Eksperymenty z Arch Linux

Jarosław Świerczyński

Arch Linux zdobywa coraz większą popularność wśród użytkowników Linuksa. Głównymi atutami tej dystrybucji są prostota i nowoczesność. Dzięki wspaniałej pracy deweloperów w łatwy sposób można skorzystać z ciekawych rozwiązań.

Arch może zostać zainstalowany na szyfrowanych partycjach, wykorzystując standard LUKS (Linux Unified Key Setup). W momencie pisania tego artykułu najnowsze oficjalne płyty instalacyjne były opatrzone numerem 0.7.2. Od chwili wydania tej wersji wiele w dystrybucji się zmieniło i bardzo łatwo wpaść w jedną z kilku pułapek czyhających na nowych użytkowników. Dlatego zanim światło dzienne ujrzy wersja 0.8, zalecane jest użyć nieoficjalnych płyt instalacyjnych, przygotowanych przez jednego z developerów, które prawdopodobnie będą podstawą do stworzenia krążków z nadchodzącym nowym wydaniem. Obrazy tych płyt można znaleźć pod adresem *www.archlinux.org/~tpowa/newisos*. Jeżeli instalacja z serwera FTP nie stanowi problemu dla posiadanego łącza, najlepiej jest wybrać właśnie tę metodę.

Rysunek 1. Nowa płyta instalacyjna pozwala także przetestować pamięć operacyjną

Tajne przez poufne

Po uruchomieniu systemu znajdującego się na płycie instalacyjnej, a jeszcze przed instalatora, przystępujemy do partycjonowania dysku, mając do dyspozycji narzędzia cfdisk i fdisk. Absolutnie konieczna jest osobna partycja na katalog */boot*, na której będą przechowywane nieszyfrowane jądro oraz pliki używane przez program rozruchowy. Jeżeli zdecydujemy się na system plików ext2 dla tej partycji, wystarczy przeznaczyć na nią 25 megabajtów. Oprócz partycji głównej stwórzmy także partycję wymiany oraz przynajmniej jedną partycję dla katalogu, na przykład */home*.

Następnie ładujemy moduły jądra dodające obsługę przezroczystego szyfrowania urządzeń blokowych oraz symetrycznego szyfru blokowego AES (*Advanced Encryption Standard*):

```
modprobe dm-crypt
modprobe aes-i586
```

Teraz należy utworzyć szyfrowane urządzenia na partycjach. Dla każdej z nich, oprócz partycji wymiany i */boot*, wykonujemy następujące polecenie:

```
cryptsetup -y luksFormat /dev/xy
```

gdzie `x` jest oznaczeniem dysku (np. `hda`, `sdb`), a `y` numerem partycji. Za każdym razem zostaniemy poproszeni o potwierdzenie (należy wpisać „YES”) oraz zapytani o hasło. Dobrze jest każdej partycji przyporządkować inne hasło.

Ostatni etap przed uruchomieniem programu instalacyjnego to otwarcie szyfrowanych urządzeń. I znów dla każdej szyfrowanej partycji z osobna wydajemy polecenie:

```
cryptsetup luksOpen /dev/xy nazwa
```

gdzie `nazwa` to łańcuch znaków, złożony z liter i cyfr, jednoznacznie identyfikujący szyfrowane urządzenie, na przykład `root` (dla partycji głównej), `home`, `varlog` itd. Po tych czynnościach w katalogu */dev/mapper* powinny pojawić się pliki urządzeń o określonych wcześniej nazwach, czyli */dev/mapper/root*, */dev/mapper/home* itd. Można ich używać jak każdych innych urządzeń partycji, jednak pisząc do nich, dane najpierw są szyfrowane, a dopiero potem umieszczane na rzeczywistych partycjach.

Nareszcie można uruchomić instalator (poleceniem `/arch/setup`). Większość procesu instalacji przebiega identycznie jak w przypadku niezastosowania szyfrowania, jednak w kilku momentach należy odstąpić od standardowej procedury.

Po pierwsze w menu „Prepare Hard Drive” pomijamy kroki „Auto-Prepare” oraz „Partition Hard Drives” i od razu przechodzimy do „Set Filesystem Mountpoints”. Na pytanie o partycję wymiany odpowiadamy opcją „NONE”. Natomiast zapytani o partycję główną, wybieramy urządzenie zmapowane */dev/mapper/root*. Podobnie czynimy z pozostałymi partycjami oprócz tej przeznaczonej na katalog */boot* – tylko dla niej wybieramy prawdziwe urządzenie partycji.

Przy wyborze pakietów należy zwrócić szczególną uwagę na to, aby zaznaczony został pakiet cryptsetup z kategorii base. Po instalacji pakietów w menu „Configure System” znajdziemy plik */etc/mkinitcpio.conf*. Na podstawie zawartej w nim konfiguracji przeprowadzana jest pierwsza faza uruchamiania systemu. Aby jądro potrafiło używać zmapowanego urządzenia zaszyfrowanej partycji głównej, dodajemy element `encrypt` do znajdującej się na końcu pliku */etc/mkinitcpio.conf* listy `HOOKS`. Bardzo ważne jest, aby `encrypt` znalazło się przed `filesystems`, w przeciwnym wypadku nie zostanie załadowany moduł obsługi systemu plików użytego na partycji głównej.

Pora na ustawienie pozostałych zaszyfrowanych partycji. W tym celu przechodzimy na inną konsolę (naciskając np. [*Alt*+*F2*]) i edytujemy plik */mnt/etc/crypttab* (posługując się edytorem vi lub nano). Dla każdej z partycji dodajemy podobny wpis:

```
home    /dev/hda5
  "haslo_ustawione_dla_partycji_home"
```

Zwróćmy uwagę, że używamy nazwy rzeczywistego urządzenia, a nie zmapowanego. Na koniec zajmujemy się partycją wymiany. W tym samym pliku umieszczamy następujący kod:

```
swap    /dev/hda2    /dev/urandom
-c aes-cbc-essiv:sha256 -h sha256 -s 256
```

Oczywiście */dev/hda5* zastępujemy urządzeniem, które ma być partycją wymiany w naszym systemie. Co oznacza dalsza część wpisu? Otóż zamiast hasła do szyfrowania będzie używany plik powstały z 256 pseudolosowych bitów, pobieranych z generatora */dev/urandom*. Ponieważ z każdym startem systemu partycja wymiany będzie szyfrowana innym kluczem, nie będzie użyteczna, zanim jej nie sformatujemy. W związku z tym do pliku */mnt/etc/rc.local* dodajemy poniższe polecenia:

```
mkswap /dev/mapper/swap
swapon /dev/mapper/swap
```

Wracamy do instalatora naciskając [*Alt*]+[*F1*] i jeżeli dokonaliśmy już edycji pozostałych plików z menu „Configure System", możemy przystąpić do instalacji jądra, a następnie programu rozruchowego. Aby uniknąć problemów, sugerujemy użyć programu GRUB. Edytując plik konfiguracyjny */mnt/boot/grub/menu.lst* należy zamienić zmapowane urządzenie partycji głównej na rzeczywiste, czyli zamiast */dev/mapper/root* powinno być np. */dev/hda3*. Ponadto w opcji `root` trzeba podać partycję z katalogiem */boot*, zachowując notację używaną przez program GRUB, tj. `(hdx,y)` gdzie `x` jest numerem dysku licząc od zera, a `y` numerem partycji, również licząc od zera.

Rysunek 3. Prawidłowo skonfigurowane szyfrowanie partycji dysku

Po zainstalowaniu systemu w ten sposób przy każdym starcie będziemy pytani o hasło do partycji głównej. Jeżeli się pomylimy, dostaniemy kolejną szansę, natomiast dane na dysku nie będą w żaden sposób zagrożone. Obraz płyty instalacyjnej użyty przy pisaniu tego artykułu posiadał błąd skutkujący komunikatem jądra o braku programu init, pojawiającym się zaraz po wpisaniu hasła do partycji głównej. Do czasu ukazania się tego numeru błąd powinien być już poprawiony. Jeżeli jednak problem nadal będzie występować, należy ponownie skorzystać z płyty instalacyjnej, podając jako opcję rozruchu (po ujrzeniu napisu „boot:") urządzenie partycji głównej, na przykład:

```
arch root=/dev/hda3
```

Rysunek 2. Przy starcie systemu jesteśmy pytani o hasło do partycji głównej

Po zalogowaniu się do zainstalowanego systemu należy wydać polecenie:

```
pacman -S kernel26
```

i odpowiedzieć twierdząco na pytanie o ponowną instalację obecnej w systemie wersji pakietu. Po zakończeniu działania polecenia system powinien uruchamiać się poprawnie.

Jeżeli wydamy polecenie `mount`, zobaczymy różnicę w zamontowaniu partycji szyfrowanych i nieszyfrowanych. Natomiast posługując się poleceniem `swapon -s`, ujrzymy zmapowane urządzenie pracujące jako partycja wymiany. Szczegółowe informacje o otwartych partycjach szyfrowanych uzyskamy na przykład w następujący sposób:

```
cryptsetup status root
```

Dobrym zwyczajem przy posiadaniu osobnej partycji na katalog */boot* jest montowanie jej w trybie tylko do odczytu. W tym celu otwieramy plik */etc/fstab* i w odpowiedniej linii dodajemy opcję `ro`:

```
/dev/hda2  /boot ext2   defaults,ro  0   1
```

Bardzo ważna rzecz: przed każdą aktualizacją jądra, czyli pakietu kernel26 i podobnych, musimy przemontować tę partycję tak, aby można było na niej zapisywać. Służy do tego polecenie:

```
mount -o remount,rw /boot
```

Po aktualizacji pakietu z jądrem możemy powrócić do poprzedniego stanu:

```
mount -o remount,ro /boot
```

Każdy odnajdzie swój kernel

W dystrybucji Arch Linux mamy do dyspozycji aż pięć różnych, gotowych do użycia jąder. Oprócz oficjalnej wersji, wybrać możemy: kernel z obsługą suspend2, eksperymentalną gałąź -mm utrzymywaną przez Andrew Mortona, oficjalne jądro z łatami Cona Kolivasa, poprawiającymi responsywność (zwłaszcza na desktopie), i wreszcie szczególnie popularny wśród użytkowników dystrybucji kernel beyond, czyli jądro oparte na poprzednio wymienionym, lecz zawierające także między innymi łaty genpatches, znane z dystrybucji Gentoo.

Domyślnym jądrem, dostępnym podczas instalacji, jest oczywiście kernel oficjalny, zawarty w pakiecie kernel26. W repozytorium pakietów zawsze można znaleźć najnowszą stabilną wersję, przy czym krytyczne poprawki (oznaczane czwartą liczbą w wersji jądra) z reguły od razu trafiają do repozytorium current, natomiast nowe wersje jądra (zwiększony trzeci element numeru wersji) najpierw dowodzą swojej wartości w repozytorium testing.

Jak każde jądro w dystrybucji, tak i to jest modularne, a do ładowania sterowników wymaganych przy starcie systemu używa obrazu initramfs, generowanego przez skrypt mkinitcpio. W pliku konfiguracyjnym */etc/mkinitcpio.conf* można wyłączyć zbędne moduły, takie jak scsi, sata czy raid, w zależności od posiadanego sprzętu. Po dokonaniu zmian należy ponownie utworzyć obraz initramfs:

```
/sbin/mkinitcpio -p kernel26
```

Pakiet kernel26suspend2 zawiera jądro potrzebne do uśpienia systemu przez zapisanie zawartości pamięci operacyjnej na dysk. Tę funkcjonalność posiada także kernel beyond, natomiast jądro suspend2 jest przeznaczone dla osób, które potrzebują funkcji uśpienia, ale nie chcą używać jądra z dodatkowymi łatami, jakim jest właśnie beyond.

Jądro z gałęzi Andrew Mortona, dostępne w pakiecie kernel26mm, służy jako poligon doświadczalny dla oficjalnego jądra. Zawiera łaty jeszcze nie przetestowane na tyle dokładnie, aby mogły znaleźć się w kernelu, którego opiekunem jest Linus Torvalds. Wśród zmian warto wymienić nowy algorytm szeregowania operacji wejścia /wyjścia oraz modyfikacje nadające Linuksowi cechy systemu czasu rzeczywistego.

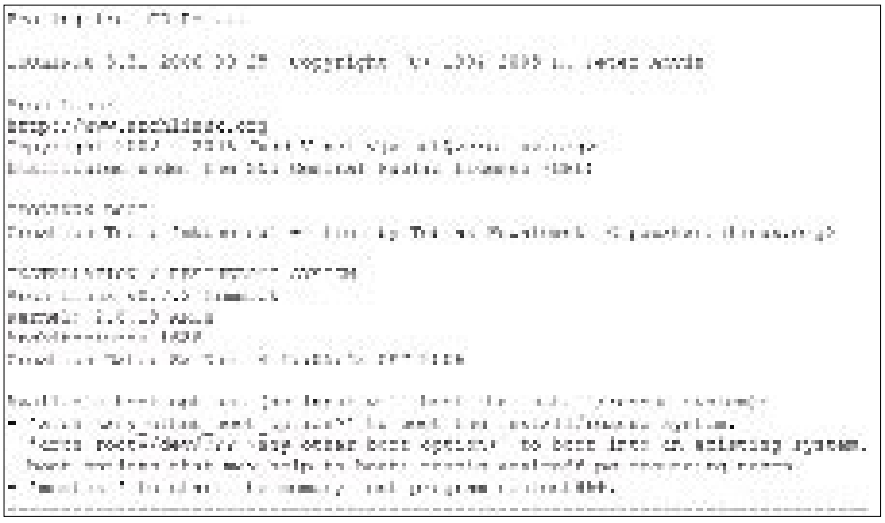
W pakiecie kernel26ck znajduje się jądro z łatami Cona Kolivasa, dzięki którym system zyska większą responsywność, co jest szczególnie przydatne na desktopach. Większa wydajność osiągana jest dzięki licznym modyfikacjom algorytmu szeregowania oraz podsystemu zarządzania pamięcią. Jeżeli zamierzamy użyć tego jądra na serwerze lub systemie bez graficznego interfejsu użytkownika, dobrze jest dodać następujące polecenie do pliku */etc/rc.local*:

```
echo 0 > /proc/sys/kernel/interactive
```

Dodatkowo dla zastosowań, które wyjątkowo intensywnie wykorzystują procesor, a bardzo rzadko przeprowadzają operacje wejścia/wyjścia, warto użyć ustawienia:

```
echo 1 > /proc/sys/kernel/compute
```

Absolutnie nie należy tego robić kiedy maszyna obsługuje bardzo duży ruch sieciowy lub po prostu wymagane są niskie opóźnienia.



Rysunek 4. Tło komunikatów pojawiających się przy starcie systemu

Zestaw łat Cona Kolivasa jest także używany przez jądro kernel26beyond, którego opiekunem jest James Rayner, jeden z developerów dystrybucji Arch Linux. Ważnym elementem tego kernela jest zbiór genpatches, używany w jądrach dystrybucji Gentoo. Wśród łat w nim zawartych możemy znaleźć poprawki błędów, ale też zupełnie nowe rzeczy, jak sterownik kart sieciowych opartych na układzie RTL8168, wyświetlający przy starcie systemu elementy graficzne czy nawet animacje fbsplash, nowy sterownik vesafb, obsługę systemu plików z kompresją danych squashfs, moduł speakup ułatwiający korzystanie z konsoli osobom niewidomym. Dodatkowo jądro beyond pozwala regulować napięcie rdzenia procesora, używać nowego Reiser4 oraz służącego do „mieszania" plików i katalogów z różnych systemów plików unionfs, przydzielać użytkownikom i grupom uprawnienia do korzystania z elementów systemu czasu rzeczywistego, a w razie problemów z ACPI (*Advanced Configuration and Power Interface*) dostosować tablicę DSDT (*Differentiated System Description Table*).

Zobaczmy w jaki sposób uzyskać graficzny start systemu używając jądra *beyond*. Przede wszystkim potrzebujemy pakietu gensplashutils z repozytorium community (które należy uaktywnić w pliku */etc/pacman.conf*). Pakiet zawiera domyślny motyw darch, przedstawiający logo dystrybucji.

Następnie edytujemy plik */etc/mkinitcpio.conf* i do listy `HOOKS` dopisujemy element `fbsplash`, a poniżej dodajemy następujący kod, określający motyw oraz rozdzielczość:

```
FBTHEMES="darch"
FBRES="1024x768"
```

Po dokonaniu zmian ponownie generujemy obraz initramfs (pamiętając, że partycja z katalogiem */boot* musi być zamontowana w trybie read-write):

```
mkinitcpio -g /boot/kernel26beyond.img
```

Na koniec trzeba dodać odpowiedni wpis do pliku konfiguracyjnego programu GRUB. Otwieramy zatem plik */boot/grub/menu.lst* i dodajemy następujące linie:

```
title Arch Linux Splash
root (hd0,0)
kernel /vmlinuz26beyond
   root=/dev/hda3 video=vesafb:
```