

## Ixia Challenge - Kprobe based tracer

---

- Termen de predare pentru Ixia Challenge: **Miercuri, 27 Martie 2013, ora 23:00**
- Termen de predare pentru Sisteme de Operare 2: **Duminică, 31 Martie 2013, ora 23:00**
- Premiera Ixia Challenge are loc **Joi, 28 Martie 2013**, în cadrul Seminarului Ixia – EC101, 14-16
- Poate participa orice student din ciclul de licență din cadrul Facultății de Automatică și Calculatoare.
- Premiul pentru câștigătorul Ixia Challenge este participarea (cu acoperirea tuturor costurilor) la LinuxCon Europe 2013 [<http://events.linuxfoundation.org/events/linuxcon-europe>], 21-23 octombrie 2013, Edinburgh, UK.

## Obiectivele temei

---

- dobândirea de cunoștințe legate de instrumentarea funcțiilor din nucleul Linux (mecanismul `kprobes`)
- obținerea de cunoștințe legate de interacțiunea cu sistemul de fișiere `/proc` din nucleul Linux.
- obținerea de deprinderi de lucru cu structurile de date specifice nucleului Linux (`hash`, `list`).

## Enunț

---

Realizați un inspector de operații de kernel.

Prin acest inspector ne propunem să interceptăm:

- apelurile `kmalloc` și `kfree`
- apelul `schedule`
- apelurile `up` și `down_interruptible`
- apelurile `mutex_lock` și `mutex_unlock`

Inspectorul va reține, la nivel de proces, numărul de apeluri pentru fiecare dintre funcțiile de mai sus. În plus, pentru apelurile `kmalloc` și `kfree` se afișează cantitatea totală de memorie alocată și eliberată.

Inspectorul va fi implementat ca un modul de kernel cu numele `tracer.ko`.

## Detalii de implementare

---

Interceptarea se va face înregistrând câte o probă (`kprobe`) pentru fiecare din funcțiile de mai sus. Inspectorul va reține o listă/hash de procese monitorizate și va contabiliza informațiile de mai sus pentru aceste procese.

Pentru controlul listei/hash-ului de procese monitorizate, se va folosi un char device denumit `/dev/tracer`, cu majorul `10` și minorul `42`. Acesta va expune o interfață `ioctl` cu două argumente:

- primul argument reprezintă o cererea către subsistemul de monitorizare.
  - `TRACER_ADD_PROCESS`
  - `TRACER_REMOVE_PROCESS`
- al doilea argument reprezintă PID-ul procesului pentru care se va executa cererea de monitorizare.

Pentru a putea crea un char device cu majorul **10** va trebui să folosiți interfața `miscdevice` [<http://lxr.linux.no/#linux+v3.8.3/include/linux/miscdevice.h#L52>] din kernel.

Întrucât funcția `kmalloc` este inline pentru instrumentarea cantității de memorie alocate se va inspecta funcția `__kmalloc`, astfel:

- se va folosi o probă de tip `kretprobe`, care va reține cantitatea de memorie alocată și adresa zonei de memorie alocată.
- se vor folosi câmpurile `.entry_handler` și `.handler` din structura `kretprobe`

Întrucât funcția `kfree` primește doar adresa zonei de memorie ce urmează a fi eliberată, pentru a determina cantitatea totală de memorie eliberată va trebui pe baza adresei zonei să determinăm dimensiunea ei. Acest lucru este posibil întrucât există o asociere adresa - dimensiune făcută la inspectarea funcției `__kmalloc`.

Pentru toate celelalte funcții se va folosi o probă de tipul `jprobe`. Kernel-ul din mașina virtuală are opțiunea `CONFIG_DEBUG_LOCK_ALLOC` activată unde simbolul `mutex_lock` este un macro care se expandează la `mutex_lock_nested`. Astfel, pentru a obține informații despre funcția `mutex_lock` va trebui să instrumentați funcția `mutex_lock_nested`.

Procesele care au fost adăugate în listă/hash și care își încheie execuția vor fi eliminate din listă/hash. De asemenea un proces va fi eliminat din lista/hash în urma operației `TRACE_REMOVE_PROCESS`.

Informațiile reținute de inspector vor fi afișate prin intermediul sistemului de fișiere `procfs`, în fișierul `/proc/tracer`. Pentru fiecare proces monitorizat se creează o intrare în fișierul `/proc/tracer` având ca și prim câmp PID-ul procesului. Intrarea va fi de tipul read-only, și o operație de citire pe aceasta va afișa rezultatele reținute. Un exemplu de afișare a conținutului intrării este:

```

$cat /proc/tracer
PID  kmalloc kfree kmalloc_mem kfree_mem sched  up    down  lock  unlock
42   12     12    2048        2048     124   2     2     9     9
1099 0       0       0           0       1984  0     0     0     0
1244 0       0       0           0       1221  100   1023  1023  1002
1337 123     99     125952     101376  193821 992   81921 7421  6392

```

## Testare

- Corectarea temelor se va realiza automat cu ajutorul unei suite de teste publice.
- Indicații despre utilizarea suitei de teste se găsesc în fișierul **README** din cadrul arhivei.
- Pentru evaluare și corectare tema va fi uploadată folosind interfața `vmchecker` [<https://elf.cs.pub.ro/vmchecker/ui>].
- Suita de teste conține un set de teste. Trecerea unui test conduce la obținerea punctajului aferent acestuia.
  - În urma rulării testelor, se va acorda, în mod automat, un punctaj total. Punctajul total maxim este de 100 de puncte, pentru o temă care trece toate testele.
  - Cele 100 de puncte corespund la 10 puncte din cadrul notei finale.
- Pot exista penalizări în caz de întârzieri sau pentru neajunsuri de implementare sau de stil.
- Penalizările pentru neajunsuri de implementare pot constitui maxim 1 punct din nota finală a temei.
  - Pe lângă penalizările precizate în cadrul listei de depuneri [<http://elf.cs.pub.ro/so2/wiki/teme/general>], se vor avea în vedere următoarele elemente:
    - 0.2:** omiterea eliminării corespunzătoare a resurselor (kprobes, intrări în `/proc`)

- **-0.2:** probleme de sincronizare la date folosite de mai multe instanțe de execuție (spre exemplu, lista/tabela de hash cu procese)

## Evaluare pentru Ixia Challenge

- Evaluarea pentru Ixia Challenge, va ține cont de elemente de stil, aranjare a codului și folosirea unor metode eficiente de coding.
- Evaluarea pentru Ixia Challenge, impune folosirea unui hash pentru menținerea listei de procese monitorizate. Înscrierea în competiție a temei este condiționată de acest lucru.
- Participarea în cadrul Ixia Challenge presupune submiterea temei până pe data de 27 Martie 2013, ora 23:00 și trecerea tuturor testelor.
- Submiterea temei până la data de 27 Martie 2013, ora 23:00 înseamnă că tema valorează **2 puncte** din nota la Sisteme de Operare 2.
  - Predarea temei cu întârziere înseamnă că tema valorează **1 punct** din nota la Sisteme de Operare 2, minus depunctarea de întârziere.

## Resurse de suport

---

- `Documentation/kprobes.txt` - descrierea subsystemului kprobes din sursele nucleului Linux.
- `samples/kprobes/` - câteva exemple de folosire a kprobes din sursele nucleului Linux.

## Suport, întrebări și clarificări

---

- Pentru întrebări legate de temă puteți consulta arhivele [<http://news.gmane.org/gmane.education.region.romania.operating-systems-design>] listei de discuții sau puteți trimite un e-mail [<mailto:so2@cursuri.cs.pub.ro>] (trebuie să fiți înregistrați [<http://cursuri.cs.pub.ro/cgi-bin/mailman/listinfo/so2>]).
- Vă rugăm să urmăriți și să respectați indicațiile de utilizare a listei [<http://ocw.cs.pub.ro/courses/so2/resurse/lista-discutii#mailing-list-guidelines>].

so2/teme/challenge.txt · Last modified: 2013/03/21 21:45 by razvan.deaconescu