

Lauterbach TRACE32 sulautetun Linux-projektin debuggaamiseen

Sulautetuista järjestelmistä on koko ajan tulossa aiempaa monimutkaisempia erityisesti silloin, kun järjestelmien pohjana käytetään Linuxia. Suunnitteluprosessin vaikeimmin ennustettavissa oleva ja ehkä myös kaikin kriittisin osa on debuggaus.

Yksi vaikeimmista asioista sulautetun järjestelmän suunnittelussa on ymmärtää, miten järjestelmä käyttäytyy. Debuggerit ovat työkaluja, joilla kehittäjä näkee mitä Linux-järjestelmässä tapahtuu.

Linux-järjestelmän debuggaukseen on periaatteessa kaksi eri mahdollisuutta: ohjelmistopohjainen ja laitepohjainen tapa. Laitepohjaisissa debuggereissa, eli tehokkaissa JTAG-pohjaisissa debuggereissa kuten Lauterbach TRACE32 on enemmän toimintoja ja mahdollisuuksia ohjata järjestelmää kuin ohjelmistopohjaisissa kuten GDB. Seuraavaksi voimmekin kysyä, että millä tavalla ne ovat parempia?

Täysi läpinäkyvyys käyttäjätason ja ydintason välillä

Koska Lauterbachissa on sekä Linux- että MMU-tuki, kernelin debuggaus sekä debuggaus yli prosessorirajojen on mahdollista. Lauterbach debuggerilla voidaan aloittaa prosessin tai säikeen debuggaus heti, kun se on alustettu kernelissä.

GDB:n tapauksessa debuggeri on osa kohdeohjelmaa, eli kaikki ohjelmistorajoitukset koskevat myös debuggeria. Prosessorin MMU "näkee" vain kernelin osoitetilan ja käynnissä olevan kohteen osoitetilan.

Tästä syystä debuggerilla on pääsy vain ohjelman näihin osiin. Sillä ei ole pääsyä vielä käynnistämättömiin prosesseihin.

Ei kuormita järjestelmää eikä tuo lisäkoodia

JTAG-debuggeri, kuten Lauterbach, ei lisää ylimääräistä koodia järjestelmään eikä käytä resursseja, kuten CPU:n tehoa tai muistia.

GDB-debuggeri taas vaatii, että kohdejärjestelmässä on agenttiohjelma jokaiselle debuggattavalle prosessille.

Fyysinen ja looginen muisti täysin hallittavissa

Tämä on erityisen tärkeää kehitettäessä ajureita, sillä ainoa vaatimus "Stop Mode Debuggaukselle" on toimiva JTAG-rajapinta jolloin debuggaus voi alkaa reset vektorista.

Tämä merkitsee sitä, että koko Linux-järjestelmän tai sen osan profilointi on mahdollista tunkeutumatta järjestelmään.

Segmentointivirheiden tunnistaminen ennen niiden suorittamista

Oletetaan, että prosessissa tapahtuu ajoittain segmentointivirhe. Lauterbach-debuggerissa kernel alueelle voidaan asettaa keskeytyskohtia käsittelemään segmentointivirheitä.

Myöhemmin segmentointivirheen tapahtuessa Lauterbach keskeyttää suorituksen, kun järjestelmä on suorittamassa segmentointivirheen. Koska nämä keskeytyskohdat ovat ohjelmallisia ja JTAG-debuggeri käsittelee ne, Lauterbach keskeyttää suorituksen ennen segmentointivirheen suorittamista.

Näin on mahdollista tutkia muistialueita, paikallisia muutujia, rekistereitä sekä koodin sitä osaa, joka aiheuttaa segmentointivirheen. GDB-debuggeri ei pysty tähän, se kaatuu yhtä todennäköisesti kuin sovelluskin segmentointivirheen vuoksi.

Tarjoaa useita skenaarioita ja liitännäismahdollisuuksia kohdejärjestelmään

Ohjelmistopohjaiset debuggerit, kuten GDB, liittyvät kohdejärjestelmään tavallisesti standardiväylällä, kuten sarjapäivälyllä tai Ethernetillä. Tämä edellyttää, että kohdejärjes-

telmän on oltava käynnissä ja vähintään käytetyn väylän keskeytyslinjojen on oltava käytössä. Bootstrap-, keskeytys- tai järjestelmän kaatumisen jälkeinen debuggaus ei siis ole mahdollista.

Lauterbach-debuggeri ei vaadi kohteelta ohjelmistoresursseja vaan käyttää debuggaukseen joko JTAG/BDM-porttia tai "mictor"-liitäntää. Näin on mahdollista käyttää sarjavylyä tai Ethernetiä muihin tarkoituksiin kuin debuggaukseen.

Voidaan asettaa monimutkaisia keskeytyksiä järjestelmää häiritsemättä

Lauterbach-debuggerilla voidaan asettaa keskeytyskohtia kaikkialle ja kaikille tasoille. Ne voivat liittyä tehtäviin, olla säiekohtaisia, ne voidaan asettaa tietyille funktioille sekä loogiseen tai fyysiseen muistiin niin käyttäjä- kuin ydintasolle.

GDB-debuggeria käytettäessä prosessin keskeytyskohta pysäyttää vain kyseisen prosessin. Ydin ja kohdejärjestelmän muut prosessit jatkavat toimintaansa.

Tästä saattaa olla apua, jos protokollapinojen toiminnan pitää jatkua debuggauksen aikana, mutta se haittaa prosessin välisten liitäntöjen debuggausta.

Lauterbach-debuggerin käyttö GDB-debuggerina

Joissakin tapauksissa halutaan käynnistää ja pysäyttää prosesseja toisistaan riippumatta. Tästä syystä Lauterbach-debuggeria voidaan käyttää myös ohjelmistodebuggerina, jolloin se toimii kuten GDB.

Tästä saattaa olla apua esimerkiksi silloin, kun protokollapinojen toiminnan pitää jatkua debuggauksen aikana, debuggattaessa prosessien välisiä liitäntöjä ja luonnollisesti silloin, kun prosesseja pitää käynnistää ja pysäyttää toisistaan riippumatta.

Toteutus on periaatteessa sama kuin tavallisen GDB:n, eli jokaista debugattavaa prosessia kohti käynnistetään yksi GDB:n ilmentymä.

Vaatimukset debuggattavalle laitteistolle ja ohjelmistolle

Kohdejärjestelmään on pystyttävä liittymään JTAG/BDM- tai mictor-liitännällä ja koko Linux-kernel on käännettävä debug-symbolit päällä. Lisäksi "vmlinux"-tiedoston symbolien on oltava ladattuna Lauterbachin muistiin.



JTAG-Debugger



Real-Time-Trace



Logic Analyzer

**Lisää tietoa,
ota yhteyttä:**

Leevi Lehtinen

leevi.lehtinen@nohau.fi,

+358 40 546 1469