onto this Node (ID, Name)
  - <services> – Links Service-IDs with names.
  - <trace> – This encloses the actual trace. Furthermore it gives information about the clock speed of the high and low counters.
    - <event>-- This represents one trace event. It consists of the following attributes:
      - type – The type of the event (Context Switch, Hub Access, Sending a Packet, ...)
      - lowCounter – The low frequency counter value, can be zero.
      - highCounter – The value of the high frequency counter.
      - param1 – event type specific information.
      - param2 – event type specific information.

This is the OpenComRTOS trace file format, but it is possible to adjust it to your specific application.

```
<?xml version="1.0" encoding="utf-8" ?>
<node name="WinNode1" id="0" type="win32">
    <tasks>
        <task name="KernelTask" id="0" type="system"/>
        <task name="IdleTask" id="1" type="system"/>
        <task name="Task1" id="2" type="user"/>
        <task name="Task2" id="3" type="user"/>
        <task name="StdioHostServer1_Task" id="4" type="user"/>
    </tasks>
    <hubs>
        <hub name="Sema1" id="0" type="user" hubType="semaphore"/>
        <hub name="StdioHostServer1" id="1" type="user" hubType="port"/>
        <hub name="StdioHostServer1_OUT" id="2" type="user" hubType="port"/>
        <hub name="StdioHostServer1_Resource" id="3" type="user" hubType="resource"/>
        <hub name="Sema2" id="4" type="user" hubType="semaphore"/>
    </hubs>
    <services>
        <service id="0x0" name="L1_SID_START_TASK"/>
        <service id="0x1" name="L1_SID_SUSPEND_TASK"/>
        <service id="0x2" name="L1_SID_RESUME_TASK"/>
        <service id="0x3" name="L1_SID_STOP_TASK"/>
        <service id="0x4" name="L1_SID_ANY_PACKET"/>
        <service id="0x5" name="L1_SID_WAIT_TASK"/>
        <service id="0x6" name="L1_SID_AWAKE_TASK"/>
        <service id="0x7" name="L1_SID_SEND_TO_HUB"/>
        <service id="0x8" name="L1_SID_RECEIVE_FROM_HUB"/>
        <service id="0x9" name="L1_SID_IOCTL_HUB"/>
    </services>

    <trace lowCounterHz="0" highCounterHz="2337949" >
        <event type="-2" lowCounter="0" highCounter="1587567178" param1="0x207" param2="0x4" />
        <event type="0" lowCounter="0" highCounter="1587567184" param1="0x1" param2="0x2" />
        <event type="-1" lowCounter="0" highCounter="1587567197" param1="0x208" param2="0x4" />
        <event type="1" lowCounter="0" highCounter="1587567203" param1="0x207" param2="0x0" />
        <event type="0" lowCounter="0" highCounter="1587567210" param1="0x1" param2="0x0" />
        <event type="2" lowCounter="0" highCounter="1587567224" param1="0x207" param2="0x0" />
     </trace>
</node>
```

Listing 1: Example trace file