

NEWS 2017

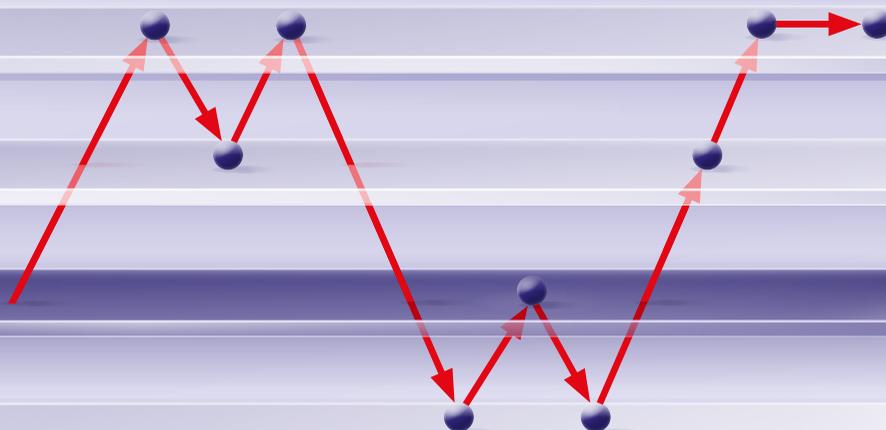
Edizione italiana

APPLICATION

GUEST OS

HYPERVISOR

HARDWARE



*Seamless debugging
through all software layers*

INDICE

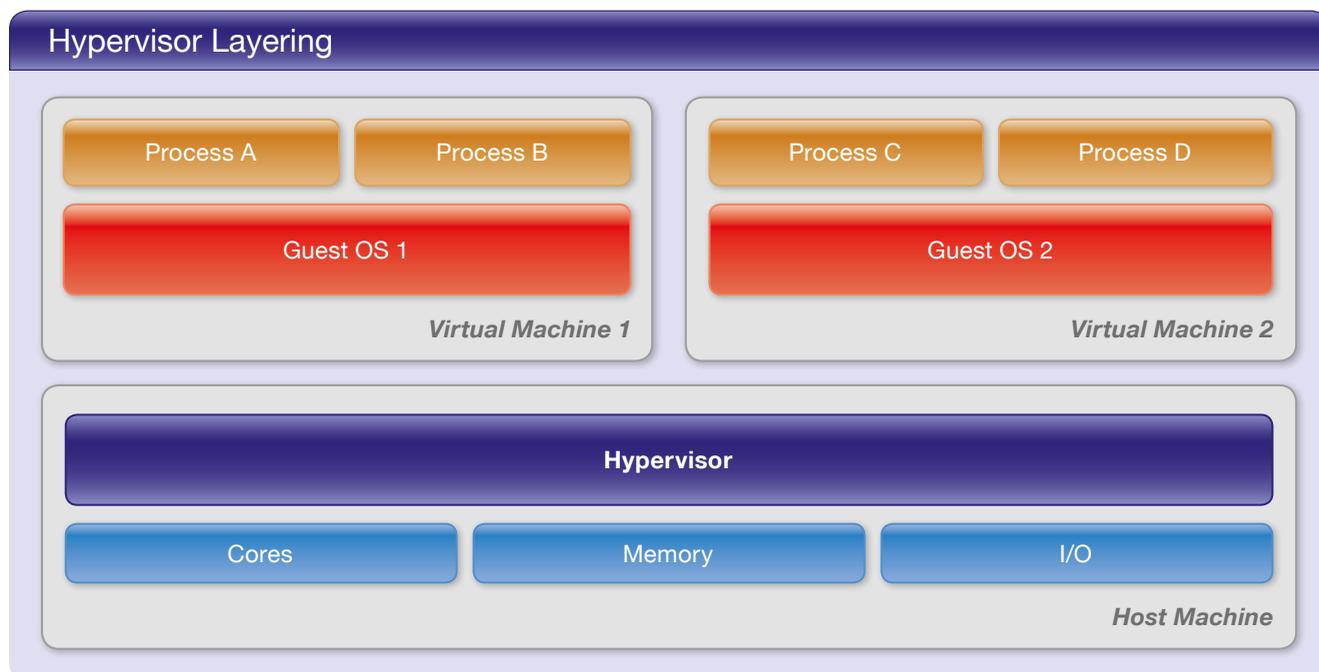
Debug di Hypervisor 2

Tools di debug per Intel® x86/x64 6

CombiProbe per TriCore DAP 7

Nuovo standard AUTOSAR ARTI 8

Debug di Hypervisor



In aprile 2017, Lauterbach rilascerà le nuove importanti funzionalità di supporto agli hypervisor. Questo articolo presenta un'implementazione di riferimento in cui un hypervisor Xen con due guest Linux gira su una scheda HiKey di LeMaker (Cortex-A53).

Virtualizzazione nei sistemi embedded

Il concetto di virtualizzazione permette a più sistemi operativi di girare in parallelo su una sola piattaforma hardware. Oggi la virtualizzazione è sempre più utilizzata anche nei sistemi embedded, per esempio nella plancia di guida di un'auto. Le applicazioni real-time tradizionalmente monitorate da un sistema operativo AUTOSAR vengono fatte girare sulla stessa piattaforma hardware, in parallelo alle interfacce utente basate su Android. Un hypervisor, che rappresenta il nucleo fondamentale della virtualizzazione, assicura che tutto funzioni in modo affidabile ed efficiente.

L'hypervisor, a cui ci si riferisce anche come monitor di una macchina virtuale, è uno strato software che svolge due funzioni:

1. avvia e gestisce le macchine virtuali (VM)
2. virtualizza per le VM le risorse fisiche dell'hardware.

Un sistema operativo che gira su una VM viene chiamato OS guest. Tutti gli accessi eseguiti dai

guest alle risorse hardware virtualizzate sono mappati dall'hypervisor sulle risorse fisiche.

La virtualizzazione delle CPU è importante per il debug. A ogni macchina virtuale vengono assegnate una o più CPU virtuali (vCPU). Il numero di vCPU non deve essere necessariamente uguale al numero di core CPU disponibili nella piattaforma hardware.

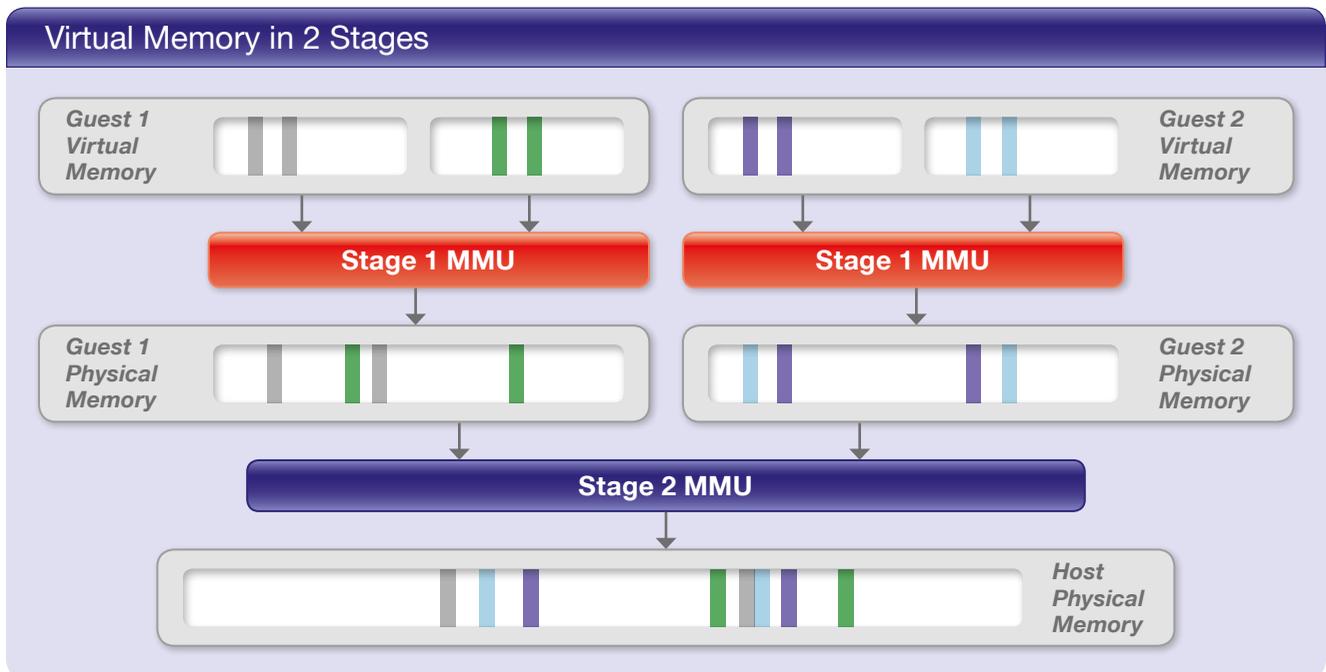
Altrettanto importante è la virtualizzazione della memoria. Le VM non vedono l'effettiva memoria fisica, ma piuttosto una memoria virtualizzata, la cosiddetta memoria fisica guest. Per controllare l'accesso alla memoria fisica l'hypervisor gestisce una tabella di pagine separate per ogni VM. Poiché i processi applicativi, almeno su sistemi operativi come Linux, lavorano già con indirizzi virtuali, nasce l'esigenza di gestire la traduzione degli indirizzi su due livelli:

- da memoria virtuale guest a memoria fisica guest
- da memoria fisica guest a memoria fisica host.

Il diagramma "Virtual Memory in 2 Stages" nell'altra pagina illustra questo concetto:

- l'informazione di mappatura delle "Stage 1 MMU" è gestita dalla tabella delle pagine del corrispondente OS guest.
- la "Stage 2 MMU" usa la tabella delle pagine dell'hypervisor.

Virtual Memory in 2 Stages



Soluzioni per debug esteso

TRACE32 è stato sistematicamente predisposto da Lauterbach nel 2016 per poter offrire ai clienti le più complete funzionalità di debug su hypervisor. Sono state aggiunte le seguenti estensioni:

- nella sintassi dei comandi TRACE32 è stato aggiunto un identificativo “machine ID”, che permette di accedere sia al contesto della VM attiva, sia ai contesti di tutte le VM non attive. Una macchina virtuale è considerata attiva quando è stato allocato un core per la sua esecuzione;
- tramite la nuova funzionalità di supporto degli hypervisor, il debugger ne rileva e ne mostra le VM;
- anziché poter eseguire il debug di un solo sistema operativo, il debug è ora possibile per più sistemi operativi contemporaneamente;
- anziché poter accedere come prima alle tabelle delle pagine di sistema per i guest attivi, il debugger ora può usare anche le tabelle delle pagine di tutti i guest inattivi.

L'obiettivo più importante per tutte le estensioni è stato permettere il debug ininterrotto dell'intero sistema. Ciò significa che, quando il sistema è fermo su un breakpoint, è possibile verificare e cambiare lo stato corrente di ogni singolo processo, di ogni VM e anche dell'hypervisor e della piattaforma hardware reale. È inoltre possibile impostare un breakpoint di programma su qualsiasi locazione di codice.

Il punto di partenza

Le illimitate possibilità di debug che Lauterbach ha offerto per quasi 20 anni per sistemi operativi come Linux, costituiscono il punto di partenza per tutte le estensioni che sono state implementate. È quindi opportuno fare un breve riassunto delle più importanti soluzioni di debug.

I processi girano sui sistemi operativi in uno spazio di indirizzamento virtuale privato. Il supporto di TRACE32 per gli OS e per l'MMU permette agli utenti il debug ininterrotto da un processo all'altro:

- con l'aiuto dello “space ID” è possibile accedere direttamente allo spazio di indirizzamento virtuale di ogni processo;
- con l'aiuto dell'opzione TASK applicata ai comandi, è possibile visualizzare il set corrente di registri e lo stack frame di ogni singolo processo.

Machine ID

Come bisogna estendere questa soluzione se i sistemi operativi girano su macchine virtuali?

1. Anzitutto è necessario identificare in modo univoco ciascuna macchina virtuale. A tal scopo TRACE32 assegna a ogni VM un numero, il machine ID. L'identificativo machine ID degli hypervisor è 0. Come lo space ID viene usato per identificare lo spazio di indirizzamento virtuale di un processo, così il machine

TRACE32 Commands

Traditional OS-Aware Debugging

```
Data.dump <space_id>:<virtual_address>
Data.LOAD.Elf <file> <space_id>:<virtual_address>
Register.view /TASK <process_name>
Frame.view /TASK <process_name>
```

< NEW >

Hypervisor Debugging

```
Data.dump <machine_id>::<space_id>::<virtual_address>
Data.LOAD.Elf <file> <machine_id>::<space_id>::<virtual_address>
Register.view /MACHINE <machine_id> /TASK <process_name>
Frame.view /MACHINE <machine_id> /TASK <process_name>
```

ID è usato per identificare lo spazio di indirizzamento privato di una VM.

2. Per poter mostrare il set di registri e lo stack frame di ogni processo, il debugger deve sapere su quale VM e su quale OS guest il processo sta girando. Per questo è stata introdotta l'opzione MACHINE.

Queste due estensioni sono sufficienti per consentire al debugger di accedere a tutte le informazioni superando i confini di un singolo processo. La presentazione "TRACE32 Commands" riportata qui sopra mostra un confronto fra la sintassi estesa dei comandi di TRACE32 per il debug degli hypervisor, rispetto alla sintassi tradizionale usata per il debug con supporto agli OS.

Supporto agli hypervisor

Analogamente alla funzionalità di supporto agli OS, è ora disponibile una funzionalità di supporto agli hypervisor. Questa prestazione fornisce al debugger tutte le informazioni sull'hypervisor che gira sulla piattaforma hardware. Per il supporto agli hypervisor, è comunque necessario caricare le informazioni simboliche di debug dell'hypervisor. In tal modo il debugger può creare una vista d'insieme di tutti i guest. La schermata "Guest List" per la nostra implementazione di riferimento – Xen, Cortex-A53 – mostra le seguenti informazioni:

- VM ID, stato delle VM, numero di vCPU per VM

Guest List

magic	id	mid	mem	nb_vcpus	vttb	state
000080001007B000	0	3	1536M	8	0001000090044000	blocked
000080000FF31000	1	1	1024M	8	000100008708A000	blocked
00008000070AE000	2	2	512M	8	000200009007E000	running

- indirizzi delle tabelle delle pagine per lo stadio 2 (vttb)

La funzionalità di supporto per un particolare hypervisor è sviluppata da Lauterbach e fornita ai propri clienti. La tabella "Currently Supported Hypervisors" a pagina 5 riporta la lista degli hypervisor finora supportati.

Configurazione del debugger

Come cambia il debug in TRACE32 in seguito all'aggiunta delle soluzioni per debug esteso? Esaminiamo dapprima la configurazione. Per configurare l'hypervisor e ciascuno degli OS guest è necessario:

Debugger Configuration

Hypervisor

- Load debug symbols
- Set up page table awareness (MMU)
- Load Hypervisor awareness

Guest OS 1

- Load debug symbols
- Set up page table awareness (MMU)
- Load OS awareness

Guest OS 2

- Load debug symbols
- Set up page table awareness (MMU)
- Load OS awareness

Guest OS 3

Guest OS 4

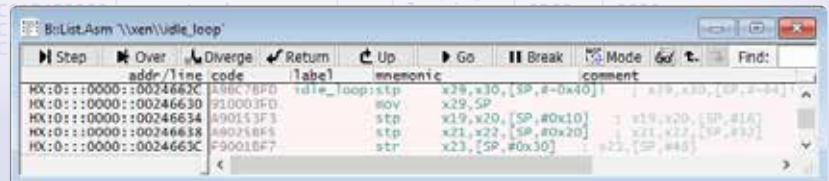
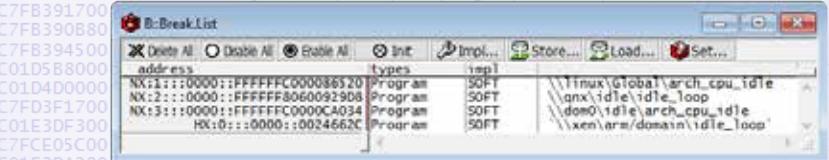
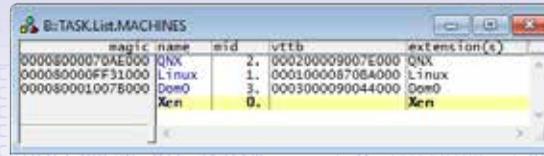
Guest OS 5

Xen Hypervisor on Cortex-A53

Global Task List



Virtual Machine List



1. caricare i simboli di debug
2. impostare il supporto alla tabella delle pagine (MMU)
3. caricare rispettivamente il supporto di TRACE32 all'hypervisor e il supporto per gli OS.

Il diagramma “Debugger Configuration” mostra una vista d’insieme dei diversi passi di configurazione.

Debug dei processi

I servizi offerti da un debugger devono spesso confrontarsi con requisiti opposti. Un gruppo di utenti desidera un’operatività semplice e intuitiva, mentre un altro gruppo richiede un’estrema flessibilità e una completa capacità di scripting. Consideriamo per primo l’utilizzo immediato. In questo caso l’idea fondamentale è molto semplice: se il debugger si ferma a un breakpoint, la GUI deve visualizzare il processo applicativo che ha fatto scattare il breakpoint.

Se l’utente è interessato a un altro processo applicativo, si può semplicemente aprire la lista completa dei task in TRACE32. Qui sono elencati tutti i task in esecuzione nell’intero sistema. Mediante doppio-click sul task, si può selezionare il task che si vuole visualizzare nella GUI. La lista completa dei task offre anche un

modo semplice di impostare breakpoint di programma per uno specifico task. Poiché i simboli di debug vengono associati a un machine ID e a uno space ID quando il file .elf viene caricato, le funzioni e le variabili possono essere indirizzate per nome, come si fa di solito nel normale debug.

La flessibilità estrema e la completa capacità di scripting sono invece ottenute usando i comandi di TRACE32. La sintassi estesa per questi comandi è stata presentata più sopra.

Conclusione

Poiché Lauterbach ha esteso sistematicamente al debug degli hypervisor le note soluzioni per il debug degli OS, gli utenti di TRACE32 non avranno problemi: basterà solo un po’ di pratica.

Currently Supported Hypervisors

KVM	Wind River Hypervisor 2.x
VxWorks 653 3.x	Xen
	<i>more to follow</i>

Intel® x86/x64 – aggiornamento dei Tools

Nel gennaio di quest'anno, Lauterbach ha introdotto il nuovo CombiProbe Whisker MIPI60-Cv2. Il CombiProbe TRACE32 e il QuadProbe TRACE32 ora offrono le stesse prestazioni di debug per il connettore Converged® Intel MIPI60:

- Standard JTAG, Intel® debug hooks con Pmode, e bus I2C
- Unione delle porte di debug (2 JTAG chains)
- Features di Intel® Survivability (threshold, slew rate, ecc.)

Tuttavia questi nuovi strumenti di debug hanno differenti aree di applicazione. Il QuadProbe TRACE32, progettato specificamente per processori server, è un sistema dedicato di debug che permette il debug SMP di centinaia di threads su schede che hanno fino a quattro connettori di debug.

Il CombiProbe TRACE32 con Whisker MIPI60-Cv2, progettato per processori client e dispositivi mobili, in aggiunta alle sue avanzate prestazioni di debug può catturare e analizzare dati di System Trace. Le capacità di trace includono il supporto a una porta a 4-bit e una a 8-bit con larghezza di banda nominale.

Il CombiProbe TRACE32 con Whisker DCI OOB è stato progettato specificamente per debug e trace di dispositivi con stringenti requisiti sui fattori di forma, privi di connettori di debug. Se il chip contiene un Manager DCI, il target e il debugger possono scambiare messaggi di debug e trace direttamente su interfaccia USB3. Il protocollo DCI usato per scambiare messaggi supporta JTAG standard e Intel® debug hooks, insieme a messaggi di trace per registrare informazioni di System Trace.

Debugger and System Trace for Intel® Converged MIPI60 Connector

(devices and client applications)



Debugger and System Trace for USB3 Connector

(all applications)



Debugger for Intel® Converged MIPI60 Connector

(server)



DAP Streaming and TRACE32 Streaming



A partire da ottobre 2016 Lauterbach ha rilasciato il suo nuovo CombiProbe TriCore DAP per la famiglia AURIX™ di Infineon. In questo modo TRACE32 permette ora di svolgere analisi complete di runtime a tutti gli utenti AURIX che lavorano con schede hardware prive di interfaccia AGBT.

DAP Streaming

Il CombiProbe implementa una nuova tecnologia, chiamata DAP streaming: i contenuti della memoria di trace onchip vengono letti mentre il programma è in esecuzione, e sono trasmessi interamente alla memoria di trace del CombiProbe, pari a 128 MB. Per garantire questa prestazione il chip deve fornire un'interfaccia di debug ad alta velocità. L'interfaccia AURIX™ DAP soddisfa i requisiti necessari, con frequenze DAP fino a 160 MHz e data rate fino a 30 MB/s. DAP non offre una larghezza di banda sufficiente a trasmettere l'intero flusso di programma, ma permette comunque un'analisi molto ampia:

- misura del tempo di esecuzione delle funzioni mediante Compact Function Trace (CFT). Si tratta di un tipo particolare di trace di programma in cui

vengono generati dati di trace solo per le chiamate a funzione (cftcall) e per i ritorni da funzione (cftret). Il diagramma "Compact Function Trace" sotto riportato, mostra un esempio di albero di chiamate e informazioni dettagliate sull'esecuzione, calcolate da TRACE32 in base ai dati di trace;

- analisi del contenuto di variabili specificate, al variare del tempo;
- misure di esecuzione per task, ISR e servizi di OS.

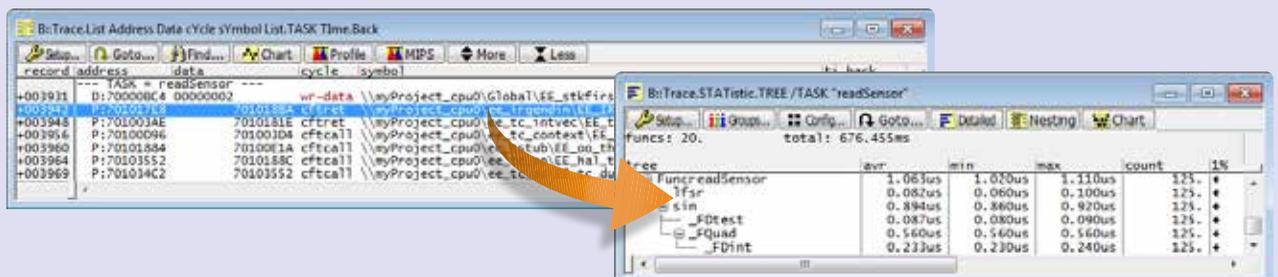
TRACE32 Streaming

Se la memoria di trace di 128 MB presente nel CombiProbe non è sufficiente per registrare tutti i dati di trace d'interesse, è comunque possibile combinare il DAP streaming con il TRACE32 streaming. Lo streaming di TRACE32 trasferisce immediatamente al computer host i dati di trace appena ricevuti dal CombiProbe, e li memorizza poi in un file. Risulta così possibile registrare diversi TB di dati contigui di trace per misure di lungo termine.

Maggiori informazioni a questo riguardo sono disponibili in:
www.lauterbach.com/8467



Compact Function Trace



ARTI – interfaccia Runtime AUTOSAR

Il nuovo standard ARTI sarà progettato specificamente per soddisfare gli attuali requisiti dell'industria automobilistica relativi a debug e trace con supporto all'OS. Lauterbach, in quanto partner ufficiale per lo sviluppo di AUTOSAR, opera attivamente nel progetto di questo standard, la cui pubblicazione è prevista all'inizio del 2018.

Lo standard ORTI, usato nell'industria automobilistica sin dal 2003, aiuta migliaia di sviluppatori nel debug e nella profilazione dei propri sistemi AUTOSAR. Lo standard è divenuto obsoleto nel corso degli anni e richiede opportuni aggiornamenti per soddisfare le attuali esigenze.

Obiettivi

Nuovi metodi di sviluppo software, sistemi multicore e multiECU, crescenti requisiti per la validazione di sistemi critici rispetto al realtime: il nuovo standard deve coprire tutti questi aspetti.

Molte delle soluzioni di debug, trace e profilazione che verranno aggiunte allo standard ARTI sono già in uso oggi come soluzioni proprietarie. Ciò significa che le funzionalità sono già state verificate. Manca solo la standardizzazione delle interfacce fra i diversi strumenti usati nel processo di sviluppo. Segue un esempio basato su un caso reale.

Esportazione dei dati di trace

Dal 2014 Lauterbach collabora strettamente con diverse aziende che producono strumenti di validazione e ottimizzazione del software in ambito automobilistico. TRACE32 esporta i dati di trace registrati in tempo reale, che vengono poi caricati su un prodotto esterno e analizzati in modo completo. Cosa manca dunque in questo processo?

1. Il file ORTI creato dal tool di compilazione contiene informazioni sui task, sui servizi dell'OS e le ISR, ma non contiene informazioni su quando i task iniziano o finiscono, né informazioni sui "runnables". Prima dell'esportazione, è necessario aggiungere manualmente in TRACE32 le informazioni mancanti.
2. Non c'è un formato standard per esportare i dati di trace. Ciò comporta che il prodotto esterno deve essere in grado di leggere il formato proprietario di TRACE32.

Il nuovo standard ARTI risolverà entrambe queste mancanze. Il file ARTI prodotto dal tool di compilazione conterrà informazioni su tutti gli oggetti AUTOSAR, standardizzando allo stesso tempo l'esportazione dei dati di trace.

Conclusione

Dato che tutti i più importanti produttori di strumenti, nonché gli utenti di questi strumenti hanno lavorato insieme nella redazione del nuovo standard ARTI, sicuramente il nuovo standard avrà successo e durerà a lungo, come il suo predecessore.



I.P.

Se il vostro indirizzo email è cambiato o non volete più ricevere le nostre newsletter, mandate una email a mailing_it@lauterbach.com

