

Poznajemy rozwiązanie User-Mode Linux

LINUX W LINUXIE

User-Mode Linux wygląda jak Linux, bo... to jest Linux.

Ten szybki wirtualny system linuksowy można wykorzystać w najrozmaitszych zastosowaniach.

**FABRIZIO
CIACCHI**



User-Mode Linux (UML) [1] jest popularnym, uniwersalnym i w pełni funkcjonalnym wirtualnym systemem linuksowym działającym na linuksowym hoście. UML ma liczne zastosowania. Może służyć do wygodnego testowania aplikacji bez narażania systemu hosta na uszkodzenie. Zwyczajny użytkownik może przetestować w UML no-

we wersje jądra bez obaw o błędy spowodowane nowymi i niesprawdzonymi poprawkami. Administratorzy testują w UML konfiguracje systemów. W jednym hoście można nawet skonfigurować wiele wersji UML i zasymulować działanie całej sieci.

Co to jest User-Mode Linux?

User-Mode Linux nie jest ani emulatorem, ani interfejsem API (jakim jest Wine). Zasadę działania tego rozwiązania najłatwiej wyjaśnić, odwołując się do architektury jądra linuksowego.

Jądro uruchamia procesy i komunikuje się ze sprzętem. Kiedy proces chce skorzystać z urządzenia (np. wyświetlić coś na monitorze, wydrukować dokument lub skopiować plik na dyskietkę), zleca jądro linuksowemu zadanie komunikowania się z podzespołami komputera (Rysunek 1).

Natomiast w rozwiązaniu User-Mode Linux jądro jest uruchamiane jako proces. Różnica w stosunku do zwykłego jądra polega na tym, że jądro UML nie komunikuje się bezpośrednio ze sprzętem.

Informacje są przekazywane do „prawdziwego” jądra hosta linuksowego i dopiero ono faktycznie „rozmawia” z elementami komputera.

Ponieważ i system wirtualny, i system hosta są Linuksami oraz posiadają niemal iden-

SELinux w UML

Pod adresem [15] dostępny jest bardzo interesujący dokument opisujący konfigurację środowiska UML z obsługą rozwiązania SELinux. Takie środowisko może bardzo pomóc w tworzeniu bezpieczniejszych serwerów i testowaniu zbiorów zasad SELinux, bez narażania na szwank produkcyjnego systemu.

Jaki emulator wybrać?

Zalety oprogramowania UML stają się dobrze widoczne wtedy, gdy spróbujemy sklasyfikować oprogramowanie do emulacji. Wyróżniamy trzy typy takiego oprogramowania:

- emulacja programowa,
- emulacja sprzętowa,
- bez emulacji.

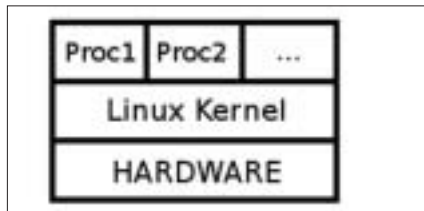
Jednym z najbardziej znanych programowych emulatorów systemów jest Bochs [4]. Jego podstawowym zadaniem jest emulacja konkretnej architektury sprzętowej (IA-32, nazywanej również x86) na bazie konkretnego systemu operacyjnego, np. Windows, MacOS, czy – oczywiście – Linux. Na emulowanym sprzęcie można zainstalować dowolny system operacyjny (Linux, Windows, DOS itd.), ale system taki działa bardzo wolno, ponieważ każda instrukcja procesora jest tłumaczona z rozumianej przez system-gościa na rozumianą przez system-host.

Emulacja sprzętowa polega na wykonywaniu kodu bezpośrednio na sprzęcie hosta. Emulator sprzętowy działa szybciej niż programowy, ale musi przechwytywać wszystkie odwołania do sprzętu. Wadą takiego rozwiązania jest fakt, że kod emulatora musi być przygotowany dla konkretnej architektury sprzętowej; architektura musi być taka sama dla środowiska hosta i systemu gościa. Przykładem emulatora działającego w ten sposób jest komercyjne oprogramowanie VMware [5].

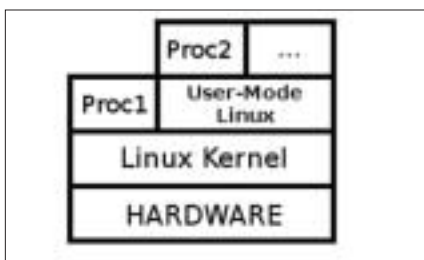
User-Mode Linux należy do ostatniej kategorii. Nie musi emulować specyficznej architektury sprzętowej; komunikuje się prawie bezpośrednio ze sprzętem komputera. Instrukcje są szybko przekazywane z jądra UML przez jądro systemu hosta. W najgorszym razie programy w środowisku UML działają o 20% wolniej niż w systemie hosta.

QEMU: Rozsądna alternatywa

Jeśli zamierzamy używać UML do testowania nowych dystrybucji Linuksa, być może lepiej jest użyć emulatora QEMU [12]. Program ten, oparty na emulatorze Bochs [13], jest bardzo prosty w instalacji, konfiguracji i użytkowaniu. Dodatkowe informacje na temat QEMU można znaleźć w artykule poświęconym tej interesującej aplikacji, który ukazał się w numerze 14 Linux Magazine (marzec 2005).



Rysunek 1: Zwykła struktura procesu liniowego.



Rysunek 2: UML działa jako proces. Na tym rysunku Proc1 działa w systemie hosta. Proc2 działa w wirtualnym systemie User-Mode Linux.

tyczną strukturę, komunikacja między nimi jest bardzo wydajna, a warstwa minimalna abstrakcji/translacji powoduje niewielkie obciążenie.

Konfiguracja UML

Oprogramowanie User-Mode Linux można zainstalować za pomocą menedżera pakietów

własnej dystrybucji. Na przykład w Debianie jako root wydajemy polecenie:

```
# apt-get install user-mode-linux2
uml-utilities kernel-patch-uml
```

W wyniku tego polecenia zostanie zainstalowane jądro UML oraz dodatkowe narzędzia. Instalacja jest równie łatwa w przypadku innych menedżerów pakietów. W razie kłopotów z instalacją tą metodą, albo gdy podczas uruchamiania występują problemy z pamięcią [2], może zajść konieczność pobrania zwykłego jądra linuksowego (zalecamy wersję 2.4.27 [3]) i zainstalowania poprawki UML [4]. Inne poprawki UML można znaleźć pod adresem [5]. Po pobraniu plików (wszystkich do tego samego katalogu), otwieramy terminal i wykonujemy następujące polecenia:

```
$ bunzip2 linux-2.4.27.tar.bz2
$ tar -xvf linux-2.4.27.tar
$ bunzip2 uml-patch-2.4.27-1.bz2
$ patch -p1 -d linux-2.4.27 >
< uml-patch-2.4.27-1
$ cd linux-2.4.27
$ make menuconfig ARCH=um
$ make linux ARCH=um
$ strip linux
```



Rysunek 3: Uruchamianie wirtualnej maszyny UML.

```

fabrizio@ubuntu:~/UserModeLinux$ ls -las
total 38844
drwxr-xr-x  2 fabrizio fabrizio   224 2005-03-09 11:49
drwxr-xr-x 12 fabrizio fabrizio  3032 2005-03-09 11:47
-rw-r--r--  1 fabrizio fabrizio 62862396 2005-03-09 13:21 root_fs_ubuntu1.7
-rw-r--r--  1 fabrizio fabrizio 216763776 2005-03-09 08:53 root_fs_ubuntu1.7
-rw-r--r--  1 fabrizio fabrizio 15728640 2005-03-09 15:39 root_fs_ubuntu1.7.toms
-rw-r--r--  1 fabrizio fabrizio 11849024 2005-03-09 15:49 uml_vml.cow
-rw-r--r--  1 fabrizio fabrizio 11849024 2005-03-09 15:49 uml_vm2.cow
fabrizio@ubuntu:~/UserModeLinux$

```

Rysunek 4: Pliki COW dwóch maszyn wirtualnych UML.

Po ich wykonaniu w bieżącym katalogu znajdziemy plik o nazwie „linux”. To jest właśnie jądro środowiska User-Mode Linux, za pomocą którego uruchomimy wirtualny system linuxowy.

Żeby UML działał jak należy, potrzebne są nam jeszcze dwa elementy układanki: główny system plików (skompresowany obraz partycji linuxowej z wszystkimi wymaganymi programami) oraz programy narzędziowe UML. Różne obrazy partycji znajdziemy pod adresem [6]. Narzędzia UML pobieramy spod adresu [7], a następnie instalujemy je w następujący sposób:

```

$ bunzip2 uml_utilities_XXXXXXX.tar.gz
$ tar -xvf uml_utilities_XXXXXXX.tar
$ cd tools
$ make all
$ make install DESTDIR=/

```

Teraz dysponujemy już zasobem zawierającym główny system plików. Program `linux` należy umieścić w miejscu, z którego możemy go uruchomić (jeśli nie został przeniesiony, wciąż znajduje się w katalogu `linux-2.4.27`). Następnie wymuszamy odczytanie głównego systemu plików:

```

$ bunzip2 root_fs_toms1.7.205.bz2
$ linux ubd0=root_fs_toms1.7.205

```

Parametr `ubd0=` oznacza, że wirtualna maszyna ma potraktować wskazany plik jako główny system plików.

Jeśli wszystko działa właściwie, powinniśmy ujrzeć proces uruchamiania się wirtualnego systemu (Rysunek 3). Po uruchomieniu można się do niego zalogować, podając nazwę `root` i hasło `root`.

Wspólny główny system plików

Na bazie tego samego głównego systemu plików można uruchomić dwie lub więcej maszyn wirtualnych. Sterownik `ubd0` obsługuje funkcję Copy-On-Write (COW), w wyniku której główny system plików jest otwierany w trybie tylko do odczytu, a wszelkie zmiany zapisywane są w osobnym, prywatnym pliku otwieranym w trybie do odczytu i zapisu (plik COW). Na przykład, jeśli chcemy uruchomić dwie maszyny wirtualne (VM1 i VM2) korzystające z tego samego systemu plików, musimy otworzyć dwa terminale i wprowadzić w nich następujące polecenia:

```

[xterm 1]$ linux ubd0=uml_vm1.cow,
root_fs_toms1.7.205
[xterm 2]$ linux ubd0=uml_vm2.cow,
root_fs_toms1.7.205

```

Wszystkie modyfikacje wprowadzane przez te dwa wirtualne systemy będą zapisy-

```

root@ubuntu:~/UserModeLinux$ # linux ubd0=root_fs_toms1.7.205 etno
netterap.tap0, Fe:fd:0:0:1,102,168,0,254
login: root
Password:
link 2 up ifconfig
eth0
Link encap:Ethernet, HWaddr 08:00:A7:0C:00:01
inet addr: 192.168.0.1 Bcast:192.168.0.254 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:674 errors:0 dropped:0 overruns:0 frame:0
TX packets:1161 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:185470 (180.1 KiB) TX bytes:122221 (119.4 KiB)
Interrupt:16 Base address:0x0000

ln
Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:144 errors:0 dropped:0 overruns:0 frame:0
TX packets:144 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:8388 (8.1 KiB) TX bytes:8388 (8.1 KiB)

```

Rysunek 5: Wirtualna maszyna UML ze skonfigurowaną siecią.

wane we wskazanych w tych poleceniach plikach COW. Tak naprawdę system plików nie jest więc współdzielony, ale na pewno zachowana jest pełna rozdzielność obu maszyn wirtualnych. Po utworzeniu plików COW trzeba pamiętać, żeby nie uruchamiać wirtualnego Linuxa z bezpośrednim dostępem do głównego systemu plików (`ubd0=root_fs_XXX`); w plikach COW zapisane są wielkość i znacznik czasowy głównego systemu plików, dlatego jakkolwiek modyfikacja uniemożliwi korzystanie z tych plików. Kiedy dysponujemy już plikami COW, przy następnym uruchomieniu należy użyć następującej składni:

```

[xterm 1]$ linux ubd0=uml_vm1.cow
[xterm 2]$ linux ubd0=uml_vm2.cow

```

Sieci wirtualne i sieci prawdziwe

Rozwiązanie UML umożliwia wykonywanie ciekawych eksperymentów z wirtualnymi sieciami. Po skonfigurowaniu i zainstalowaniu wirtualnego systemu UML można utworzyć połączenie sieciowe z systemem hosta albo innymi systemami wirtualnymi. Szczegółowy opis konfiguracji sieciowej w środowiskach UML znajdziemy pod adresem [8].

Podstawową rzeczą, o której trzeba wiedzieć przy konfigurowaniu sieci UML jest to, że dostępne są różne opcjonalne warstwy transportowe, zapewniające wymianę pakietów między systemem wirtualnym i prawdziwym. Niektóre typy warstwy transportowej dostępne w środowisku UML wymieniono w Tabeli 1.

Aby włączyć urządzenie sieciowe w maszynie wirtualnej, jądro uruchamiamy z parametrem o składni:

Tabela 1: Warstwy transportowe w środowisku UML

Etherap, TUN/TAP	Służy do wymiany pakietów między systemem wirtualnym a rzeczywistym.
Switch daemon	(Demon emulujący przełącznik sieciowy) Zupełnie wirtualne środowisko sieciowe umożliwiające komunikację z innymi systemami UML.
Multicast	Inny rodzaj warstwy transportowej do tworzenia wirtualnej sieci.
Slip, slirp	Warstwy stosowane głównie wtedy, gdy nie są dostępne mechanizmy TUN/ TAP, albo gdy nie mamy dostępu do konta roota hosta, co uniemożliwi skonfigurowanie ustawień sieciowych.
Pcap	Umożliwia uzyskanie dostępu do interfejsu sieciowego w trybie tylko do odczytu i może być wykorzystywana do monitorowania sieci.

`eth<n>=<warstwa transportowa>, >`
`<argumenty warstwy transportowej>`
 gdzie `<n>` oznacza rzeczywisty interfejs hosta (np. `eth0`), do którego wirtualna maszyna zostanie „podczepiona”. Z teoretycznego punktu widzenia wygląda to tak, że w wirtualnej maszynie UML istnieje urządzenie `eth0` przypisane urządzeniu `tap0` systemu hosta. Urządzenie `tap0` jest natomiast bezpośrednio połączone z jego interfejsem `eth0`.

W związku z tym polecenie:

```
linux ubd0=root_fs_slac >
k8.1 eth0=ethertap,tap0, >
fe:fd:0:0:0:1,192.168.0.254
```

powoduje skonfigurowanie interfejsu `eth0` w maszynie wirtualnej, z przypisaniem własnego adresu IP. W prostej konfiguracji adres rzeczywistego interfejsu `tap0` i wirtualne-

go `eth0` może być taki sam (o bardziej złożonych konfiguracjach sieciowych można przeczytać pod adresem [8]).

Następnie konfigurujemy interfejs w maszynie wirtualnej (`/etc/hosts>`, `/etc/resolv.conf`, `/etc/network` itd.), po czym powinniśmy już dysponować pełnym dostępem do Internetu w środowisku UML.

Podsumowanie

Oprogramowanie User Mode Linux umożliwia szybkie i wygodne tworzenie wirtualnych systemów linuksowych. Może służyć do planowania, modelowania, testowania i rozwiązywania problemów z systemami linuksowymi. UML jest wykorzystywany jako baza dla innych projektów i eksperymentów, jak również do udostępniania aplikacji biznesowych oraz świadczenia usług spersonalizowanego hostingu. UML nie jest może łatwy w instalacji i konfiguracji, ale kiedy już uda nam się go uruchomić, znajdziemy dla niego bardzo wiele zastosowań. ■

INFO

- [1] Strona projektu User-Mode Linux: <http://user-mode-linux.sourceforge.net>
- [2] UML na hostach 2G/2G: <http://user-mode-linux.sourceforge.net/UserModeLinux-HOWTO-4.html#2G-2G>
- [3] Oficjalne jądro Linuksa 2.4.27: <http://ftp.ca.kernel.org/linux/kernel/v2.4/linux-2.4.27.tar.bz2>
- [4] Poprawka UML dla jądra 2.4.27: <http://prdownloads.sourceforge.net/user-mode-linux/uml-patch-2.4.27-1.bz2>
- [5] Zasoby UML: <http://user-mode-linux.sourceforge.net/dl-sf.html>
- [6] Główne systemy plików do pobrania: <http://user-mode-linux.sourceforge.net/dl-jails-sf.html>
- [7] Narzędzia UML: http://prdownloads.sourceforge.net/user-mode-linux/uml_utilities_20040406.tar.bz2
- [8] Konfiguracja sieci UML: <http://user-mode-linux.sourceforge.net/networking.html>
- [9] Kompilacja jądra: <http://user-mode-linux.sourceforge.net/compile.html>
- [10] Debugowanie jądra: <http://user-mode-linux.sourceforge.net/debugging.html>
- [11] Sesja debugowania UML: <http://user-mode-linux.sourceforge.net/debug-session.html>
- [12] Strona projektu QEMU: <http://fabrice.bellard.free.fr/qemu/>
- [13] Strona projektu Bochs: <http://bochs.sourceforge.net>
- [14] Artykuł LM o QEMU: http://www.linux-magazine.com/issue/52/QEMU_System_Emulation.pdf
- [15] SELinux i UML: <http://www.golden-gryphon.com/software/security/selinux-uml.xhtml>

AUTOR

Fabrizio Ciacchi (<http://fabrizio.ciacchi.it>, fabrizio@ciacchi.it) jest Włochem i studiuje informatykę na Uniwersytecie w Pizie. Zajmuje się poznawaniem systemu Linux, tworzeniem stron WWW w języku PHP oraz programowaniem w Javie. Jest także konsultantem pracującym dla kilku firm oraz autorem tekstów na temat systemu Linux.



ZAMÓW RENUMERATĘ

SPECJALNA OFERTA DLA PRENUMERATORÓW LINUX MAGAZINE

Proponujemy Państwu zakup popularnych produktów linuksowych W SPECJALNYCH CENACH

Szczegóły pod adresem: www.linuxmagazine.pl/subs

Infolinia: 0801 800 105