Линукс курс 2022/2023

Дебаговање руковалаца и кернела

ЦИЉ

Научити како се користи debugFs и како се проналазе грешке у кернел коду.

ИСХОД

Након ове вежбе ћете моћи да:

- Креирате debugFs уносе у оквиру свог модула
- Пронађете грешку у оквиру кернел кода
- Декомпајлирате кернел

ПОСТАВКА

Позиционирајте се у директоријум `~/linux-kernel-labs/src` који садржи коренски систем датотека за RPI и затим од полазне гране `master` направите нову грану `lab08` и позиционирајте се у исту:

```
git checkout master
git checkout -b lab08
```

Позиционирајте се у директоријум `~/linux-kernel-labs/src/linux` који садржи кернел за RPI са `github` репозиторијума https://github.com/raspberrypi/linux и затим направите нову грану `lab08` и позиционирајте се у исту:

```
git checkout -b lab08 8e1110a580887f4b82303b9354c25d7e2ff5860e
```

Линукс курс 2022/2023

PR_DEBUG( ) И ДИНАМИЧКО ДЕБАГОВАЊЕ

Додајте `pr_debug()` исписе у све функције `hello_version` модула. Такође, додајте и глобалне бројаче који прате укупан број уписаних бајта, укупан број прочитаних бајта и тренутно доступан број бајта за читање.

Проверите шта се дешава са модулом? Да ли су исписи који су додати видљиви? Уколико нису видљиви, укључите `CONFIG_DYNAMIC_DEBUG` и поново преведите кернел. Након превођења маунтујте `debugfs` на путању `/sys/kernel/debug` и конфигуришите исписе на следећи начин:

1. Излистајте све доступне дебаг поруке у кернелу,
2. Укључите све поруке из `hello_version` модула и проверите да ли су заиста видљиве,
3. Укључите само једну поруку из `hello_version` модула и уверите се да се види само та порука, а остале не.

Више детаља можете пронаћи у документацији кернела на путањи: `Documentation/admin-guide/dynamic-debug-howto.rst`.

Сада имате добар механизам да држите пуно дебаг исписа у модулу и селективно бирате које ћете да прикажете.

DEBUGFS

Користећи `debugfs` додајте нове уносе у дебаг систему датотека. У коду руковаоца додајте одговарајуће позиве функција из `debugfs` API тако да оне направе:

- директоријум `hello` у оквиру `debugfs` система датотека,
- у том директоријуму датотеке које ће да приказују вредности глобалних бројача.

Поново преведите и учитајте модул и проверите да ли су новокреирани уноси у оквиру debugfs-а видљиви на маунтованој путањи `/sys/kernel/debug/hello/`.

АНАЛИЗА ГРЕШКЕ У КЕРНЕЛУ

Позиционирајте се у директоријум

`linux-kernel-labs/modules/nfsroot/root/debugging/`.

Проверите да ли је кернел преведен са:

- укљученим `CONFIG_DEBUG_INFO`
- искљученим `CONFIG_UNWINDER_ARM`

Преведите модул `drvbroken`, покрените га на платформи и видите поруку о грешци.

АНАЛИЗА ПОРУКЕ О ГРЕШЦИ

Пажљиво анализирајте поруку о грешци. Знајући да на ARM-у PC регистар садржи локацију инструкције која се извршава, покушајте да откријете која функција је изазвала грешку у кернелу.

Коришћењем LXR-а кернел кода, погледати декларацију проблематичне функције. Ово би требало да буде довољно да се открије и отклони грешка.

ДУБЉА АНАЛИЗА ПРОБЛЕМА

У случају када се једноставним прегледом кода и поруке о грешци не може утврдити шта је проблем, може да се погледа дисасемблирана верзија функције на један од два начина:

1. cd linux-kernel-labs/src/linux

   arm-linux-gnueabihf-objdump -S vmlinux > vmilinux.disasm

2. коришћењем gdb-multiarch

   sudo apt-get install gdb-multiarch

   gdb-multiarch vmlinux

   (gdb) set arch arm

   (gdb) set gnutarget elf32-littlearm

   (gdb) disassemble function_name

Даље је могуће пронаћи тачну инструкцију на којој је дошло до грешке. Померај унутар функције се може видети у поруци о грешци, а такође се може видети и део кода око проблематичне инструкције.

Наравно, за овакву врсту дебаговања је потребно одређено знање асемблера за дату архитектуру.

САЧУВАЈТЕ СВЕ ИЗМЕНЕ

Да бисте потврдили и сачували све измене, најбоље је да их додате, а потом и локално комитујете на GIT, док сте позиционирани у неки од директоријума репозиторијума који је мењан, нпр. `~/linux-kernel-labs` и исто за `~/linux-kernel-labs/src/linux`:

```
git add -A

git commit -as -m "lab08 done"
```

Да би измене постале видљиве и у репозиторијуму на серверу, потребно би још било урадити нпр. `git push`, али то у овом случају није неопходно нити имамо неопходна права за то.