

Wiki předmětu

15. května 2022 15:20:36

Login xdobro23, **Dobroň Samuel**, 2. ročník BITP, prezenční, FIT

Ak. rok 2021/2022 studium č.1: řádný zápis

[Zpět na termíny](#)

[[IAN Home]]

Navigace

- [Hlavní stránka](#)
- [Poslední změny](#)
- [Nápověda](#)

Stránka

Zobrazit

Hledat

Nástroje

- [Odkazuje sem](#)
- [Nahrát soubor](#)
- [Seznam souborů](#)
- [Seznam stránek](#)
- [Historie stránky](#)

[Stránka](#)[Editovat](#)[Historie](#)[Změny](#)[Sledovat](#)

4. projekt - SystemTap

Napište systemtap script, který bude zbírat a vypisovat nasledovné dat

- Scriptu zadáte popri spúšťajúcom "stap" príkaze aj meno procesu (str (vo zvyšku zadania ho budem referencovať ako "target_process").
- Váš script bude monitorovať všetky procesy, ktorých názov obsahuje string, tzn. aj ako substring.
- Vypíšte vždy, keď niektorý target_process vytvorí nový process (systé a clone()).
- Vypíšte vždy, keď nejaký process spustí program s target_process v r volania execve() a execveat()).
- Monitorujte čas, ako dlho každý target_process reálne bežal (actively každom CPU.
- Monitorujte čas, ako dlho bol každý target_process na každom CPU p časovačmi/prerušeniami (hrtimer_interrupt). Počas prerušenia bude p "nebežiaci", tzn. čas prerušenia sa logicky nepočíta ako run-time. HIN výstupu, si prečítajte ako tieto veci merať.
- Štatistiku monitorovania run-time a prerušenia vypíšte každých 5 sekú scriptu.
 - Údaje monitorovania vypíšte v celočíselných percentách, relatívne = plných 5 sekúnd run-time). POZN: V systemtap scripte sa nedaj desatinné čísla (netriviálny prístup ku floating-point unit), takže te aritmetikou obísť. (stačí integer aritmetika)
 - Časová presnosť (granularita) údajov stačí v mikrosekundách us.
 - Ak zaznamenaný čas prerušenia nepresiahne aspoň 1% (čož bude v celých mikrosekundách.

POZN: Konkrétny formát výstupu je čiste na vás, ale musia z neho byť i vyžadované informácie.

- Váš script musí fungovať (a my ho budeme testovať) na verzi kernelu 4.18.0-348.12.2.el8_5.x86_64, ktorú máte nainštalovanú na vašich v "guru módu"!

HINT: Pre nastavenie 5-sekundového timeru a vytvorenie probe na konk pozrite na "man stapprobes" (na seminárnych skupinách sme vám tieto neukázali, ale není to nič komplikované :))

POZN: Keďže môžu stap probe bežať paralelne na viacerých CPU, prirodz "race condition". V rámci tohoto zadania si nemusíte robiť robiť starosť s odchylkami (napr. keby vám 1 z 20 výpisov ukázal 101%, tak OK). Dôle: probe bola v zmysle tohoto paralelizmu korektná.

Príklad textového výstupu scriptu:

Example output from process CPU usage monitoring of "stress

```
=====
Scheduling stats:
CPU    PID    run    hrtimer    comm
  0    82896    99      24754 us    stress
  1    82899    49      7030 us    stress
        82898    49      6953 us    stress
  2    82901    31      4408 us    stress
  3    82901    68      9351 us    stress
=====
// next output after 5 seconds
=====
Scheduling stats:
CPU    PID    run    hrtimer    comm
  0    82896    99      24995 us    stress
  1    82898    49      7465 us    stress
        82899    49      7584 us    stress
  3    82901    99      13883 us    stress
=====
```

Example of syscall outputs:

```
PID 89374 executed execve syscall for pathname: "/usr/local
PID 89374 called clone syscall
PID 89374 called clone syscall
```

Keďže toto zadanie vyžaduje určitú znalosť interných funkcií kernelu, čož predmetu, tu máte pár užitočných informácií vrátane mien a popisu funk mohlo hodiť napojiť vaše stap-proby:

- Zmena aktuálne bežiaceho (actively running / current) procesu na CP rutinou "schedule()". Keď kernel zavolá funkciu schedule(), vymení d ďalší vybraný podľa plánovacieho algoritmu. V podstate môžete počíta na CPU od schedule() po schedule(), napr. "process-A -> schedule() -> process-A -> schedule() -> ..."
- Samotná funkcia s menom "schedule", je ale assembly linkovaná (a t ako wrapper pre hlavnú funkciu "__schedule()". Vaše proby teda môž "__schedule@kernel/sched/core.c".
- Prerušenie (interrupt / IRQ) je viacero typov. V tomto zadaní sa máte : resolution timer". Jedná sa o nastavené časovače, ktoré majú na star funkcie alebo aj prípadný procesing niektorých asynchronných event môžete využiť funkciu "hrtimer_interrupt@kernel/time/hrtimer.c" a p vykonávaného procesu či rutiny je prerušený konkrétne po celú dobu

Samozrejme berte do úvahy, že máte k dispozícii celú radu tapsetových probe, takže je len na vás, aké konkrétne proby naprogramujete (nemus zmienené funkcie). Zoznam všetkých užitočných manuálových stránok r je vypísaný v tabulke "man stap".

Prípadný zdrojový kód jednotlivých tapsetových funkcií a probe nájdete i /usr/share/systemtap/tapset/linux. Zdrojový kód samotného kernelu-4.1 na VMkách v /home/src/linux-4.18.0-348.12.2.el8_5.tar.xz, prípadne si l OpenSource :).

V prípade akýchkoľvek nejasností sa nám neváhajte ozvať.