

Dinamikus rendszermagmodul-támogató keretrendszer

Kezeld a modulokat egy egyszerű háttérrendszerrel a rendszermagtól elkülönülten, és hozd ki többet a csomagkezelődből!

A forráskód önmagában is csodás dolog; a rendszermagfába foglalt modulforrás még csodásabb. Ennek a forrásnak a támogatása és támogatottsága az egyik legfontosabb tényező. Napjainkban a Linux robbanásszerűen terjed a vállalatoknál, és az a lehetőség, hogy egyetlen telefonhívással segítséget tudjunk kérni, nélkülözhetetlen. A Linux fejlesztését meghatározó kereteket soha nem látott mértékben formálják át a vállalatok elvárásai. A közösség részéről sokszor kételkedő és aggódó hangok hallatszanak emiatt, ám ha mindenki jól végzi a feladatát, a fejlesztések előnyeit bárki élvezheti. A dinamikus rendszermagmodul-támogatási (DKMS) keretrendszer nagyszerű példa minderre. A DKMS segítségével – a rendszert a Dell Computer Corporation számára készítették, a céget a javításoknak az ügyfelekhez ellenőrzött módon való eljuttatásában segíti – az illesztőprogramok fejlesztése, a tesztek és az ellenőrzések elvégzése is felgyorsítható, és ez a lehetőség a teljes közösség számára adott.

A DKMS keretrendszer alapjában véve egy kettős, a rendszermagfától különálló fa, amely a modulforrásokat és a lefordított, futtatható modulokat tartalmazza. A kettősség lehetővé teszi a moduloknak a rendszermagtól való szétválasztását, amely hasznos lehetőség a linuxos megoldásokat szállítók és telepítők számára. A rendszer nagyszerűsége abban rejlik, hogy – szabályozott és támogatható módon – lehetővé teszi az illesztőprogramok beillesztését a meglévő rendszermagokba. Így a szolgáltatók és ügyfelek az adott rendszermag változatától függetlenül átadhatják-átvehetik a javításokat. Ha egy illesztőprogramhoz megjelenik egy javítás, a DKMS áthidaló megoldásként lehetővé teszi annak terjesztését, amíg bele nem kerül a rendszermagba.

Ha tovább vizsgálódunk, a DKMS a felhasználók számára további lehetőségeket is kínál. Forrásból fordítani, telepíteni, forrás-RPM-ekkel küzdeni: ez az, amire a legtöbben a legkevésbé sem vágyunk. A valóságban számos olyan Linux-felhasználó van, aki kevés tapasztalattal rendelkezik, és egyszerűbb megoldásokat szeretne. A DKMS segít megoldani ezeket kérdéseket, mivel egyetlen futtatható állománnyal oldja meg a modulok fordítását, telepítését és eltávolítását. A modulok új rendszermagokon való üzembe helyezését is megkönnyíti, hiszen a telepítendő modulok beállításai kizárólag egy korábban már működött rendszermagon is alapulhatnak. Termelési környezetben ez nagy könnyebbséget jelent a rendszergazdák számára, hiszen ezentúl nem kell választaniuk az előre megadott programkészletek, valamint az újabb rendszermagok biztonsági szolgáltatásai között.

A DKMS a fejlesztők és a tapasztalt Linux-rajongók számára is tartogat meglepetéseket. A modulok és a rendszermagkettőzés által való (nem teljes) szétválasztásának már említett lehetősége nagyszerű környezet létrehozását teszi lehetővé illesztőprogramok fejlesztéséhez. Nem kell különféle rendszermagokhoz újra és újra javításokat készíteni, ehelyett a javításokat széles körben lehet terjeszteni és éles helyzetben ki lehet próbálni. A próbák felgyorsításával a teljes fejlesztési folyamat is

lerövidül. A különféle rendszermagkiadások léte a továbbiakban nem jelenthet akadályt a modulok széles körű terjesztése előtt, így alaposabban kipróbált kód jöhet létre, amit hamarabb be lehet építeni a rendszermagba – máris mindenki jól járt, fejlesztők és felhasználók egyaránt.

A DKMS használatával – a rendszermagfüggő programok előállításának egyszerűsödésével – könnyebbé válik a fejlesztők élete. Korábban például a Dell elsősorban rendszermagtól függő, előre lefordított RPM-ek formájában terjesztette a moduljait. Ahogy újabb és újabb rendszermagok jelentek meg, gyakran egyhangú, végtelennek tűnő munkába merültünk, amikor ezekhez újra kellett fordítani a futtatható fájlokat – ezt valószínűleg egyetlen fejlesztő sem kívánja magának. A Dell mégis ragaszkodott ehhez az eljárásához, mert ez kívánta meg a legkevésbé munkát és hozzáértést azon felhasználóktól, akiknek telepíteniük kellett a modulokat. A DKMS egyszerre könnyen használható és segít csökkenteni a fejlesztői oldalra háruló munka mennyiségét. A DKMS használatakor mindössze a modul forráskódjának kell a felhasználó gépén rendelkezésre állnia. Bármilyen rendszermagot futtat is a felhasználó, a DKMS elvégzi a modul fordítását és telepítését, mindenkit megkímélve a magváltozatok követése kapcsán előforduló bosszúságoktól.

A DKMS használata

Ha sikerült felkeltenem az érdeklődést a DKMS iránt – márpedig ez volt a szándékom –, próbáljunk meg egy kicsit elmélyülni a használatában. Ha rá szeretnénk venni a DKMS-t valamilyen modul kezelésére, akkor a modul forrásának rendelkezésre kell állnia a helyi gép `/usr/src/(modul)-(modul-változat)/` könyvtárában. Szükség van egy `dkms.conf` állományra, ebben megfelelően formázott utasításokkal kell előírni a DKMS-nek többek közt azt, hogy hová telepítse a modult, illetve hogyan fordítsa le. A `dkms.conf` állomány formátumáról később még szólni fogok. Ha ez a két követelmény teljesül, és a DKMS telepítve van a rendszerre, meg is kezdetjük a használatát, ha a kívánt modul/modulváltozat ággal kibővítjük a DKMS-fát. Lássunk egy példaparancsot a hozzáadás elvégzésére:

```
dkms add -m qla2x00 -v v6.04.00
```

A parancs lefutásakor a `qla2x00/v6.04.00` könyvtárral bővül a meglévő `/var/dkms` fa. A parancs kiadása után létrejön a `/var/dkms/qla2x00/v6.04.00/` könyvtár, egy közvetett hivatkozás a `/var/dkms/qla2x00/v6.04.00/source` könyvtárról a `/usr/src/qla2x00-v6.04.00/` könyvtárra, valamint az eredeti `dkms.conf` állományról készül egy másolat `/var/dkms/qla2x00/v6.04.00/dkms.conf` név alatt.

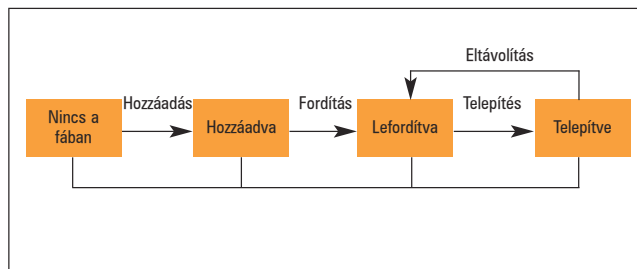
A hozzáadás után készen állunk a modul lefordítására.

A `dkms build` parancsának működéséhez a megfelelő rendszermagforrásnak a `/lib/module/kernel-version/build` közvetett hivatkozással elérhetőnek kell lennie. A modul lefordításához

használt `make` parancsot a `dkms.conf` beállítófájl határozza meg. Az előbbi példát folytatva:

```
dkms build -m qla2x00 -v v6.04.00
↳ -k 2.4.20-8smp
```

A paracssal lefordítjuk a modult, de nem telepítjük. A `build` parancs egy `kernel-version` átadott értéket is vár, ha ezt elhagyjuk, akkor az éppen futó rendszermag változatszámát veszi figyelembe. Mint sejtethető, az éppen futótól eltérő rendszermaghoz is könnyedén készíthetünk modulokat. Ezt a lehetőséget egy rendszermag-előkészítő programrész biztosítja, amely a modulok létrehozása előtt jut szerephez. A minden részletre kiterjedő előkészítés során lefut egy `make mrproper` parancs, a megfelelő rendszermag `.config` fájlja a rendszermag forrásának könyvtárba másolódik, lefut a `make oldconfig`, és végül a `make dep`. Ezek a lépések garantálják, hogy a létrejövő modul a megfelelő rendszermagszimbólumokkal épüljön fel. Az alapértelmezett beállítás szerint a DKMS a `/lib/modules/kernel-version/build/configs/` könyvtárban keresi a rendszermag `.config` fájlját, és a beállításfájlok kezelésekor a Red Hat névadási szokásait követi. Ha a rendszermag `.config` fájlja nem ebben a könyvtárban található, akkor a helyét a `build` parancs kiadásakor a `--config` kapcsolóval kell megadni a DKMS-nek.



Modulok állapota DKMS alatt

A fordítás végrehajtása során – eddigi példánknál maradván – létrejön a `/var/dkms/qla2x00/v6.04.00/2.4.20-8smp/` könyvtár, illetve ezen belül egy `log` és egy `module` alkönyvtár. A `log` könyvtárba a modul fordításának naplója, a `module` könyvtárba pedig a lefordított `.o` állományok kerülnek.

A fordítás befejezése után a modult azonnal telepíteni lehet ahhoz a rendszermaghoz, amelyhez készült. A telepítés során a rendszer a `/lib/modules/` fa megfelelő helyére másolja a modul lefordított, futtatható állományát, ahogy azt a `dkms.conf` fájlban elírtuk. Ha ezen a helyen már létezik ilyen nevű modul, azt a DKMS menti a saját fájába mint eredeti modult, így később, az újabb modul eltávolításakor a régebbi visszamásolható a helyére. Példa a telepítés elvégzésére:

```
dkms install -m qla2x00 -v v6.04.00 -k
2.4.20-8smp
```

A parancs futása során létrejön egy `/var/dkms/qla2x00/v6.04.00/kernel-2.4.20-8smp - /var/dkms/qla2x00/v6.04.00/2.4.20-8smp` közvetett hivatkozás. A DKMS ilyen hivatkozások segítségével tartja nyilván, hogy mely rendszermaghoz melyik illesztőprogram-változat van telepítve. Mint említettem, ha azonos névvel már létezik modul, a DKMS ennek másolatát a saját fájába, a `/var/dkms/module-name/original_module/` könyvtárba menti.

Esetünkben a modul a `/var/dkms/qla2x00/original_module/2.4.20-8smp/` könyvtárba kerül.

A DKMS-féle életciklus a modulok eltávolításával, a fából való kivételével ér véget. Az eltávolítás során a rendszer törli a korábban telepített modult, és – ha van ilyen – az eredeti helyezi vissza. Ha a DKMS-fán belül egy modulnak több változata is megtalálható, akkor az eltávolítás során a DKMS nem tesz kísérletet arra, hogy kitalálja, melyik korábbi változatot szeretnénk újra használatba venni. Ehelyett, ha rendelkezünk valós `original_module` mentéssel az eredeti DKMS-telepítésből, akkor ez kerül vissza a rendszermagba. A modul minden egyéb változata ugyanúgy marad. Példa az eltávolításra:

```
dkms uninstall -m qla2x00 -v v6.04.00
↳ -k 2.4.20-8smp
```

Ha a rendszermag változatát meghatározó átadott értéket elhagyjuk, a program az éppen futó rendszermagot veszi figyelembe. Hasonló célokra használható a `remove` parancs is. A `remove` és az `uninstall` hasonló egymáshoz: a `remove` is végrehajtja azokat a lépéseket, amelyeket az `uninstall` elvégez. Ha az eltávolított modulváltozat az adott változat utolsó példánya a rendszeren lévő összes rendszermaghoz, akkor a `remove` parancs használatakor az eltávolítási folyamat elvégzése után a rendszer az adott modul összes nyomát eltünteti a DKMS-fából. Másként fogalmazva: amikor az `uninstall` parancs lefut, a modulok „lefordított” állapotban maradnak. Ezzel szemben a `remove` parancs használatakor az `add` parancsig kell visszalépnünk, ha újra használni akarjuk az adott modult a DKMS-sel. Lássunk két példát a `remove` használatára:

```
dkms remove -m qla2x00 -v v6.04.00
↳ -k 2.4.20-8smp
dkms remove -m qla2x00 -v v6.04.00 --all
```

Az első `remove` parancs a modul eltávolítását végzi el. Ha ez a modulváltozat semmilyen más rendszermaghoz nincs telepítve, akkor a program nyomtalanul eltünteti a DKMS-fából. Ha például a `qla2x00/v6.04.00` a `2.4.20-8bigmem` rendszermaghoz is telepítve van, akkor az első `remove` parancs érintetlenül hagyja a DKMS-fában. Nem így a példa második parancsa. Ez a `qla2x00/v6.04.00` modul összes változatát minden rendszermagból eltávolítja, majd a `qla2x00/v6.04.00` összes hivatkozását teljesen eltünteti a DKMS-fából. A `remove` paracssal tehát megtisztítható a DKMS-fa.

További DKMS-parancsok

A DKMS jól használható állapotlekérdező paracssal rendelkezik, amely a fában éppen megtalálható tartalomról ad tájékoztatást. Ha kapcsolók nélkül használjuk, akkor az összes elérhető adatot visszaadja. Értelemszerűen a kapott adatok jellege a lekérdező parancs futtatásakor megadott kapcsolóktól függ. Minden bejegyzéshez vagy hozzáadott (`added`), vagy lefordított (`built`), vagy telepített (`installed`) állapot tartozhat. Ha az eredeti modult is mentette a program, akkor erről is kapunk visszajelzést. Néhány példa az állapot lekérdezésére:

```
dkms status
dkms status -m qla2x00
dkms status -m qla2x00 -v v6.04.00
```

```
dkms status -k 2.4.20-8smp
dkms status -m qla2x00 -v v6.04.00
↳ -k 2.4.20-8smp
```

A DKMS egy másik fontos szolgáltatása a `match` parancs által érhető el. A `match` parancs megvizsgálja a DKMS által adott rendszermaghoz telepített modulok beállításait, és egy másik rendszermagra alkalmazza őket. A `match` lefutása után ugyanazok a modulváltozatok, amelyeket egy rendszermaggal már használunk, egy másik rendszermaghoz is telepíthetők. Erre elsősorban akkor lehet szükség, ha újabb rendszermagra frissítünk, ám meg szeretnénk tartani meglévő DKMS moduljainkat. Például:

```
dkms match --templatekernel 2.4.20-8smp
↳ -k 2.4.20-9smp
```

A `--templatekernel` az a rendszermag, amelynek alapján a beállításokat meghatározzuk. A `-k` rendszermag az, amelyre a beállításokat alkalmazzuk.

A DKMS rendszerfelügyeleti célokra az `mktarball` és az `ldtarball` parancsot is támogatja. Ezekkel a parancsokkal a felhasználó `.tar`-állományokat hozhat létre és nyithat meg, valamint illeszthet be a DKMS-fába, például arra az esetre, ha a DKMS segítségével több hasonló rendszerbe kell valamit telepítenie. Így a rendszergazda egyetlen számítógépen készítheti el az összes modult. Nincs szükség arra, hogy ugyanazt a modult minden gépen külön lefordítsuk, ehelyett a már lefordított futtatható állományt kell csak beilleszteni az adott géphez tartozó DKMS-fába. Az `mktarball` parancsral adott modulváltozat forrásából lehet `.tar`-állományt készíteni. Így menthető minden olyan magváltozat DKMS-fába, amely a megadott modulváltozat alapján fordított modullal rendelkezik. Példa:

```
dkms mktarball -m qla2x00 -v v6.04.00
↳ -k 2.4.20-8smp,2.4.20-8
```

A `-k` átadott értéktől függően az `mktarball` csak a megadott rendszermagok futtatható állományait archiválja. Ha ilyen értéket nem adunk át, akkor az adott modulváltozat összes lefordított futtatható állományát összecsomagolja.

Az `ldtarball` használatakor a DKMS egyszerűen kibontja az `mktarball` segítségével létrehozott archív állományt, és a benne találtakat beilleszti a futtató rendszer DKMS-fájába. Ennél az eljárásnál minden modul lefordított állapotban marad, majd a `dkms install` parancsral telepíteni lehet a futtatható modulokat a `/lib/modules` fába. Rendes körülmények között az `ldtarball` nem írja felül az adott rendszer DKMS-fájában már meglévő állományokat. Ha mégis erre van szükség, a `--force` kapcsolóval írhatjuk elő a tartalmának az archív állományával történő felülírását. Példa az `ldtarball` használatára:

```
dkms ldtarball --config
↳ qla2x00-v6.04.00-kernel2.4.20-8smp.tar.gz
```

A DKMS parancsai közül utolsóként az `mkdriverdisk`-et említeném meg. Mint neve is utal rá, az `mkdriverdisk` kikeresi a megfelelő forrásokat a DKMS-fából, majd egy illesztőprogramot tartalmazó hajlékonylemez-lenyomatfájlt készít, amit frissített illesztőprogramok terjesztésére lehet használni. Példa az `mkdriverdisk` használatára:

Példa a `.spec` fájlra

```
%define module qla2x00

Summary: Qlogic HBA module
Name: %module_dkms
Version: v6.04.00
Release: 1
Vendor: Qlogic Corporation
Copyright: GPL
Packager: Gary Lerhaupt
↳ <gary_lerhaupt@dell.com>
Group: System Environment/Base
BuildArch: noarch
Requires: dkms gcc bash sed
Source0: qla2x00src-%version.tgz
Source1: dkms.conf
BuildRoot: %{_tmppath}/%{name}-%{version}-
↳ %{release}-root/

%description
This package contains Qlogic's qla2x00
↳ HBA module meant
↳ for the DKMS framework.

%prep
rm -rf qla2x00src-%version
mkdir qla2x00src-%version
cd qla2x00src-%version
tar xvzf
↳ $RPM_SOURCE_DIR/qla2x00src-%version.tgz

%install
if [ "$RPM_BUILD_ROOT" != "/" ]; then
    rm -rf $RPM_BUILD_ROOT
fi
```

A lista folytatását lásd a következő oldalon.

```
dkms mkdriverdisk -d redhat -m qla2x00
↳ -v v6.04.00 -k 2.4.20-8BOOT
```

Jelenleg egyedül a Red Hat illesztőprogram-lemezformátumának a támogatása érhető el, ám a közösség segítségével a többi terjesztés követelményei és formátumai is teljesíthetők, támogathatók. A DKMS által a Red Hat illesztőprogram-lemezek készítéséhez igényelt külön fájlokról és formátumokról a <http://people.redhat.com/dledford> weboldalon lehet bővebb tájékoztatást találni. Ezeket a fájlokat a modulforrások könyvtárában kell elhelyezni.

A `dkms.conf` beállítófájl formátuma

A DKMS-csomagokkal foglalkozók számára a `dkms.conf` beállítófájl az egyetlen olyan külső forrás, amelyre a forrás `.tar`-állományok DKMS-re való készítéséhez szükség van. A beállítófájl valójában héjváltozók listáját tartalmazza, a DKMS ezek alapján végzi munkáját a csomaggal. Példaként

A lista folytatása az előző oldalról.

```

mkdir -p
↳ $RPM_BUILD_ROOT/usr/src/%module-%version/
install -m 644 $RPM_SOURCE_DIR/dkms.conf
↳ $RPM_BUILD_ROOT/usr/src/%module-%version/
install -m 644 qla2x00src-%version/*
↳ $RPM_BUILD_ROOT/usr/src/%module-%version

%clean
if [ "$RPM_BUILD_ROOT" != "/" ]; then
    rm -rf $RPM_BUILD_ROOT
fi

%files

%defattr(0644,root,root)

%attr(0755,root,root)
↳ /usr/src/%module-%version/

%pre

%post
/sbin/dkms add -m %module -v %version
/sbin/dkms build -m %module -v %version
/sbin/dkms install -m %module -v %version
exit 0

%preun
/sbin/dkms remove -m %module
↳ -v %version --all
exit 0

```

egy részlet a qla2x00/v6.04.00 *dkms.conf* fájlból:

```

MAKE="make all
↳ INCLUDEDIR=/lib/modules/$kernelver/
↳ build/include"
MAKE_smp="make SMP=1 all
↳ INCLUDEDIR=/lib/modules/$kernelver/
↳ build/include"
LOCATION="/kernel/drivers/addon/qla2200"
REMAKE_INITRD="yes"
MODULE_NAME="qla2200.o:qla2200_6x.o
↳ qla2300.o:qla2300_6x.o"
CLEAN="make clean"
MODULES_CONF_ALIAS_TYPE=
↳ "scsi_hostadapter"MODULES_CONF0=
↳ "options scsi_mod
↳ scsi_allow_ghost_devices=1"

```

Mint látható, a héjváltozókat csupa nagybetűvel kell írni. Ez alól az egyik kivétel a `MAKE` utasítás. A DKMS az általános `MAKE` parancsot használja a modulok lefordításához. Ha azonban létezik egy `MAKE_kernel-regex-text` parancs, és a `MAKE` utáni rész (karakterlánc-részletként) egyezést mutat

azzal a rendszerrel, amelyhez a modul készül, akkor ezt a másodlagos `make` parancsot használja. A fenti példában kiderül, hogy a DKMS a `MAKE_smp` utasítás használatára is képes, ha `smp` rendszerrel kell modult fordítani. Hasonló `PATCH` parancsok is léteznek. Ha az alulhúzás utáni szöveg egyezést mutat azzal a rendszerrel, amelyhez a modul készül, akkor elsőként ez a `make` kerül be a modul forrásába. Így a fejlesztőknek csak egyetlen forrás `.tar`-állományt kell terjeszteniük, amelyhez egy *dkms.conf* fájl és akár több `make` tartozik. Természetesen szükség szerint különféle `make`-eket is lehet építeni a forrásba, így biztosítva, hogy minden modul minden rendszerrel megfelelően együttműködjön. Érdekes megjegyezni, hogy a *dkms.conf* `kernelver` változót is tartalmazhat, amelyet a fordítás alatt a program annak a rendszernek a változatszámára cserél le, amihez a modul készül. Ez különösen fontos, hiszen így mindig megfelelő `include` könyvtárakra lehet hivatkozni, amikor az éppen futótól eltérő rendszerrel fordítunk modult.

A DKMS és az RPM együttes használata

A DKMS és az RPM remekül együtt tud működni. A hibátlan működéshez mindössze olyan RPM-et kell készíteni, amely elvégzi a forrás telepítését. A forrásokat általában forrás RPM-ek segítségével telepítjük, ám ezek nem feltétlenül működnek együtt a DKMS-sel, így a forrás telepítésén túl másra nem nagyon alkalmasak. Jobban járunk, ha a forrás `.tar`-fájl egy RPM-be kerül, így el lehet helyezni a megfelelő `/usr/src(modul)-(modulváltozat)/` könyvtárba, és a megfelelő DKMS-parancsokat is meg lehet hívni. A `%post` és a `%preun` alapvetően DKMS-parancs. *Listánk*on jó példa látható egy *spec* fájlra.

A jövő

Mivel a DKMS nagyon új keretrendszer, számos dologgal kell még bővíteni, másokat ki kell venni belőle, esetleg újra kell írni – ahogy a közösség határoz. A DKMS-tervezet előrehaladásával kapcsolatosan a <http://www.freshmeat.net/projects/dkms> oldalon található naprakész tudnivalók. Kérdéseit, észrevételeit bárki felteheti vagy elküldheti, ha csatlakozik a DKMS-devel levelezési listához, amely a lists.us.dell.com/mailman/listinfo címről érhető el.

A Linux Journal 2002. decemberi számában *Linus Torvaldstól* a következő idézet jelent meg:

„Úgy látom, minden kereskedelmi szereplőnek megvan a maga ütemterve, és bár ezek sokszor ütköznek egymással, végül egy olyan állapot felé terelik az egész rendszert, amely mindannyiunk számára biztosítja a kívánt szolgáltatásokat.”
Linuxos termékek kereskedelmi értékesítőjeként a Dellnek alapvető érdeke, hogy megfelelő megoldást találjon erre a modul-, illetve rendszerrel kapcsolatos kérdésre, amivel nemcsak saját ügyfeleinek nyújthat magasabb szintű szolgáltatást, de a közösséget is segíti. Ezeket a szempontokat a DKMS tervezése során sem veszítették szem elől.

Linux Journal 2003. szeptember, 113. szám



Gary Lerhaupt (gary_lerhaupt@dell.com)

Programmérnök a Dell Linux Development csoportjánál. Részt vett a Dell Red Hat Linux alapú Oracle9i Real Application Clusters (RAC) tervezésében is. Gary az Ohio Egyetemen szerezte meg informatikai diplomáját.