

Линукс курс 2020/2021

Писање и превођење модула

ЦИЉ

Научити како се пишу и преводе модули.

ИСХОД

Након ове вежбе ћете моћи да:

- Преузмете изворни код са званичне локације
- Напишете кернел модул са неколико могућности, укључујући параметре
- Приступите кернелу из вашег модула
- Поставите окружење за превођење модула
- Додате код вашег модула у кернел стабла и да изградите кернел закрпу из вашег новог изворног кода
- Креирате једноставан знаковни руковалац
- Добијете од кернела слободан `major` број и направите одговарајућу датотеку уређаја
- Коришћењем `kmalloc` и `kfree` функција управљате меморијом.

ПОСТАВКА

Позиционирајте се у директоријум `~/linux-kernel-labs/modules` који садржи коренски систем датотека за RPI и затим од полазне гране `master` направите нову грану `lab2` и позиционирајте се у исту:

```
git checkout master
```

Линукс курс 2020/2021

```
git checkout -b lab2
```

Позиционирајте се у директоријум `~/linux-kernel-labs/src`.

Са `github` репозиторијума преузмите код помоћу одговарајуће `git` команде:

```
git clone -b rpi-4.4.y --single-branch https://github.com/raspberrypi/linux --depth=1
```

Позиционирајте се у директоријум `~/linux-kernel-labs/src/linux` и затим направите нову грану `lab2` и позиционирајте се у исту:

```
git checkout -b lab2
```

За каснији рад (комитовање измена) на GIT-у, конфигуришите своје податке:

```
git config --global user.email "ime.prezime@mail"
```

```
git config --global user.name "Ime Prezime"
```

УПОЗНАЈТЕ СЕ СА КОДОМ

Као корисник кернела Линукса, често ћете морати да нађете која датотека имплементира дату функцију. Стога, корисно је упознати се са изворним кодом кернела прегледањем датотека.

1. Пронађите слику Линукс логоа у изворном коду.
2. Пронађите ко одржава 3C505 мрежни драјвер.
3. Пронађите декларацију `platform_device_register()` функције.

Поред ручне претраге кода постоје и алати који нам у томе помажу. Испробајте претрагу помоћу LXR (Linux Cross Reference) на адреси <http://lxr.free-electrons.com> и изаберите верзију Линукса најближу оној коју користите.

Такође, користан алат у ситуацијама када приступ интернету није омогућен је `cscope`. Коришћењем `cscope` алата и LXR пронађите претходно ручно пронађене датотеке и информације и уочите предност коришћења алата за индексирање. Да би се извршило индексирање целог Линукс кода са `cscope`

Линукс курс 2020/2021

алатом потребно је покренути команду `cscope -Rk`. Додатне информације о коришћењу овог алата пронаћи покретањем команде `cscope --help`. Уколико је потребно претходно инсталирајте `cscope` са командом `sudo apt-get install cscope`.

ИНСТАЛАЦИЈА АЛАТА ЗА ПРЕВОЂЕЊЕ

Пре него што пређемо на конфигурисање и превођење Линукс језгра потребно је инсталирати скуп алата за превођење. За превођење Линукса за RPI плочу користимо `gcc-arm-linux-gnueabi` преводилац. Инсталирајте поменути преводилац са командом:

```
sudo apt-get install gcc-arm-linux-gnueabi
```

КОНФИГУРАЦИЈА КЕРНЕЛА

Пошто кернел преводимо за RPI плочу која подржава ARM скуп инструкција, а само превођење се обавља на x86 архитектури, најпре је потребно подесити одговарајуће окружење за превођење. Минимално што је потребно да се уради је да се подесе циљна архитектура и префикс алата за превођење који ће се користити. Ово се може урадити на више начина од којих сваки има своје предности и мане.

Први начин је директна промена вредности `ARCH` и `CROSS_COMPILE` варијабли у `Makefile` датотеци. Овај приступ је погодан уколико кернел увек преводимо за исту архитектуру и са истим преводиоцем. Ипак, како то није чест случај, овај начин подешавања окружења за превођење се не препоручује.

Други начин, који је и најчешћи је подешавање варијабли окружења `ARCH` и `CROSS_COMPILE` помоћу команде `export`. Уколико желимо да подесимо окружење за превођење за RPI плочу на овај начин потребно је да извршимо следеће команде:

```
export ARCH=arm
```

```
export CROSS_COMPILE=arm-linux-gnueabi-
```



Линукс курс 2020/2021

Мана овог приступа је у томе што су подешене варијабле видљиве само у терминалу у ком су извршене претходно наведене команде.

Трећи начин подешавања окружења за превођење је да се вредности варијабли `ARCH` и `CROSS_COMPILE` проследе при сваком позиву `make` команде, на пример:

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- -j2
```

При коришћењу овог приступа постоји велика шанса да се у неком од позива `make` команде изоставе потребне варијабле што резултује грешком у превођењу.

Након што сте подесили окружење за превођење за RPI плочу можете да приступите конфигурисању кернела.

Позиционирајте се у преузети кернел директоријум. Примените подразумевану конфигурацију за `bcm2709` (`bcm2709_defconfig`).

```
make bcm2709_defconfig
```

Покрените команду `xconfig` да покренете спрегу за конфигурацију кернела. Спрега за конфигурацију кернела је дата као изворни код у самом кернелу и команда `make xconfig` је преводи аутоматски, али су неопходна заглавља и библиотеке. Можда ће вам бити потребно да инсталирате додатне пакете, попут `libqt4-dev` који садржи Qt развојне датотеке, као и `g++` пакет који представља C++ преводилац.

У покренутој спрези за конфигурацију, укључите `Options --- Show Name` опцију. Ово је понекад корисно, када је име параметра експлицитније од описа или када пратимо нека упутства која су нам задата са именима параметара.

Такође, пробајте и опције `Options --- Show All Options` и `Options -- Show Debug Info`. Ове опције ће вам дозволити да видите све параметре који се иначе не би приказали, јер зависе од вредности других параметара. Тако што кликнете на опис неког од тих параметара, видећете његове предуслове за укључивање и разумећете зашто није могуће да се одаберу.

Специфицирајте суфикс верзију (опција `LOCALVERSION`); тако ћете моћи да разликујете свој кернел командом `uname -r` или `cat /proc/version`, кад покренете систем.



Линукс курс 2020/2021

Такође, можете пробати `make menuconfig`. Иако није графичка спрега, неки људи преферирају овај начин конфигурације. Пошто је `menuconfig` спрега базирана на `Ncurses` `biblioteci`, мораћете да инсталирате `libncurses-dev` пакет да бисте је користили.

ПРЕВЕДИТЕ И ПОКРЕНИТЕ КЕРНЕЛ

Потребно је само да покренете:

```
make
```

Добијени кернел је потребно покренути на RPI плочи. Преведен кернел покренути из `U-boot-a`. За `rfs` користити директоријум `nfsroot` преко мреже.

ПИСАЊЕ И ПРЕВОЂЕЊЕ МОДУЛА

Позиционирати се у `nfsroot/root/hello` директоријум. Попунити `hello_version.c` датотеку са одговарајућим кодом тако да се при учитавању модула прикаже порука:

```
Hello Master. You are currently using Linux <version>.
```

Такође, потребно је приказати и поздравну поруку приликом уклањања модула.

У истом директоријуму се налази и `Makefile` датотека која омогућава превођење модула ван структуре кернел стабла. Упознајте се са садржајем ове датотеке, по потреби прилагодите путање и преведите модул.

Напомена: Обавезно је поставити окружење за унакрсно превођење и приликом превођења модула.

Линукс курс 2020/2021

ТЕСТИРАЊЕ МОДУЛА – САВЕТИ

Учитати нови модул. Проверити да ли ради као што је очекивано. Док не проради, уклонити га, модификовати код, превести и поново покренути колико год је пута потребно.

Покренути команду за проверу учитаних модула. Након тога, покушати доћи до исте информације само коришћењем `cat` команде.

ДОДАВАЊЕ ПАРАМЕТАРА У МОДУЛ

Додати “who” параметар у модул. Модул би требало да испише “Hello <who>...” уместо “Hello Master...”.

Превести и тестирати модул проверавајући да ли прихвата `who` параметар при читавању.

ДОДАВАЊЕ ИНФОРМАЦИЈЕ О ВРЕМЕНУ

Изменити поздравну поруку модула тако да прикаже и колико је секунди протекло од читавања модула.

За добављање тренутног времена се може користити функција `do_gettimeofday()`.

ЈЕДНОСТАВАН ЗНАКОВНИ РУКОВАЛАЦ

Претворити модул у једноставан знаковни рукавалац. Препустити кернелу доделу `major` броја, а за `minor` број користити 0. Попунити `read`, `write` и `ioctl` операције модула на следећи начин:

read

Попунити бафер из корисничког простора садржајем бафера динамички заузетог приликом иницијализације модула.

Линукс курс 2020/2021

write

Потребно је прослеђени стринг из корисничког простора сместити у унапред динамички заузет бафер.

ioctl

У зависности од прослеђене команде урадити следеће:

- 0 – пребацити у мала слова цео садржај бафера динамички заузетог приликом иницијализације модула
- 1 - пребацити у велика слова цео садржај бафера динамички заузетог приликом иницијализације модула

Омогућити приступ модулу из корисничког простора креирањем одговарајуће датотеке уређаја уколико је потребно тестирати и додате функције. За тестирање `write` операције користити команду `echo`, за `read` операцију користити команду `cat`, а за `ioctl` операцију превести `ioctl.c` са истим преводиоцем и обавезно **статичким увезивањем** (опција `-static`) и користити добијени програм као тестни.

ЛИНУКС СТАНДАРД КОДОВАЊА

Приликом кодовања би требало да се придржавате стриктних стандарда кодовања, уколико желите да једног дана и тај код буде укључен у изворни код Линукса. Један од главних разлога за то је читљивост кода. Ако би свако користио свој стил, узимајући у обзир велики број контрибутора, читање кода кернела би било веома непријатно.

Срећом, Линукс кернел заједница нуди алат за проверу доследности у придржавању стандарду кодовања.

Покрените `scripts/checkpatch.pl -h` команду у изворном коду кернела да бисте проверили доступне опције. Након тога покрените:

```
~/linux-kernel-labs/src/linux/scripts/checkpatch.pl --file  
--no-tree hello_version.c
```

Линукс курс 2020/2021

Проверите колико недоследности је пријављено за ваш код и унесите исправке док год има недоследности. Уколико је пријављено много грешака, проверите да ли вам је на коректан начин конфигурисан едитор, према правилима стила кодовања кернела из Documentation/CodingStyle.

ДОДАВАЊЕ HELLO_VERSION КОДА У КЕРНЕЛ КОД

Додати изворни код модула у `drivers/misc/` директоријум у кернел коду. Модификовати одговарајуће конфигурационе датотеке и `Makefile` како би модул постао видљив у конфигурационим алатима (`xconfig`, `menuconfig...`) и како би могао да се преведе заједно са кернелом.

Када модул постане видљив у конфигурационим алатима изабрати да се преводи као модул. Покренути превођење кернела. По завршетку превођења кернела, модуле инсталирати у `nfsroot` командама:

```
sudo chown -R rtrk.rtrk <putanja do nfsroot
direktorijuma>/lib/
```

```
make modules_install INSTALL_MOD_PATH=<putanja do nfsroot
direktorijuma>
```

```
sudo chown -R root.root <putanja do nfsroot
direktorijuma>/lib/
```

КРЕИРАЊЕ ЗАКРПЕ

Креирати закрпу која ће додавати изворни код новог модула у чист кернел код, али и извршити потребне промене у конфигурационим датотекама и `Makefile`-овима.

Тестирати закрпу применом на чист кернел код.



Линукс курс 2020/2021



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 "modul zavrsen"
```

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.