Kirkegaard Winkler (Kirkegaard)

A SILICON BUG, L'EPILOGO.

19 November 2011

La coscienza e' il caos delle chimere, delle cupidigie e dei tentativi, la fornace dei sogni, l'antro delle idee di cui si ha vergogna; e' il pandemonio dei sofismi, e' il campo di battaglia delle passioni. Penetrate, in certe ore, attraverso la faccia livida d'un uomo che sta riflettendo, guardate in quell'anima, in quell'oscurità; sotto il silenzio esteriore, vi sono combattimenti di giganti come in Omero, mischie di dragoni ed idre e nugoli di fantasmi, come in Milton, visioni ultraterrene come in Dante. Oh, qual abisso è mai quest'infinito che ogni uomo porta in sé e col quale confronta disperatamente la volontà del cervello e gli atti della vita! (Victor Hugo)

Guardo fisso il quadrato rosso che sapevo non sarebbe mai diventato verde. Quel parallelepipedo rosso è il modo con cui il sistema di sviluppo dello **IAR** segnala la presenza di un **breakpoint** nel codice.

Quel quadratino è un po' come una tappa parziale dove si scatta una istantanea all'atleta che la raggiunge. In questa metafora figurativa il microcontrollore, che corre sul codice, una volta raggiunto il traguardo intermedio si ferma e in quella attesa si visualizzano le proprietà dei registri interni.Alcuni registri hanno a loro volta i valori evidenziati di un colore rosso e anche questa è una buona utility che ti permette di capire quali sono stati aggiornati dall'ultimo **break**.

Da tre giorni armeggio dentro quella procedura. Ho cambiato lo stato di tutti i registri con il preciso intento di verificare le funzionalità alternative che il micro mi mette a disposizione ma il vero problema era che non avevo la più pallida idea di come uscire da quello stato di impasse. Anche io ero dentro un quadratino verde. Fermo, immobile, stanziale nella peggiore delle posizioni, quella condizione infame di non avere strade da percorrere. Un vicolo cieco. Il tempo in quei momenti si rallenta... i suoni si fanno vacui mentre entri in una palla di vetro. Tutto intorno a te si muove velocemente ed è frenetico. Tu invece rallenti i movimenti e addirittura li ripeti senza controllo... Il micro si blocca e poi dopo qualche secondo resetta e poi riincomincia..... si blocca ... resetta .... e rincomincia in un ciclo infinito. Quell'unico **led** sulla scheda si spegne per un attimo durante il reset per poi riaccendersi nella interminabile sequenza. Squilla il telefono, in amministrazione non si accende un monitor e la segretaria non può lavorare....

Una delle parti dove il codice nei giorni precedenti si bloccava era in un modulo assembly che viene fatto partire allo startup prima del main. Questo modulo ha al suo

interno due sezioni di codice che devono inizializzare due zone di memoria che lo IAR mette a disposizione. Queste due sezioni si chiamano

```
DATA16_I
DATA16_Z
```

Sostanzialmente lo IAR prevede due tipi di variabili, quelle che vengono inizializzate a zero e quelle con valori diversi da zero. Le variabili inizializzate a zero venivano resettate utilizzando una funzione di libreria standard del c chiamata **memset**. Le variabili invece inizializzate con valori diversi da zero utilizzano una altra funzione di libreria chiamata **memcpy**. Con la nuova releasa dello IAR, necessaria per utilizzare il nuovo microcontrollore, la funzione memcpy falliva andando in deadlock (che noia...).
Disassemblo il codice e vado a vedere al suo interno fin dove effettivamente si blocca. Passo, passo, opcode su opcode... sembrava che i registri fossero inizializzati a casaccio. Allora modifico il main e provo la memcpy. Disassemblo e strabuzzando gli occhi vedo che effetivamente l'inizializzazione della memcpy è diversa da quella che veniva implementata nel modulo assembly. Cosa era successo ? Niente di più banale. La versione aggiornata dello iar aveva rivisiato tutte le librerie interne e riiscritto i moduli. La riscrittura delle funzioni di libreria prevedeva un diverso passaggio dei parametri. Nel mio caso il parametro della lunghezza non veniva passato più sullo **stack** ma veniva passato attraverso un registro del micro.

Il mio predecessore per altri motivi che non sto a spiegare aveva fatto un merge di questi file e aveva implementato queste funzioni manualmente. Così visto che non siamo qui a lavare gli scogli della battigia ho riiscritto le due funzioni di memcpy e memset per essere sicuro che non ci fossero più problemi con le librerie al prossimo aggiornamento del compilatore.

```
        MOV     #SFB DATA16_Z, W0
        MOV     #sizeof DATA16_Z, W1
        mov.w   #00,R15

?memset:
        mov.w   R12,R10
?memloop1:
        mov.b   R15,0x0(R10)
        inc.w   R10
        dec.w   R14
        tst     R14
        jnz     ?memloop1
```

Questa è invece la memcpy

```
       MOV     #SFB DATA16_I,  W0
       MOV     #SFB DATA16_ID, W1
       mov.w    #sizeof DATA16_I,R15

?memcpy:
       mov.w   R12,R10
?memloop2:
       mov.b   @R14+,0x0(R10)
       inc.w   R10
       dec.b   R15
       tst     R15
       jnz     ?memloop2
```

Ma riprendiamo la nostra storia dopo questo breve **flashback**. Mi alzai stirandomi la schiena e andai in bagno utilizzato sempre come personale ufficio per i casi estremi. Il bagno in quella azienda era identico a quello delle carceri americane, chiuso, senza finestre con condutture scatolate appese sul soffitto che fungevano da aspiratori. Lampade al neon morenti lampeggiavano gli ultimi istanti di vita. Mi siedo sulla tavola abbassata considerando appunto che sono li per pensare o forse no.... forse sono li per non pensare. Le porte sono sfondate ci sono due bagni rotti, poi la luce intermittente rende l'ambiente davvero surreale.

Il codice era collaudato da migliaia di installazioni effettuate su un progetto precedente che aveva una piattaforma hardware identica e invece in questa vita nefasta su quel nuovo microcontrollore si bloccava, continuamente... Ripresi il datasheet e cercai di interpretare i flag che non venivano utilizzati dal mio codice, ma che era possibile modificare implementando nuove funzioni. All'interno di un registro di configurazione delle modalita operative del timer c'era un flag chiamato di sync utilizzato per allineare gli eventi del gate aperto dal quarzo esterno con lo stato del timer aggiornato direttamente dal **DCO**. Se lo resetto questo flag di SYNC l'evento scatta subito sbloccando il micro dal dead lock fatale. Osservo e allargo la sottofinestra dei registri interni. Il valore del timer alla fine della procedura è vuoto non ha contato nemmeno un fronte, non c'era niente da fare nemmeno quella era la strada da percorrere. Un giorno.... e poi un altro ... e poi un altro ancora. Ho un pesante sospetto nella testa ma non posso credere che ci sia un baco nel silicio. Somatizzo la situazione e resetto anche io in continuazione, guardo il led verde di un cellulare gsm appoggiato sul tavolo che lampeggia velocemente e capisco che si è appena registrato sulla rete, forse dovrei fidarmi più di me stesso, non c'è niente di sbagliato nel codice.....

Passa ancora un giorno e alla fine esausto chiamo il **FAE** della famiglia di questo microcontrollore. Loro ne sentono di tutti i colori e francamente non so come ma riesco a convincerlo che sono in una fossa delle marianne di liquami, allora in un atto di benevolenza mi mette in contatto con il tecnico tedesco. Quello che contatterò via e-mail sarà un super tecnico europeo della Texas Instrument. Gli descrivo il problema e gli spedisco la funzione maledetta, per cercare di agevolarlo. Dopo qualche ora una prima doccia fredda, il tedesco risponde in un inglese stentato che nel suo sistema la funzione non si blocca e mi rispedisce il codice purificato implementato. Apro la email e scarico l'allegato. Guardo il suo progetto ma in quel momento scopro una differenza importante.
La lettura della frequenza DCO il tedesco l'aveva implementata utilizzando come clock di sistema il principale proprio DCO stesso mentre io invece utilizzavo il clock esterno **XT2**. Cosa c'entrava questo con il dead lock ? Perchè il timer non segnava nulla utilizzando come sorgente il DCO ? Non c'era niente di logico in quello che vedevo e non c'era niente di associabile fra le due frequenze master e il Timer. Metto a conoscenza il tedesco del problema che ho visto sulla mia scheda implementando il suo codice e dopo un giorno mi conferma che anche lui (deo gratias) aveva visto il blocco del codice utilizzando come clock di sistema l'XT2 esterno. A questo punto anche lui riconobbe l'estrema anomalia e ammise il limite delle sue competenze e così purtroppo non poteva far altro che passare il problema direttamente ai progettisti del silicio del micro. Dopo una settimana arrivò il responso.... ero incappato nel peggiore incubo che un firmwarista può incontrare nella sua carriera il **Silicon Bug**.

Il problema dei silicon bug è che sono difficilissimi da diagnosticare perchè tipicamente e (fortunatamente) vengono fuori in condizioni di utilizzo estreme. Il problema è che quando si arriva a concludere che c'è un baco nel silicio del micro (e non nel silicone) sapete cosa accade ? Nulla. l tecnici del silicio della Texas riconobbero il problema sulla attivazione del DCO come clock del sottosistema e mi inviarono un round tips che avevo già scoperto. Le tre settimane precedenti svaniscono in un attimo, come quando ti risvegli da un incubo. Ma adesso che sei sveglio ti accorgi che hai un altro problema, spiegare a Mangusta il perchè di quel ritardo che oramai aveva superato il mese. Spiegargli il silicon Bug ? Rinunciai. Mi presi un altro po' di pesciate d'ordinanza che invece non potevo rinunciare.

Esco dal parcheggio e imboccando la strada che poi arriva alla statale, guardo i capannoni che circondano la zona. Uno era un gigantesco deposito di giocattoli cinesi chiuso cinque anni fa e rimasto tale dopo l'ingiunzione fallimentare, a fianco un altro abnorme capannone scheletrico; ne avevano alzato le pareti prefabbricate e i pilastri portanti ma il fatale fallimento era arrivato prima di raggiungere il tetto. Mi giro di lato e vedo il furgone degli zingari che attraversa come una sentinella la zona in cerca di rame. Poco più in là un call center arnia con un continuo andirivieni di servi della globalizzazione. Ma quella sera era come dire... diverso. Ero molto più rilassato in quel posto di cemento e acciaio e con rammarico ripenso alle sere precedenti... che

avevo buttato via. Metto la freccia e mi fermo in una enoteca perché un collega mi ha parlato di un bolgheri blasonato che non ti depaupera le tasche e compro così un Guidoalberto del 2003. Piove ma non mi importa...

In questo **link** potrete vedere lo scambio di messaggi accaduto realmente fra un utente con uno strano nome Kirk&gaard (davvero questi ragazzi hanno una gran fantasia... ;-)...) con il FAE tedesco della Texas che poi ha dato spunto a questo raccontino

Estratto da "http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Kirkegaard:a-silicon-bug-l-epilogo"