

Линукс кернел

Шта кернел ради?

Вероватно сте чули да људи причају о компиловању кернела или изградњи кернела, али шта је заправо кернел и шта он ради? Кернел је средиште вашег рачунара. То је основа целог оперативног система. Кернел представља мост између хардвера и апликација. Ово значи да је кернел (обично) једини део софтвера одговоран за руковођење хардверским компонентама вашег рачунара. Кернел налаже јединици чврстог диска претрагу за одређеним током података. Кернел налаже вашем мрежном уређају да преноси брзе промене напона. Кернел такође ослушкује хардвер. Када мрежни уређај детектује информације које шаље удаљени рачунар, он те информације проследи кернелу. Ово кернел чини најважнијим појединачним делом софтвера на вашем рачунару као и најкомплекснијим.

Рад са модулима

Комплексност модерног линукс језгра је запањујућа. Некомпримовани изворни код језгра тежи приближно 400 МБ. Постоје хиљаде развијача, стотине опција, и када све буде изграђено заједно, кернел би могао сâм да пређе величину од 100 МБ. Како би се величина језгра смањила (као и количина меморије са директним приступом (енгл. RAM) потребне за кернел), већина опција су изграђене као модули. Ове модуле можете посматрати као драјвере за уређај који могу бити укључени или уклоњени из језгра по жељи. Истина, многи од њих уопште не представљају драјвере за уређај, али садрже подршку за ствари као што су мрежни протоколи, сигурносне мере, па чак и системе датотека. Укратко, скоро сваки део линукс језгра може бити изграђен као учитавајући модул.

Веома је битно упамтити да Слеквер аутоматски регулише учитавање већине модула. Када се систем покрене, **udev**(8) се стартује и почиње испитивање вашег системског хардвера. За сваки пронађени уређај, учитава се одговарајући модул и креира се чвориште уређаја у `/dev`. Ово значи да нећете морати да учитавате било који модул како бисте користили ваш рачунар, али повремено ће то бити непходно учинити.

Дакле који модули су тренутно учитани и како их учитати или уклонити? Срећом, имамо читав комплет алата за ово. Као што можете предпоставити, алат за листање модула је **lsmod**(8).

```
darkstar:~# lsmod
Module                Size  Used by
nls_utf8              1952   1
cifs                  240600 2
i915                  168584 2
drm                   168128 3 i915
i2c_algo_bit          6468   1 i915
tun                   12740   1
... многе линије су изостављене ...
```

Поред тога што вам показује који су модули учитани, приказује и величину сваког модула као и

који га други модули користе.

Постоје две апликације за учитавање модула: **insmod(8)** и **modprobe(8)**. Обе ће учитати модуле и пријавити грешке (као што је учитавање модула за уређај који није присутан у вашем систему), али **modprobe** је пожељнији јер може да прочита зависности модула. Коришћење обе апликације је исправно.

```
darkstar:~# insmod ext3
darkstar:~# modprobe ext4
darkstar:~# lsmod | grep ext
ext4                239928  1
jbd2                 59088  1 ext4
crc16                1984  1 ext4
ext3                 139408  0
jbd                  48520  1 ext3
mbcache              8068  2 ext4,ext3
```

Уклањање модула може да буде тежак процес, и опет имамо два програма за уклањање: **rmmmod(8)** и **modprobe**. Како бисте уклонили модул са **modprobe**, морате да користите `-r` аргумент.

```
darkstar:~# rmmmod ext3
darkstar:~# modprobe -r ext4
darkstar:~# lsmod | grep ext
```

Компиловање кернела и зашто бисте то урадили

Многи корисници Слеквера неће никад морати да компилују кернел. Огромна и генеричка језгра садрже сву подршку која вам требати.

Међутим, неки корисници ће морати да компилују кернел. Уколико ваш рачунар има најновији хардвер, нова верзија кернела може да понуди бољу подршку. Понекад ће бити доступна закрпа за кернел која ће исправити проблем који сте искусили. У овим случајевима је компиловање кернела оправдано. Корисници који желе последњу верзију кернела или верују да ће им коришћење прилагођеног кернела дати боље перформансе, могу надоградити језгро, али највероватније неће уочити веће промене.

Уколико сматрате да је компиловање сопственог кернела нешто што желите или вам је потребно, овај одељак ће вам бити водич. Компиловање кернела није компликован процес, али могуће је начинити одређене грешке које могу да спрече покретање вашег рачунара и доведу до већих фрустрација.

Први корак треба да буде провера да ли имате инсталиран изворни кôд језгра на ваш систем. Тражени пакет се налази у „k“ директоријуму на инсталационом диску, а другу верзију можете преузети са <http://www.kernel.org/>. Традиционално, извор кернела се налази у `/usr/src/linux` и представља симболичку везу до одређеног издања кернела који користите, али ово није искључиво правило. Изворни кôд језгра можете да поставите било где без проблема.

```
darkstar:~# ls -l /usr/src
```

```
lrwxrwxrwx 1 root root 14 2009-07-22 19:59 linux -> linux-2.6.29.6/  
drwxr-xr-x 23 root root 4096 2010-03-17 19:00 linux-2.6.29.6/
```

Најтежи део компиловања било ког кернела је конфигурација кернела. Постоји стотину опцију, а многе од њих опционо могу бити компиловане у модуле. То значи да постоји хиљаде начина конфигурисања кернела. Срећом, постоји одређен број трикова који вам могу помоћи да се не увалите у невоље. Конфигурациони фајл кернела је `.config`. Уколико сте храбри, можете ручно уређивати овај фајл у омиљеном уређивачу текста, али је веома препоручљиво користити уграђене језгарне алате за манипулацију овим фајлом.

Осим ако сте упознати са конфигурацијом кернела, увек би требало да кренете од стабилне основе у правцу модификовања. Ово ће вас спречити у прескакању битних опција што може довести до компиловања кернела изнова и изнова, све док не урадите тачно шта треба. Најпоузданији `.config` фајлови кернела за почетак су они коришћени у Слекверовим подразумеваним језгрима. Њих можете наћи на Слекверовом инсталационом диску или на вашем омиљеном огледалу, у `/kernels` директоријуму.

```
darkstar:~# mount /mnt/cdrom  
darkstar:~# cd /mnt/cdrom/kernels  
darkstar:/mnt/cdrom/kernels# ls  
VERSIONS.TXT huge.s/ generic.s/ speakup.s/  
darkstar:/mnt/cdrom/kernels# ls genric.s  
System.map.gz bzImage config
```

Подразумевани `.config` фајл можете једноставно заменити умножавањем или преузимањем `.config` фајла за кернел који желите да користите како основу. Овде користим Слекверов **generic.s** кернел за основу који је препоручљив, али ћете ви можда желети да користите **huge.s** конфигурациони фајл. Генерички кернел гради већи број ствари као модуле па стога креира мању слику језгра, али обично захтева коришћење `initrd`-а.

```
darkstar:/mnt/cdrom/kernels# cp generic.s/config /usr/src/linux/.config
```



Фајл Слекверовог језгра не садржи тачку у називу док је фајл језгра садржи. Уколико заборавите или једноставно прекопирате `config` у `/usr/src`, уместо прекопираног биће коришћен било који `.config` фајл који већ постоји.

Уколико желите да користите конфигурациони фајл тренутног језгра као основу, највероватније ћете га наћи у `/proc/config.gz`. Ово је посебан фајл у вези кернела који садржи целокупну конфигурацију језгра у компримованом формату и захтева да је кернел изграђен да га подржи.

```
darkstar:~# zcat /proc/config.gz > /usr/src/linux/.config
```

Сада кад смо креирали стабилну основу конфигурације, време је да начинимо измене које желимо. Целокупан процес изградње кернела од конфигурације до компиловања се врши командом **make**(1) и додавањем посебних аргумената. Сваки аргумент врши различиту функцију.

Уколико вршите надоградњу на новије издање кернела, дефинитивно ћете желети да

користите *oldconfig* аргумент. Ово ће кренути од основног *.config*-а и претражиће недостајуће елементе јер обично новија издања кернела садрже додатне опције. Обзиром да се опције додају у сваком новом издању кернела, ово је најбољи начин.

```
darkstar:/usr/src/linux# make oldconfig
scripts/kconfig/conf -o arch/x86/Kconfig
*
* Restart config...
*
*
* File systems
*
Second extended fs support (EXT2_FS) [M/n/y/?] m
  Ext2 extended attributes (EXT2_FS_XATTR) [N/y/?] n
  Ext2 execute in place support (EXT2_FS_XIP) [N/y/?] n
Ext3 journalling file system support (EXT3_FS) [M/n/y/?] m
  Ext3 extended attributes (EXT3_FS_XATTR) [Y/n/?] y
  Ext3 POSIX Access Control Lists (EXT3_FS_POSIX_ACL) [Y/n/?] y
  Ext3 Security Labels (EXT3_FS_SECURITY) [Y/n/?] y
The Extended 4 (ext4) filesystem (EXT4_FS) [N/m/y/?] (NEW) m
```

Овде можете видети да је нови кернел који компилујем додао подршку за нови систем датотека: *ext4*. *oldconfig* је прошао кроз изворну конфигурацију, задржао све старе опције тачно онако како су постављене, и сада ме пита шта ради са новим опцијама. Најбезбедније је одабрати подразумевану опцију, али је ви свакако можете изменити. *oldconfig* је веома zgodna алатка за упознавање са новим конфигурационим опцијама, што је идеално за кориснике који само желе да пробају ново издање кернела.

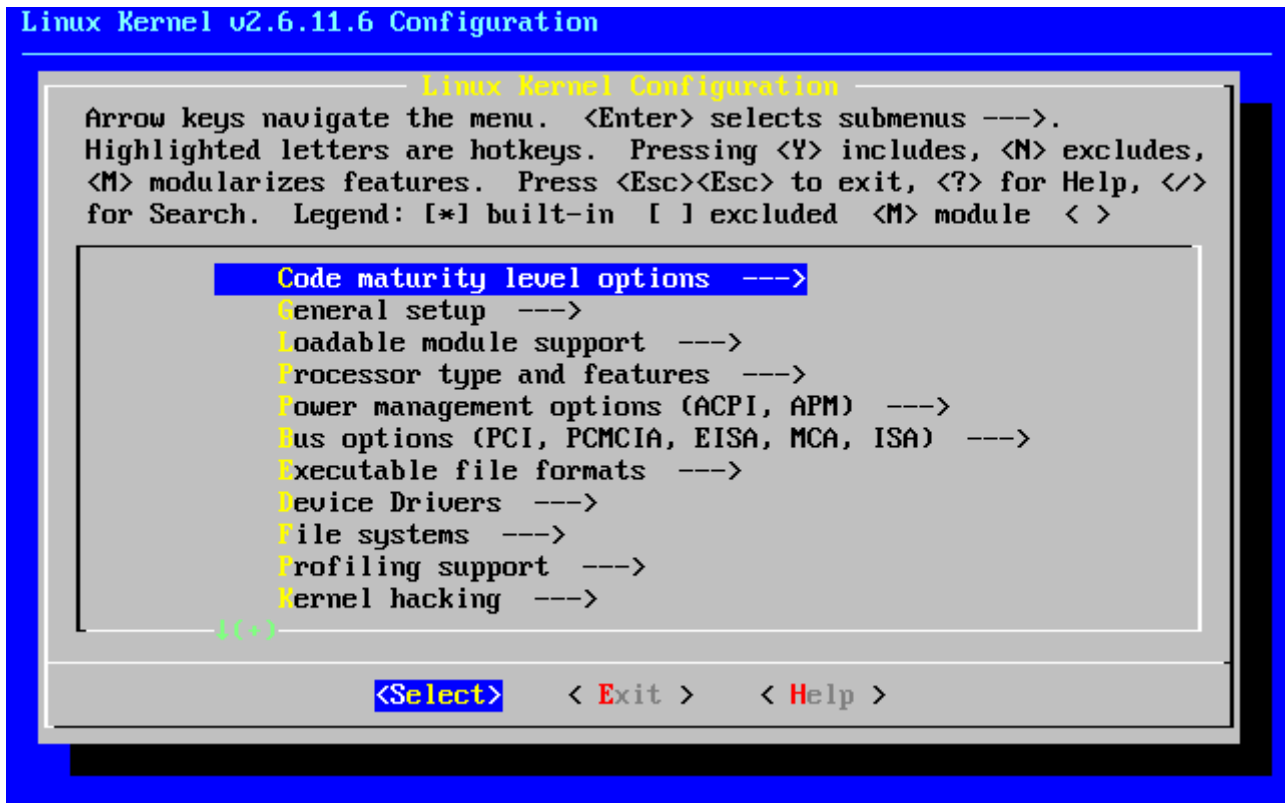
За озбиљније задатке, када је конфигурација у питању, ту је мноштво опција. Линукс језгро може бити конфигурисано на три основна начина. Први је *config*, који ће вас водити кроз сваку опцију корак по корак и питаће шта желите да урадите. Ово је тако досадно да овај начин ретко ко користи данас.

```
darkstar:/usr/src/linux# make config
scripts/kconfig/conf arch/x86/Kconfig
*
* Linux Kernel Configuration
*
*
* General setup
*
Prompt for development and/or incomplete code/drivers (EXPERIMENTAL) [Y/n/?]
Y
Local version - append to kernel release (LOCALVERSION) [] -test
Automatically append version information to the version string
(LOCALVERSION_AUTO) [N/y/?] n
Support for paging of anonymous memory (swap) (SWAP) [Y/n/?]
```

Срећом, постоје два лакша начина за конфигурацију кернела: *menuconfig* и *xconfig*. Оба поменута креирају програм којим управљате из изборника и који вам омогућује да једноставно обележите или уклоните опцију, без да прелазите кроз сваку корак по корак. *menuconfig* је

најкоришћенији метод и њега препоручујем. *xconfig* је користан једино ако желите да компилујете кернел из графичког сучеља у **X**-у. Оба су веома слична, тако да ћемо документовати само *menuconfig*.

Покретање **make menuconfig** у терминалу представиће вам познато *curses* сучеље као на слици испод. Сваки део кернела има свој подмени кроз који се можете кретати тастерима са стрелицама.



Уколико компилујете кернел који предтсавља исто издање као и ускладиштен Слекверов кернел, морате да подесите *Local version* опцију. Она се налази у *General setup* подменију. Изостављање ове опције ће резултовати преписивањем свих модула коришћених од стране ускладиштених кернела, што може довести до онемогућавања покретања система.

Када завршите са конфигурацијом кернела, време ја за компиловање. Постоји више метода за ово, али је најпоузданији коришћење *bzImage*-а. Када доделите овај аргумент команди **make**, компиловање кернела ће почети и видећете мноштво података како се исписују у терминалу све док се процес компиловања не заврши или се не појави нека грешка.

```
darkstar:/usr/src/linux# make bzImage
scripts/kconfig/conf -s arch/x86/Kconfig
CHK      include/linux/version.h
CHK      include/linux/utsrelease.h
SYMLINK  include/asm -> include/asm-x86
CALL     scripts/checksyscalls.sh
CC       scripts/mod/empty.o
HOSTCC   scripts/mod/mk_elfconfig
```

```
MKELF    scripts/mod/elfconfig.h
HOSTCC   scripts/mod/file2alias.o
... многе линије су изостављене ...
```

Уколико се процес оконча грешком, прво треба да проверите конфигурацију крнела. Грешке при компиловању обично настају због недостатака у `.config` фајлу. Предпоставимо да је све прошло како треба, још нисмо потпуно готови јер треба да изградимо модуле.

```
darkstar:/usr/src/linux# make modules
CHK      include/linux/version.h
CHK      include/linux/utsrelease.h
SYMLINK  include/asm -> include/asm-x86
CALL     scripts/checksyscalls.sh
HOSTCC   scripts/mod/file2alias.o
... многе линије су изостављене ...
```

Ако се компиловање и крнела и модула успешно завршило, ред је да их инсталирамо. Слика крнела треба да се пребаци на безбедну локацију, обично је то **/boot** директоријум, и треба да јој дате јединстваено име како бисте избегли преписивање других слика крнела које се ту налазе. Традиционално се слике крнела именују са `vmlinuz`, на шта се додаје издање крнела и локална верзија.

```
darkstar:/usr/src/linux# cat arch/x86/boot/bzImage > /boot/vmlinuz-
release_number-local_version
darkstar:/usr/src/linux# make modules_install
```

Када се све ово оконча, имаћете нову слику крнела смештену у `/boot` и нови директоријум језгарних модула смештен у `/lib/modules`. Како бисте користили ново језгро, мораћете да уређујете `lilo.conf`, креирате `initrd` (само уколико желите да учитате један или више ових језгарних модула при стартовању система), и да покренете **lilo** како бисте ажурирали програм за подизање оперативног система. Уколико након рестарта све крене по плану, требало би да имате опцију да стартујете новокомпиловани крнел. Уколико нешто крене по злу, мораћете да потрошите још мало времена да исправите проблем.

Навигација кроз поглавља

Претходно поглавље: [Праћење ажурирања](#)

Извори

- Оригинални извор: <http://www.slackbook.org/beta>
- Изворно писали Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
https://docs.slackware.com/sr:slackbook:linux_kernel

Last update: **2013/07/02 11:38 (UTC)**

