

OSADL RTLWS 2009 - 2. časť

Fodrek Peter · Informačné technológie

16.10.2009



V predošlej časti som informoval čitateľov o prvom dni konferencie RTLWS. V prvom dni som sa značne vyčerpal dvoma prednáškami a tak som sa tešil na deň druhý. Ten som celý strávil ako poslucháč zaujímavých prednášok. S rozhodovaním ktorú navštíviť som nemal problém, nakoľko druhý deň sa odohrával len v jednej sekcii.

Prednášky druhého dňa sa sústredili na reportovanie uzavretej akcie vývojárov *RT Preempt Patch*. Uzavretej akcie sa zúčastnilo asi 11 ľudí. Táto uzavretá skupina ľudí sa v predošlých dňoch bavila o budúcnosti *RT PP*, snažili sa určiť smerovanie a stratégiu presadzovania *RT PP* do hlavnej časti jadra.



Obr. 1. Uzavretá skupina - *RT kernel mini-summit*

Informovaniu o výsledkom uzavretého jednanja bol venovaný temer celý druhý deň. Thomas Gleixner rozprával o spolupráci s ostatnou komunitou pracujúcou na vývoji jadra. Načrtol hlavné témy vývoja patchu so zameraním na sleeping spinlocks a veľký zámok jadra (*big kernel lock*). Akademická obec sa zamerala skôr na teóriu plánovania s dodržaním konečného času vykonania procesu (*dead line scheduler*).

Po úvodnej prednáške Thomasa Gleixnera sa k slovu dostal Jonathan Corbet - šéfredaktor [Linux Weekly News](#) a hlavný štatistik jadra. Jeho prednáška pojednávala o hlavných zmenách v jadre od verzie 2.6.22 do 2.6.31 (aktuálna verzia z 9.9.2009). Jednalo sa teda o prehľad za viac ako dva roky. Medzi iným spomenul aj časti, ktoré nemali byť vôbec zaradené do hlavnej vetvy jadra, no napriek tomu boli. Jedná sa o časti *memory managementu* (Ingo Molnár) a *spiacich spin lockov*.



Obr. 2. Úvod prednášky Jonathana Corbeta

Krátky coffee break oddelil prezentácie Corbeta od Clarka Williamsa. Jeho prednáška *Real-time Linux in the enterprise* porovnávala požiadavky firiem na real-time s akademickou teóriou reálneho času. Ukázal, že je rozdiel v chápaní požiadaviek priemyslu a teórie operačných systémov. Tvorcovia operačných systémov sa zaoberajú latenciami, algoritmami plánovania, zatiaľ čo priemysel zaujímajú veličiny ako *deadline* (maximálny čas ukončenia úlohy). Okrem iného spomenul aj možnosť merať latenciu pri každom spustení časovača a po skončení úlohy zabezpečiť jej konštantnú dĺžku vrátane latencie.

Paul McKenney predložil svoju prednášku o 30 minút. Hovoril predovšetkým o problémoch s reálnym časom na multijadrových a multiprocessorových systémoch. Zistil, že real-time latencia výrazne narastá pri použití viac ako 8 jadier a pri 16 je už jej veľkosť neprijateľná. Naproti tomu nereal-time latencia je rovnaká od jedného až po 128 jadier. Jedná sa o pozoruhodné zistenie. Ako možné riešenia navrhol použitie plánovacích domén, alebo centrálného plánovača s viacerými frontami. Ďalšou možnosťou je modifikácia algoritmu *earliest deadline first*. Jedná sa o stratégiu plánovania, kde na procesor je preplánovaná úloha, ktorá ma skončiť v najbližšom čase. Modifikácia znižuje vplyv „klamania“ procesu o čase svojho ukončenia. Matematický základ metódy je enormne komplikovaný. Po náročnej prednáške sme sa odobrali na obed.



Obr. 3. Pohľad na predsálie prednáškových miestností

Poobednú časť programu zahájil Darren Hart. Predviedol modifikáciu dynamicky linkovanej knižnice, ktorú využíva každý program v jazyku C (*glibc*), aby lepšie zodpovedala realnečasovým požiadavkám. Zaoberal sa najmä problematikou inverzie priorít, čo je hlavná požiadavka na hard real-time.

Po Hartovi sa ujal slova *kernel hacker* Peter Zijlstra. Predstrel všetky dôvody ktoré doposiaľ bránia dosiahnutia hard real-time, napriek tomu že sa tento problém rieši od sedemdesiatych rokov 20. storočia. Načrtol ako dôvody hardvérové tak aj softvérové. Spomedzi softvérových je to hlavne rozpor medzi požiadavkami na činnosť desktopu a hard real-time. Desktop požaduje odozvu, ktorá v priemere neprekročí stanovený interval, ale jednotlivito ho prekročiť môže. Hard real-time nedovoľuje žiadne prekročenie. V hardvérových dôvodoch sa odvolával na neprítomného Prof. McGuire. Tieto dôvody podrobne rozobral nasledujúci rečník Peter Okech.

Okech prezentoval metódy merania hardvérových latencií. Potom prezentoval prácu, ktorou sa uchádza o doktorandské štúdium (dostať sa v Keni na doktorandské štúdium nie je tak jednoduché) u Prof. McGuira. V práci opísal zdroje neurčitostí v jadre. Za základný zdroj označil *big kernel lock*, ktorý sa nachádza v ešte 220 zdrojových súboroch. Aj napriek odstráneniu 70% výskytov *BKL* je toto číslo ešte stále vysoké. Ďalším zdrojom sú *SMI* (*system management interrupt*). Tieto vznikajú v systéme náhodne napr. pri požiadavke na zmenu frekvencie procesora, zbernice alebo inej vnútornej správy systému. Pri vypnutí podpory *SMI* v jadre sa neurčitosti minimalizovali tak, že maximálna latencia bola do 5µs, ale po zhruba dvoch hodinách prevádzky systém zmrzol. *SMI* nemá kompletnú dokumentáciu a každý výrobca implementuje svoju verziu. Činnosť by mal zabezpečovať BIOS alebo EFI, ale často tak nerobí a necháva ich obsluhu na operačný systém. Preto je nevyhnutné mať *SMI* zapnuté.

Nasledoval ďalší *kernelový hacker* Steven Rostedt, ktorý sa zaoberal nástrojmi na meranie hardvérových latencií. Špeciálne šlo o nástroj *Ftrace*. Ďalším používaným nástrojom bol *hwlat*. Predviedol rôzne modifikácie nástrojov a ich možnosti použitia pri meraní latencií v hard real-time systéme.

Druhý deň prednášok zakončil šéf OSADL Carsten Emde. Pokračoval tematicky v prednáške predrečníka Rostedta. Praktické príklady predošlej prednášky teoreticky podložil. Záverom pozval všetkých účastníkov na spoločnú večeru v jednej z odporúčaných reštaurácií.

Reportáž z posledného dňa OSADL RTLWS 2009 prinesieme v ďalšej časti.

