

Instalacja jądra w wersji 2.6 – instrukcja krok po kroku

Instalator

Udoskonalona obsługa aplikacji multimedialnych, zwiększona wydajność systemu, lepsza obsługa sprzętu oraz nowy moduł zabezpieczeń IPsec zwiększający bezpieczeństwo systemu – to tylko niektóre z powodów, dla których warto już teraz skorzystać z nowego jądra systemu. Przedstawiamy instrukcję instalacji nowego jądra.

EVA-KATHARINA KUNST

JÜRGEN QUADE



Pięć milionów linii kodu w pliku o wielkości 32 MB – statystyka nowego jądra jest rzeczywiście imponująca [1], ale to właśnie jest przyczyną tego, że konfiguracja i instalacja jądra 2.6 na pewno nie należą do prostych czynności. Jeżeli mieliśmy już wcześniej jakieś doświadczenia z kompilacją i konfiguracją jądra 2.4, nie powinniśmy napotkać większych problemów z jądrem w wersji 2.6. Musimy tu jednak ostrzec mniej doświadczonych użytkowników – aktualizacja jądra systemu jest bardzo złożoną modyfikacją całego systemu, która może skończyć się katastrofą dla nierozważnego użytkownika.

W niniejszym artykule opiszemy prostą drogę do uruchomienia standardowego jądra w wersji 2.6. Dostosowanie jądra do indywidualnych potrzeb danego środowiska wymaga już więcej cierpliwości. Dodatkowe informacje

umieszczono na stronie [2] w Post-Halloween Document, którego autorem jest Dave Jones oraz w archiwum listy dyskusyjnej jądra Linuksa, znajdującym się pod adresem [4].

Pocieszające jest to, że w odróżnieniu od większości projektów Open Source, do kompilacji jądra nie potrzeba żadnych dodatkowych programów. Wszystkie niezbędne na-

Jądro 2.6 w systemie SuSE

System SuSE 9.0 jest gotowy do współpracy z jądrem w wersji 2.6. Wraz z dystrybucją dostarczane jest jądro 2.6 w wersji testowej, nowe wersje są dostępne pod adresem [7]. Możesz jednakże napotkać pewne problemy korzystając z polecenia `/usr/sbin/hwbootscan` w połączeniu z nowym jądrem systemu. W takim przypadku trzeba wyłączyć ten program, umieszczając znak komentarza „#” przed zmienną `HWBOOTSCAN_BIN` w skrypcie `/etc/init.d/hwscan`, jak to pokazano poniżej.

```
# HWBOOTSCAN_BIN=/usr/sbin/hwbootscan
```

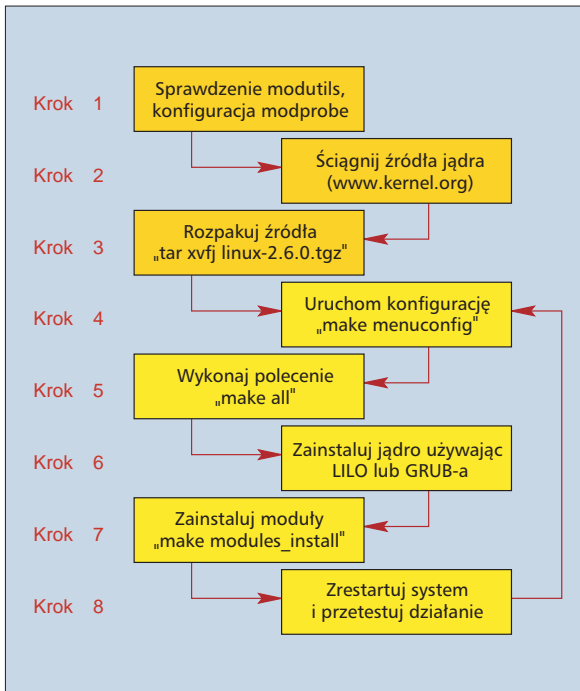
Jądro 2.6 dostarczone z dystrybucją ułatwia kompilację jądra dostosowanego do potrzeb użytkownika, gdyż możemy do tego celu wykorzystać jako szablon istniejącą konfigurację. Aby zainstalować nowe jądro testowe, należy skorzystać z pakietów RPM (nie można użyć programu YaST). Następnie powinniśmy skopiować plik konfiguracyjny jądra 2.6 do

katalogu z obecnymi źródłami Linuksa:

```
cp vmlinuz-2.6.0-default.config /usr/src/linux-2.6.0/
```

Po skopiowaniu pliku należy ponownie wywołać polecenie `make menuconfig`. W ten sposób nowe opcje zostaną automatycznie dołączone do konfiguracji jądra. Oczywiście musisz pamiętać, żeby wcześniej zainstalować narzędzia programistyczne – kompilator gcc, polecenie `make`, `ncurses` i `Qt3-devel` (niezbędny dla działania polecenia `xconfig`).

Po konfiguracji jądra powinieneś wykonać kroki 5-8, zgodnie z zaleceniami opisanymi w artykule. Po dodaniu niektórych modułów konieczne może być ręczne edytowanie pliku `/etc/modprobe.conf`. System SuSE przechowuje dodatkowe wpisy w pliku `/etc/modprobe.conf.local`. Składnię takiego pliku konfiguracyjnego opisano na stronach z instrukcją obsługi (`man modprobe.conf`).



Rysunek 1. Osiem kroków do stworzenia własnego jądra. Początkujący użytkownicy powinni przyzwyczaić się do częstego wykonywania kroków 4-8.

rzędzia, czyli kompilator i asembler oraz program make, są dostępne prawie w każdej dystrybucji Linuksa. W najgorszym wypadku trzeba doinstalować pakiety programistyczne. Do kompilacji i instalacji jądra sys-

temu można wykorzystać w zasadzie dowolną wersję pakietu gcc (2.95, 3.2 lub 3.3) oraz make w wersji 3.80 lub nowszej. Jądro 2.6 jest praktycznie niewrażliwe na różne wersje tych programów. Kompilując jądro naszego redakcyjnego systemu Debian, nie mieliśmy problemów z żadną z trzech wymienionych powyżej wersji. Niestety, zależności pakietu modutil (Krok 1 na Rysunku 1) komplikują nieco sprawę. Pakiet modutil jest odpowiedzialny za ładowanie i zwalnianie modułów jądra, lecz twórcy jądra 2.6 całkowicie przeprojektowali ten mechanizm działania. Moduły jądra 2.6 wymagają do pracy najnowszej wersji insmod. W razie potrzeby (np. starszego modułu) może być uruchomiona starsza wersja insmod.

temu można wykorzystać w zasadzie dowolną wersję pakietu gcc (2.95, 3.2 lub 3.3) oraz make w wersji 3.80 lub nowszej. Jądro 2.6 jest praktycznie niewrażliwe na różne wersje tych programów. Kompilując jądro naszego redakcyjnego systemu Debian, nie mieliśmy problemów z żadną z trzech wymienionych powyżej wersji.

Niestety, zależności pakietu modutil (Krok 1 na Rysunku 1) komplikują nieco sprawę. Pakiet modutil jest odpowiedzialny za ładowanie i zwalnianie modułów jądra, lecz twórcy jądra 2.6 całkowicie przeprojektowali ten mechanizm działania. Moduły jądra 2.6 wymagają do pracy najnowszej wersji insmod. W razie potrzeby (np. starszego modułu) może być uruchomiona starsza wersja insmod.

insmod.old w katalogu */sbin/* lub */usr/local/sbin/*, znaczy to, że *insmod* powinien poradzić sobie z nową wersją jądra. Na szczęście wiele obecnych dystrybucji Linuksa posiada już zaktualizowaną wersję *insmod* (np. SuSE 8.2 i 9.0). Użytkownicy Debiana (i Knoppix-a) mogą w tym celu uruchomić program *apt-get*.

```
apt-get install module-init-tools
```

Użytkownicy innych systemów mogą pobrać najnowszą wersję *modutils* ze strony [3], np. *0.9.15-pre4*. Po uruchomieniu programu utworzy on katalog *module-init-tools.0.9.15-pre4*, w którym odnajdziesz plik *README*, zawierający zwięzły opis najważniejszych szczegółów. Przede wszystkim nakazuje on wykonanie następujących poleceń:

```
./configure >
--prefix=/
make moveold
make
make install
```

Program oryginalny (*/sbin/insmod/*) zostanie przeniesiony do */sbin/insmod.old*. Nowy program – */sbin/insmod* będzie wywoływał poprzednią wersję, gdy tylko system załaduje jądro w wersji 2.4 – będzie jednak korzystał z nowej wersji przy ładowaniu modułów jądra w wersji 2.6. Ta sama zasada dotyczy także programów */sbin/rmmmod* i */sbin/mdoprobe*.

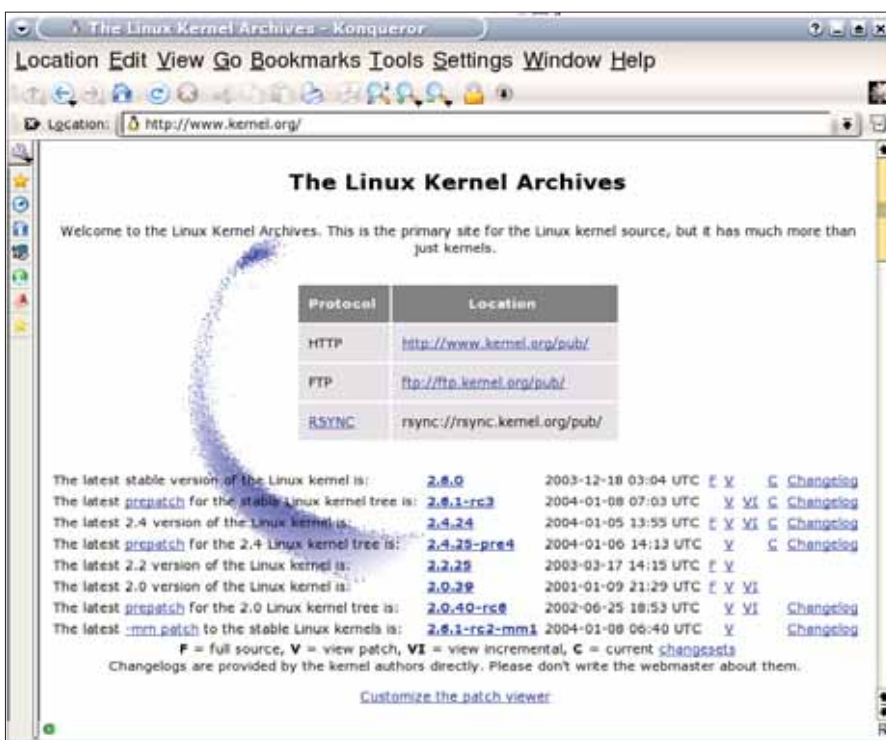
Nowe pliki konfiguracyjne

Nowe programy pakietu *modutils* wymagają nowego pliku konfiguracyjnego: */etc/modprobe.conf*. Jeżeli podany plik nie istnieje, skopiuj plik szablonu z */generate-modprobe.conf* do */etc/modprobe.conf*.

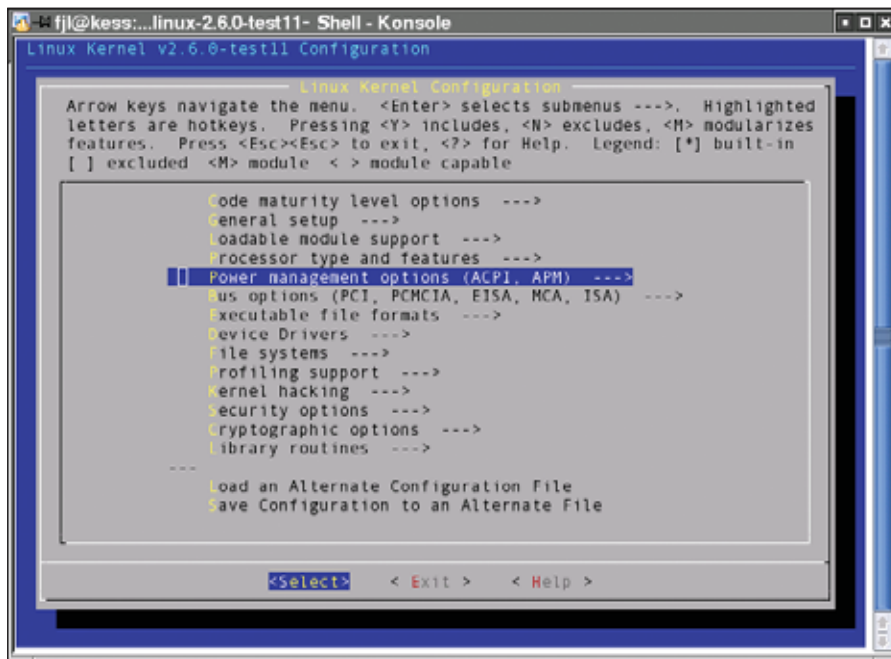
Po wykonaniu opisanych czynności możesz przejść do kroku 2 (patrz Rysunek 1), czyli do pobrania źródeł jądra 2.6 z sieci Internet. Informacje o bieżącej wersji jądra widać zawsze na stronie powitalnej serwisu [1]. Aby pobrać pełną wersję jądra (a nie tylko poprawkę), musisz kliknąć na wersji oznaczonej literą F (jak „full”), która znajduje się po prawej stronie opisów (patrz Rysunek 2). Źródła należy umieścić w katalogu */usr/src/* (krok 3):

```
cd /usr/src
tar xvfj /tmp/>
linux-2.6.0.tar.bz2
```

Najtrudniejszym i najbardziej czasochłonnym zadaniem jest krok 4 – konfiguracja jądra (zaraz po rozpakowaniu archiwum). Na-



Rysunek 2. Jądro posiada pełen zestaw swoich wersji. Na ekranie powitalnym znajduje się odnośnik do bieżącej wersji. Poprzednie wersje są dostępne przez odnośniki umieszczone w tabeli na górze strony.

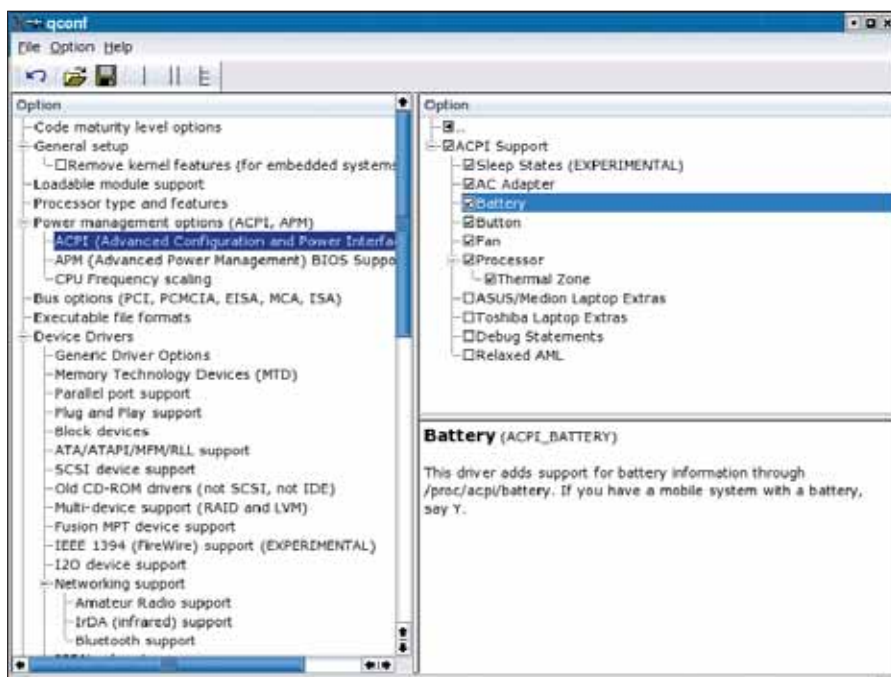


Rysunek 3. Konfigurowanie jądra stało się bardziej zrozumiałe, mimo ogromu dostępnych opcji w wersji 2.6. Rysunek przedstawia konfigurację programu menuconfig podczas pracy.

szym zdaniem najlepiej nadaje się do tego polecenie `make menuconfig` (patrz Rysunek 3). Dzięki niemu tworzony jest program konfiguracyjny oparty na bibliotece `ncurses`, który możesz uruchomić w trybie znakowym. Dostępna jest także graficzna odmiana tego programu (oparta na Qt), co pokazano na Rysunku 4. Aby użyć standardowej wersji graficznej, należy wydać polecenie `make xconfig`.

Cierpliwość

Pomimo wielkiej liczby dostępnych opcji, proces konfiguracji jest teraz bardziej intuicyjny. Na początek powinien zmienić domyślne ustawienie typu procesora (menu *Processor type and features*) i typu architektury. Zwróć szczególną uwagę na system plików, np. jeżeli jesteś użytkownikiem systemu SuSE, będziesz z całą pew-



Rysunek 4. Program xconfig oparty na Qt jest bardzo przejrzysty. Obszary konfiguracji umieszczono po lewej stronie, opcje indywidualne na górze, a poniżej nich znalazło się miejsce na tekst pomocy dla obecnie wybranego parametru.

nością potrzebować obsługi ReiserFS. Jeżeli jednak nie potrzebujesz funkcji rzadziej wykorzystywanych (np. obsługi SCSI), lepiej je wyłączyć. W zasadzie dla każdej opcji wyboru dostępna jest pomoc. Przegląd i poznanie wszystkich opcji menu za pierwszym razem zajmie zapewne około dwóch lub trzech godzin. Pamiętając jednak o ponad pięciu milionach linijek kodu, które w ten sposób konfigurowujemy, trzeba stwierdzić, że i tak jest to duże osiągnięcie.

Jeżeli posiadasz wcześniejszą, działającą konfigurację jądra 2.6 (patrz ramka Jądro 2.6 w systemie SuSE), to właśnie na niej trzeba oprzeć nową konfigurację. Jeżeli nie dysponujesz taką konfiguracją, najlepiej na początku zdecydować się na opcję minimalistyczną – wybrać tylko te elementy systemu, które są niezbędne. Gdy tylko nowe jądro będzie działać, dodanie nowych funkcji nie będzie problemem.

Kompilacja

Po zachowaniu wszystkich opcji w pliku konfiguracyjnym można zająć się kompilacją jądra. Użytkownicy systemów Debian lub Knoppix powinni wcześniej przeczytać ramkę Kroki 5-7 w systemie Debian.

Użytkownicy innych systemów powinni po prostu uruchomić polecenie kompilacji (krok 5). Od tej chwili można ignorować ostrzeżenia przebiegające przez ekran. W systemie z dużą ilością pamięci z procesorem Intel 2.6 GHz, operacja kompilacji jądra trwa około 10 minut.

Kroki 5-7 w systemie Debian

Wersja niestabilna Debiana (Sarge) oferuje duży wybór, jeśli chodzi o jądro 2.6. Dopasowanie i stworzenie jądra, które zapewniłoby optymalną obsługę twojego systemu, wymaga jednak wykonania pewnych czynności ręcznie. Po zakończeniu konfiguracji (krok 4) wywołujemy polecenie `make-kpkg` w katalogu kodu źródłowego:

```
./make-kpkg kernel_image
```

Skrypt generuje jądro i moduły (krok 5), a także tworzy pakiet `kernel_image` dla Debiana, zapisując go w katalogu źródeł (zwykle `/usr/src`). Przed wykonaniem tego (kroki 6 i 7) powinien dodać nową opcję w pliku konfiguracyjnym `bootloader-a`. Instalator Debiana, wraz z uruchomieniem `dpkg -i kernel-image-numer_wersji.deb`, uruchomi także LILO. Dzięki temu LILO wyświetli odpowiedniej opcji w module ładującym rozruchu.

Instalacja

Jeżeli wszystko przebiegło zgodnie z planem, następnym krokiem jest instalacja jądra (kroki 6-7). Instalacja modułów jest bardzo prosta. Wystarczy wykonać polecenie `make modules_install`, po kilku chwilach moduły pojawią się w nowym katalogu o nazwie `/lib/modules/2.6.0/`. Konieczne będzie jednak ręczne skopiowanie jądra do katalogu `/boot/`:

```
cp /usr/src/linux-2.6.0/arch/
i386/boot/bzImage /boot/
vmlinuz-2.6.0
```

Po uruchomieniu komputera w module ładującym rozruchu (ang. boot loader) powinna pojawić się opcja dla nowego jądra systemu. Dla LILO [5] oznacza to edycję pliku `/etc/lilo.conf`, a dla GRUB [6] pliku `/boot/grub/menu.lst`. Nowy wpis powinien zawierać nazwę jądra (np. `/boot/vmlinuz-2.6.0`) i wybraną nazwę dla pozycji menu. Jeżeli do jądra dodasz później nowy sterownik, zwykle nie trzeba już edytować plików `lilo.conf` i `menu.lst`. Jeżeli korzystasz z LILO jako bootlader-a, przed ponownym uruchomieniem systemu wykonaj polecenie `lilo`, które zaktualizuje konfigurację.

Uruchomienie próbnego

Możemy przystąpić do próbnego uruchomienia komputera. Jeżeli jądro nie wystartuje poprawnie, trzeba będzie szukać problemu na konsoli. Wyświetlane tam komunikaty zazwyczaj dostarczają odpowiednich wskazówek i informują, dlaczego jądro nie mogło dokończyć procesu uruchomienia systemu. W wielu przypadkach jest to spowodowane np. brakiem sterownika, co uniemożliwia dostęp do twardego dysku lub systemu plików.

Jeżeli ekran monitora pozostaje czarny jak noc, odszukanie winowajcy może być trudne. Zwykle w ten sposób objawia się problem z buforem ramki. Możesz wtedy działać na dwa sposoby:

- usunąć wpis `vga=XXX` z pliku `/etc/lilo.conf`, lub włączyć opcję `vga=normal`.
- sprawdzić, czy w opcji `CONFIG_VT`, w pliku konfiguracyjnym jądra `/usr/src/linux-2.6.0/.config` ustanowiono wartość `y`.

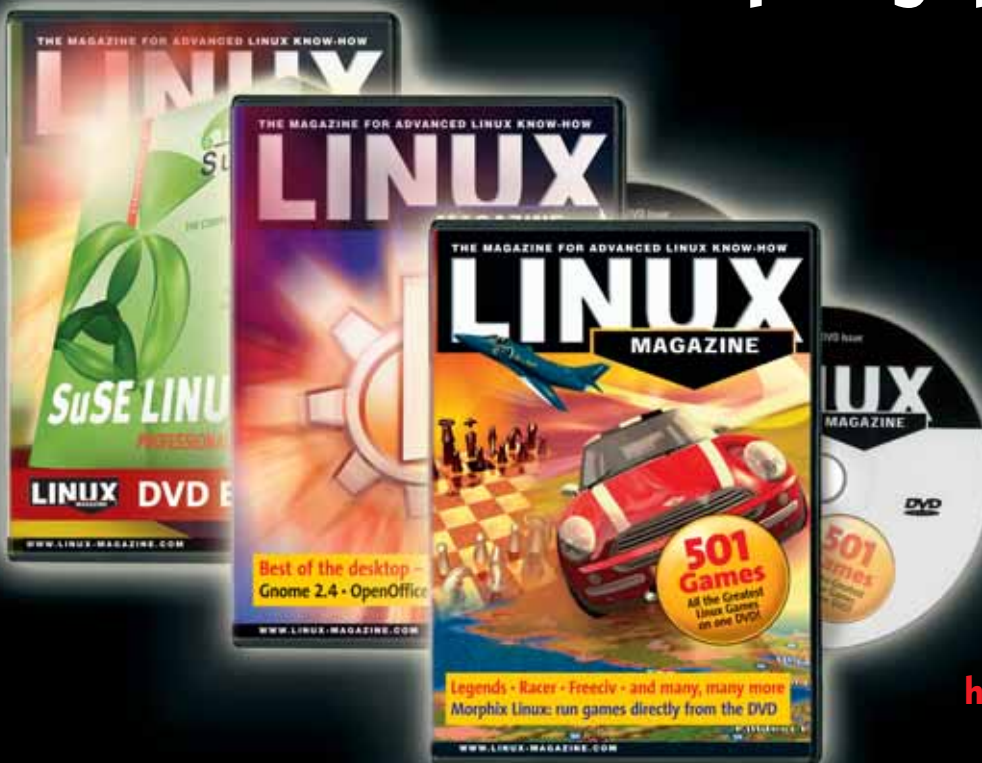
Jeżeli uruchomienie nowego jądra systemu zakończy się powodzeniem, można przystąpić do rekomilacji sterowników urządzeń, których nie obsługuje jądro. Dobrym przykładem są karty graficzne oparte na procesorach firmy nVidia. Więcej informacji znajdziesz pod adresem [8].

Na ostatnim miejscu jest sprawdzenie komunikatów pojawiających się podczas uruchomienia systemu. Dzięki poleceniu `dmesg` można przejrzeć komunikaty wygenerowane w czasie startu jądra. Jeżeli napotkasz jakieś problemy, trzeba wrócić do kroku 4. Jeżeli wszystko jest OK, to składamy gratulacje nowemu użytkownikowi jądra Linuksa w wersji 2.6. ■

INFO

- [1] Kod źródłowy jądra Linuksa:
<http://www.kernel.org>
- [2] Dave Jones, The Post-Halloween Document: <http://www.linux.org.uk/~davej/docs/post-halloween-2.6.txt>
- [3] Nowy pakiet modutils:
<http://www.kernel.org/pub/linux/kernel/people/rusty/modules/>
- [4] Lista dyskusyjna Linux Kernel:
<http://lkml.org>
- [5] LILO: <http://lilo.go.dyndns.org/>
- [6] GRUB: <http://www.gnu.org/software/grub/grub.html>
- [7] Jądro 2.6 dla SuSE 9.0:
<ftp://ftp.suse.com/pub/suse/i386/9.0/unsigned/kernel-2.6/>
- [8] Moduły jądra nVidia:
<http://www.minion.de/>

Prenumerata Linux Magazine Nie przegap takiej okazji



- Zamawiając prenumeratę oszczędzasz!
- Płacisz jak za 9 numerów, a otrzymujesz 12!
- Z każdym numerem DVD lub płyta CD-ROM.

Najszybszy sposób
zamówienia prenumeraty:

<http://www.linux-magazine.pl>

Infolinia: 0801-800-105