



Piotr Kamiński, Wojciech Ciesielski  
*Akademia im. Jana Długosza*

## WIRTUALIZACJA SERWERÓW I STACJI ROBOCZYCH- NA PRZYKŁADZIE WYBRANEGO OPROGRAMOWA- NIA - WIRTUALNE ŚRODOWISKO LABORATORYJNE DLA STUDENTÓW

### Streszczenie

W artykule przedstawiono zagadnienia związane z wirtualizacją oraz oprogramowanie służące do wirtualizacji systemów operacyjnych oraz praktyczne zastosowanie wirtualnej maszyny na zajęciach z przedmiotów dotyczących administrowania serwerami.

*Słowa kluczowe:* wirtualizacja, maszyna wirtualna, system operacyjny, laboratorium, ćwiczenia, administracja

### Wstęp

Wirtualizacja jest szerokim pojęciem odnoszącym się do abstrakcji zasobów w różnych aspektach komputeryzacji. Wirtualizacja umożliwia efektywniejsze wykorzystanie istniejących zasobów sprzętowych środowiska informatycznego poprzez dowolne (w ramach możliwości sprzętowych czy programowych oraz założeń projektowych) modyfikowanie cech wirtualizowanych zasobów dostosowując je do wymagań użytkownika. Jedną z pierwszych szeroko stosowanych technik wirtualizacji była technika wirtualizacji pamięci operacyjnej. Obecnie technika ta jest powszechna nie tylko w superkomputerach, serwerach czy komputerach osobistych, ale także w systemach wbudowanych. Inną techniką wirtualizacji jest wirtualizacja sprzętu komputerowego w celu umożliwienia jednoczesnego uruchamiania wielu systemów operacyjnych [1].

## Metody wirtualizacji

W obrębie pojęcia "wirtualizacja" można wyróżnić trzy grupy rozwiązań:

- **Emulacja API** – rozwiązanie programowe wykorzystujące sposób działania aplikacji jako rozdzielnych procesów w stosunku do systemu operacyjnego. Aplikacje wykorzystują interfejs API w celu dokonania procesu komunikacji z systemem operacyjnym. Emulatory API wprowadzają do głównego systemu operacyjnego otoczenie API pochodzące z innego systemu i niezbędne dla danej, emulowanej aplikacji. Tak działa Wine będący implementacją WinAPI dla środowiska systemu Linux. Rozwiązanie takie eliminuje konieczność posiadania całego systemu operacyjnego, pod którym działa emulowany program. Wymaga ono tylko emulacji jego interfejsu API.
- **Emulacja pełna** – stanowi sposób na uruchamianie aplikacji pochodzących z niekompatybilnego komputera, w stosunku do wykorzystywanego (np. PC/Mac). Emulowane są podstawowe podzespoły komputera (CPU, RAM, HDD, CD itp.) wraz z systemem operacyjnym (virtual OS), zapewniając dużą przenośność przy pewnym spadku wydajności. Praktycznie każda operacja na uruchomionym wirtualnym systemie operacyjnym jest emulowana. Emulatory pełne są zatem wirtualnymi odpowiednikami całego komputera. Typowy emulator wykonuje w pętli wszystko to, co robiłby rzeczywisty procesor maszyny emulowanej, co prowadzi do spadku wydajności pracy komputera.
- **Wirtualizacja** – pozwala jednocześnie uruchomić wiele systemów operacyjnych na tej samej platformie sprzętowej i systemowej przy maksymalnej możliwej wydajności. Wirtualizacja jest połączeniem podstawowych zalet emulacji pełnej oraz emulacji API. Opiera się ona na uruchamianiu w maszynie wirtualnej systemu operacyjnego, rezygnując z uniwersalności emulowania wielu architektur komputerów. Ograniczenie się wyłącznie do wykorzystywanej platformy sprzętowej umożliwia wykonywanie pewnej ilości procesów systemu operacyjnego gościa (systemu emulowanego) bezpośrednio na zasobach sprzętowych komputera. Dopiero w sytuacji, gdy operacje takie nie dadzą się bezpośrednio wykonać wirtualizator emuluje je. Oznacza to, że wirtualizator uruchamia system operacyjny tak, aby mógł on koegzystować z systemem głównym i osiągać maksymalną zgodność i wydajność [2].

## Typy wirtualizacji

- **Parawirtualizacja** – technika wirtualizacji, w której wirtualizowany system operacyjny (gość – ang. guest, partycja – ang. partition lub domena – ang. domain) współpracuje ze środowiskiem operacyjnym komputera w zakresie obsługi tych elementów sprzętowych, których obsługa kolidowałaby z działalnością innych środowisk wirtualizowanych.
- **Pełna wirtualizacja** – technika wirtualizacji, w której wirtualizowany system operacyjny (gość – ang. guest) ma wrażenie, że działa na prawdziwym, fizycznym sprzęcie (komputerze). W rzeczywistości odwołania wirtualizowanego systemu operacyjnego (gościa) do tych elementów fizycznych komputera, które kolidowałyby z działalnością innych środowisk wirtualizowanych lub systemu operacyjnego gospodarza (ang. host), są przechwytywane przez oprogramowanie wirtualizacyjne a następnie emulowane. Emulacja taka spowalnia pracę wirtualizowanego środowiska, dlatego pożądane jest sprzętowe wspomaganie wirtualizacji.

## Maszyna wirtualna

Jest to szczególne środowisko, które jest tworzone dla systemu gościa, za pomocą służącego do tego programu. Innymi słowy, można uruchomić system operacyjny gościa w wirtualnej maszynie. Uruchomiony system pojawi się w oknie na pulpicie komputera, ale w zależności od różnych interfejsów może być wyświetlany w trybie pełnoekranowym lub zdalnie na innym komputerze [3].

## Zastosowanie

Techniki i funkcje, które są przydatne obejmują kilka scenariuszy:

- Bezpieczny sposób na wypróbowanie oraz testowanie nowego oprogramowania. Firmy mogą dostarczyć wcześniej skonfigurowany system operacyjny wraz z zainstalowaną aplikacją dla potencjalnego odbiorcy który może bez kłopotu zapoznać się z programem.
- Instalacja i testowanie systemów operacyjnych bez konieczności restartowania komputera. Tworzenie maszyn z systemami 32 i 64 bitowymi z rodziny Windows i Linux.
- Uruchamianie starszych aplikacji jest prostsze niż tryb „Windows XP mode”. Możliwość przekonwertowania istniejącego systemu operacyjnego na maszynę wirtualną.

- Oszczędność energii elektrycznej w przypadku wirtualizacji serwerów.

## **Oprogramowanie**

### **VMware Player**

VMware Player to darmowy program przeznaczony do tworzenia, oraz uruchamiania wirtualnych maszyn z 32- lub 64-bitowym systemem operacyjnym. Pozwala na uruchomienie ponad 200 obsługiwanych systemów operacyjnych m.in. różnych wersji Windowsa i dystrybucji Linuksa. Pozwala to na przetestowanie wybranego systemu operacyjnego lub nowego oprogramowania na komputerze w specjalnie wyizolowanym środowisku, bez ingerencji w systemie macierzystym i utraty jakichkolwiek danych.

Wirtualne systemy uruchomione w VMware Playerze mogą korzystać z napędów CD/DVD, interfejsów sieciowych, urządzeń podłączanych przez USB. Dużym ułatwieniem podczas pracy jest wzajemny dostęp do udziałów sieciowych komputera rzeczywistego i wirtualnego, a także przenoszenie plików pomiędzy systemami [4].

### **Virtualbox**

VirtualBox jest narzędziem do obsługi wirtualnych maszyn, dostępnym bezpłatnie. Dostępny jest także w wersji dla przedsiębiorstw jest to wersja płatna i licencjonowana na tradycyjnych zasadach. Wersja darmowa, pozbawiona jest pewnych funkcji określanych jako przydatne głównie przedsiębiorstwom. Aplikacja współpracuje zarówno z 32, jak i 64-bitowymi systemami Windows i Linux [5].

### **Wirtualne środowisko laboratoryjne**

Do utworzenia wirtualnego środowiska w którym studenci będą mogli testować konfigurację sieciową wykorzystano komputer klasy PC o następujących parametrach:

- Dysk twardy Seagate 160GB ST3160215AS
- 1 GB RAM,
- Processor Intel Core 2 Duo E4500 @ 2,20GHz 2,21 GHz,
- Karta graficzna Nvidia GeForce 6600 256MB,

- *System operacyjny Ubuntu 11.04PL 32bit „Niebiańska Nimfa”.*

Do utworzenia wirtualnego laboratorium wymagana jest instalacja oprogramowania VMware player w środowisku Linux. Po instalacji oprogramowania:

#### I. Konfiguracja VMware player 4

1. Z menu należy wybrać „utwórz nową wirtualną maszynę”. Pojawi się okno „czarodzieja”, dzięki któremu skonfigurujemy wirtualną maszynę dla naszych potrzeb.
2. Decydujemy, czy instalujemy system operacyjny, czy oprogramowanie ma przygotować czysty dysk pod instalację którą wykonamy później – wybieramy „zainstaluj system operacyjny później”. Wybieramy system operacyjny gościa – Linux wersja Debian.
3. Podajemy nazwę oraz wskazujemy dysk na którym będzie przechowywany wirtualny host w postaci pliku.
4. Określamy pojemność dysku, Maximum disk size (GB) wpisujemy 4 GB, zaznaczamy opcje „Store virtual disk as single file”.
5. Po skonfigurowaniu wirtualnej maszyny, pojawi się okno podsumowujące nasz wybór. Z tego miejsca mamy możliwość „Dostosować sprzęt” wybieramy tę opcję. Pojawi się okienko w którym możemy skonfigurować z jakich podzespołów ma się składać nasz wirtualny komputer.
6. Po zakończeniu konfiguracji klikamy „Finish”, a nasza maszyna zostaje utworzona.

#### II. Pobieramy nośnik instalacyjny z systemem operacyjnym

1. Aby pobrać system operacyjny Debian 6 należy wejść na stronę <http://www.debian.org/distrib/netinst>. Pod tym adresem znajduje się dystrybucja systemu Debian w wersji instalowanej przez sieć. Wybieramy wersję i386.

#### III. Instalujemy wirtualny system operacyjny

1. Wybieramy utworzoną wcześniej wirtualną maszynę i klikamy „Edit virtual machine settings”, wybieramy „cd/dvd”. Wybieramy „Use ISO image file” i wskazujemy ścieżkę do wcześniej pobranego obrazu ISO z systemem operacyjnym. Uruchamiamy wirtualną maszynę i konfigurujemy system operacyjny podczas instalacji.

Po pomyślnej instalacji nasze środowisko laboratoryjne jest gotowe, tak utworzoną wirtualną możemy skopiować i przenieść na inne komputery znajdujące się w pracowni komputerowej z zainstalowanym oprogramowaniem VMware player. System operacyjny Debian 6 zainstalowano celowo bez nakładki

graficznej, tak aby student miał możliwość przećwiczenia wykonywania poleceń w trybie tekstowym oraz manualnym, edytując odpowiednie pliki.

## Przykładowe ćwiczenia

### Ćwiczenie nr 1. - Aptitude

Sprawdzić czy są skonfigurowane mirrory debiana:

- plik **/etc/apt/sources.list**.

Sprawdzamy czy system uaktualniania i instalacja z sieci działa poprawnie.

- **aptitude update** (wykonać) – sprawdzić czy system łączy się z zdefiniowanymi w **/etc/apt/sources.list** mirrorami
- **aptitude safe-upgrade** (wykonać upgrade systemu)
- **aptitude install** – zapoznać się
- **aptitude search** – zapoznać się
- <http://packages.debian.org> – zapoznać się

Za pomocą **aptitude search** spróbować znaleźć nazwę pakietu zawierającego program midnight commander. Spróbować zrobić to samo za pomocą wyszukiwarki [packages.debian.org](http://packages.debian.org). Dlaczego **aptitude search** tego nie znajdowało? Znaleźć w sieci, jak „zmusić” **aptitude**, aby to zrobiło. Doinstalować za pomocą **aptitude install** midnight commander. Obejrzeć nakładkę **aptitude**. Za pomocą nakładki **aptitude** doinstalować edytor joe. W ustawieniach midnight commandera przestawić defaultowy edytor na wbudowany. Doinstalować program stronicujący less.

### Ćwiczenie nr 2. – Konta użytkowników

Najistotniejsze polecenia i pliki przydatne do wykonania ćwiczenia:

- polecenia: **useradd, groupadd, usermod, chmod, chown, mount**
- pliki: **/etc/passwd, /etc/shadow, /etc/group, /etc/gshadow /etc/fstab**

Stworzyć 3 użytkowników:

- z ograniczoną czasowo ważnością konta.
- z nieograniczoną czasowo ważnością konta.
- startującego z midnight commander jako powłoką.

Stworzyć nowy, mały 100MB wirtualny dysk, zamontować go ręcznie (mount) go jako /home2 (system plików na partycji reiserfs lub ext4).

Skonfigurować – na wzór /home- montowanie automatyczne /home2 przy uruchamianiu systemu - plik **/etc/fstab**) – wykonać reboot systemu i poleceniem **df** sprawdzić poprawność konfiguracji.

Stworzyć CAŁKOWICIE „ręcznie” (**wyłącznie edytując właściwe pliki!!!**) 4. użytkownika należącego do nowej grupy (też stworzyć grupę „ręcznie”), mającego /home2 jako katalog domowy , shell-a csh jako defaultową powłokę. Przetestować pracę interakcyjną na tym koncie.

Założyć 2 nowe grupy. Przenieść do nich odpowiednio 1. i 2. użytkownika. Założyć hasło na dostęp do tych grup.

Dopisać 4. użytkownika do obu tych grup.

### Ćwiczenie nr 3. – Instalacja nakładki „Webmin”

Zainstalować oprogramowanie webmin. Wykonać procedurę z strony instalacyjnej webmina z wykorzystaniem apt-get i pliku /etc/apt/sources.list. Instalacja musi być wykonana ze sprawdzeniem czy pakiet jest podpisany poprawnym kluczem gpg opiekuna webmina.

Po uruchomieniu webmina (i czasowym przejściu na administrowanie systemem z przeglądarki ubuntu) :

- ustawić polską wersję językową w webminie
- zapoznać się z możliwościami konfiguracyjnymi jakie daje nam webmin
- zobaczyć jak wyglądają logi webmina
- sprawdzić czy założeni użytkownicy i grupy są zgodni z wymaganiami danymi na poprzednich ćwiczeniach. Jeżeli nie, to poprawić za pomocą webmina.
- sprawdzić w webminie, jakie usługi są włączone na naszym serwerze
- odblokować na poziomie TCP-wrappera wejście na serwer z całego świata na port serwera ftp sprawdzić czy odpowiedni wpis faktycznie pojawił się w /etc/hosts.allow
- udostępnić webmina na porcie 6091 tylko dla dostępu z własnego ubuntu.

**Bibliografia**

- [1] <http://pl.wikipedia.org/wiki/Wirtualizacja>
- [2] <http://students.mimuw.edu.pl/SO/Projekt04-05/temat3-g5/index.html>
- [3] <http://www.virtualbox.org/manual/ch01.html>
- [4] <http://www.dobreprogramy.pl/VMware-Player,Program,Windows,13312.html>
- [5] <http://www.dobreprogramy.pl/VirtualBox,Program,Windows,13122.html>

Piotr Kamiński, Wojciech Ciesielski  
*Akademia im. Jana Długosza w Częstochowie*

**VIRTUALIZATION SERVERS AND WORKSTATIONS ON THE EXAMPLE OF SELECTED SOFTWARE - VIRTUAL LABORATORY ENVIRONMENT FOR STUDENTS**

Summary

This article presents issues related to virtualization and virtualization software for operating systems and practical application of virtual machines in the class of objects relating to the administration of servers.

*Keywords:* virtualization, virtual machine, operating system, laboratory exercises, administration