

Rozdział 1.

Windows XP Professional od środka

W tym rozdziale:

- Architektura systemu Windows XP: tryb jądra i tryb użytkowy
- Pamięć chroniona i jądro
- Usługi rezydentne
- Wielozadaniowość i wieloprocusowość równoległa
- Co to jest DirectX?

System operacyjny Windows XP Professional — najnowsza wersja z rodziny systemów operacyjnych Windows — to rzeczywiście zupełnie nowe zjawisko. Windows XP Professional to technologia budowana i szlifowana od czasu, kiedy to wiele lat temu po raz pierwszy pojawił się na rynku Windows NT. W dzisiejszych czasach mamy środowiska wielozadaniowe dla stacji roboczych, które pozwalają na zwiększenie wydajności. W nowym systemie Windows XP nowy jest nie tylko elegancki pulpit, ale również obsługa wielu urządzeń i zgodność z kodem źródłowym zarówno systemów *Windows 9x*, jak i Windows NT/2000. Dzięki temu nowy system pozwala na uruchamianie tych samych aplikacji zarówno w biurze, jak i w domu. Dzięki nowym narzędziom — *Przywracanie systemu* (z ang. *System Restore*) i *Przywróć sterownik* (z ang. *Device Driver Rollback*) — pozbędziesz się również bólu głowy, który miałeś podczas instalacji nowych urządzeń i oprogramowania.

W tym rozdziale zaczniemy od podstaw systemu Windows XP, czyli w jaki sposób zapewnia on tak duży obszar pamięci pozwalając jednocześnie uruchamiać wiele aplikacji. Jest to podstawowa zaleta większości nowoczesnych systemów operacyjnych, pracujących na komputerach mainframe. Teraz możesz mieć to wszystko na stacji roboczej w biurze, a nawet na komputerze domowym.

Podstawowe elementy systemu Windows XP, o których przeczytasz w tym rozdziale, to:

- Pamięć chroniona — system Windows XP wykorzystuje możliwość pracy jednostki centralnej w dwóch trybach: *trybie jądra* i *trybie użytkownika*. Większość funkcji systemu operacyjnego jest zdefiniowana w trybie jądra i jest to część systemu

operacyjnego odpowiedzialna za zarządzanie dostępem do pamięci fizycznej komputera. Aplikacje użytkowe zgłaszają żądania do systemu operacyjnego o przydzielenie dostępu do pamięci wirtualnej. Ponieważ jądro systemu chroni dostęp do pamięci fizycznej, zmniejsza to prawdopodobieństwo, że aplikacja uda się spowodować awarię systemu operacyjnego!

- **Pamięć wirtualna** — system Windows XP obsługuje 4 GB pamięci wirtualnej, korzysta z pliku stronicowania, w którym przechowuje fragmenty obszarów pamięci, które w danej chwili nie są wykorzystywane przez żadną aplikację ani system operacyjny. Ogromna przestrzeń adresowa nie jest dzielona pomiędzy różne aplikacje. Każda aplikacja otrzymuje swoją własną wirtualną przestrzeń adresów o rozmiarze 4 GB, dzięki czemu bardziej wyrafinowane aplikacje do obróbki dźwięku i obrazu mogły rozwinąć skrzydła. Wielbiciele gier szybko zauważają, że system operacyjny Windows XP działa w zakresie obsługi multimediiów równie dobrze, jeśli nie lepiej, jak systemy operacyjne Windows 9x.
- **Wielozadaniowość** — zdolność systemu operacyjnego do uruchomienia równocześnie kilku aplikacji. Dla potrzeb biurowych nie ma nic lepszego niż możliwość pracowania na kilku aplikacjach jednocześnie, przełączania się pomiędzy nimi, wymiany danych pomiędzy nimi metodą „wytnij i wklej”.
- **Wieloprocessowość równoległa** — Windows XP potrafi obsługiwać systemy wieloprocessorowe, co pozwala w znacznym stopniu skrócić czas odpowiedzi systemu na zgłoszenie. Do tej pory wysoko specjalizowane graficzne stacje robocze były wyposażone w systemy operacyjne typu Unix lub we własne systemy operacyjne, od teraz system Windows XP zapewnia takie same możliwości po znacznie niższej cenie.
- **Warstwa uniezależnienia od sprzętu (HAL)** — HAL to warstwa kodu, który znajduje się pomiędzy systemem operacyjnym a fizycznymi urządzeniami zainstalowanymi w komputerze, takimi jak karty sieciowe czy dyski twarde; umożliwia systemowi operacyjnemu zarządzanie tymi urządzeniami.
- **Technologia DirectX** — najnowsza wersja pakietu DirectX pozwala na ominięcie ograniczeń narzuconych przez HAL w zakresie obsługi urządzeń, dzięki czemu aplikacje multimedialne mogą znacznie lepiej i szybciej współpracować z podległymi urządzeniami.

W tym rozdziale przyjrzymy się architekturze Windows XP i — związanych z nią — zalecom systemu.

Architektura systemu Windows XP: tryb jądra i tryb użytkowy

Windows XP jest zbudowany na podstawie takiej samej architektury jak systemy Windows NT i Windows 2000. Jest to system operacyjny bazujący na pamięci wirtualnej pracujący w dwóch podstawowych trybach. *Tryb jądra* (z ang. *kernel mode*) jest to warstwa kodu

systemu operacyjnego odpowiedzialna za takie podstawowe segmenty systemu operacyjnego, jak pamięć wirtualna oraz kolejkovanie działania aplikacji. Kod systemu operacyjnego uruchamiany (wykonywany) w trybie jądra ma bezpośredni dostęp do zasobów sprzętowych i pamięci komputera. Czasem można spotkać określenie tego trybu jako *tryb uprzywilejowany* (z ang. *privileged mode*). Jako że Windows XP Executive Services działają w trybie jądra, ochraniają one pamięć systemową przed dostępem przez aplikacje próbujące zapisywać w zarezerwowanych obszarach pamięci, co mogłoby doprowadzić do błędnego działania.

Uwaga

Usługi rezydentne (z ang. *Executive Services*) działają w trybie jądra i mogą wykonywać działania niedozwolone dla aplikacji użytkowych. Na przykład, jeśli aplikacja próbuje odczytać dane z dysku, nie sięga ona bezpośrednio do dysku fizycznego. Zamiast tego używa elementu zarządzania operacjami wejścia-wyjścia (z ang. *input/output*) usług rezydentnych, który wykonuje zadanie zlecone przez aplikację.

Tryb użytkownika (z ang. *user mode*) to obszar, w którym działają programy użytkowe, zarządzany przez elementy jądra. Na przykład jądro systemu jest odpowiedzialne za przydział aplikacjom potrzebnej pamięci. Aplikacje wystawiają żądania przydziału pamięci do jądra systemu poprzez wywoływanie funkcji systemowych. Dzięki temu kod trybu jądra systemu Windows XP zarządza pracą aplikacji i, jeśli to konieczne, może przerwać jej działanie.

Tryb jądra a warstwa uniezależnienia od sprzętu

Jądro systemu odpowiada za komunikację ze sprzętem w imieniu aplikacji. Element jądra systemu, zwany *warstwą uniezależnienia od sprzętu* (z ang. *hardware abstraction layer* — *HAL*), jest odpowiedzialny za kontrolę urządzeń podłączonych do komputera. Wyjątkiem od tego są sterowniki urządzeń, które pozwalają na komunikację z systemem plików i siecią komputerową, podsystemami graficznymi *Win32K* i *GDI*, używanymi do komunikacji z monitorem i urządzeniami wejściowymi. Rysunek 1.1 przedstawia ogólny obraz podziału obowiązków podlegających różnym elementom systemu operacyjnego. Elementy systemu operacyjnego zostaną omówione szczegółowo w tym rozdziale.

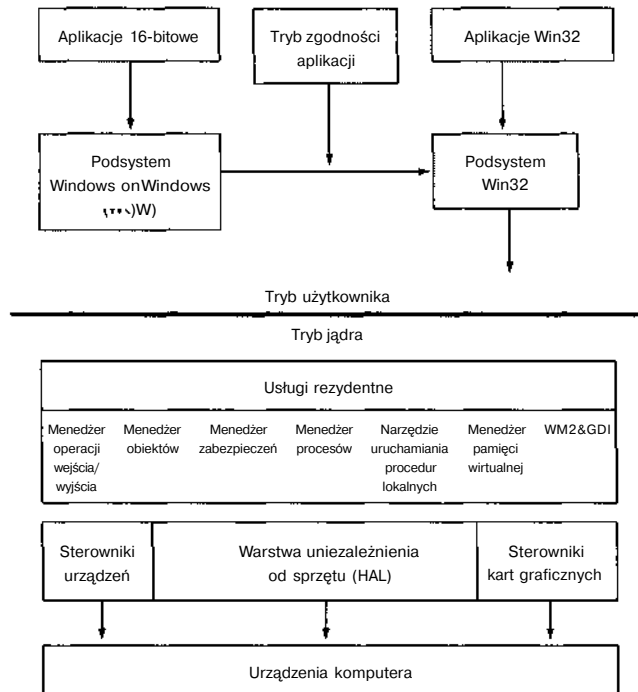
W poprzednich wersjach systemów operacyjnych firmy Microsoft aplikacje mogły bezpośrednio kontaktować się z urządzeniami podłączonymi do komputera. Z różnych powodów warstwa uniezależnienia od sprzętu (HAL) została stworzona, gdy powstał system Windows NT. Po pierwsze, system operacyjny Windows NT został stworzony do pracy na różnych platformach sprzętowych. Obejmował on zarówno obowiązujący obecnie standard procesorów *Intel*, jak i procesory typu *PowerPC*, *Alpha* i *MIPS*. Utworzenie warstwy „uniezależnienia” pomiędzy sprzętem fizycznym a aplikacjami pozwoliło na działanie systemu Windows NT na różnych platformach. Korzystanie z warstwy uniezależnienia od sprzętu pozwoliło ograniczyć spustoszenia w dostępie do urządzeń poczynione przez aplikacje, które potrafiły zablokować dostęp do konkretnego urządzenia, zmuszając do restartowania komputera.

Uwaga

Warstwa uniezależnienia od sprzętu to warstwa na poziomie oprogramowania, która pozwala systemowi operacyjnemu na współpracę z obiektami na poziomie ogólnym raczej niż na poziomie warstwy szczegółowej.

Rysunek 1.1.

System Windows XP jest zbudowany na podstawie architektury warstwowej, która chroni kluczowe elementy systemu przed bezpośrednim dostępem aplikacji



Na etapie projektowania systemu Windows NT 4.0 zrezygnowano z obsługi systemów działających na podstawie procesorów PowerPC i MIPS, natomiast na etapie testowania wersji Windows 2000 zrezygnowano z obsługi systemów działających na podstawie procesorów Alpha. Istotne jest natomiast, że w fazie projektowania systemu Windows XP zadbano nie tylko o obsługę 32-bitowych procesorów Intel, ale także 64-bitowych procesorów *Intel Itanium*. Warstwa niezależnienia od sprzętu jest nadal niezbędna do celów łatwiejszego rozdziału komunikacji systemowej od platformy sprzętowej.

Kolejną zaletą warstwy niezależnienia od sprzętu jest ochrona systemu przed błędnym kodem aplikacji. Warstwa niezależnienia od sprzętu jest odpowiedzialna za obsługę elementów sprzętowych podlegających systemowi operacyjnemu, a zatem kod programu uruchamiany w trybie użytkownika nie może wykonywać takiej operacji, jak zapisywanie buforów pamięci fizycznej wykorzystywanych przez urządzenia do obsługi sieci. Nawet elementy systemu operacyjnego komunikujące się ze sterownikami urządzeń są częścią jądra i dlatego aplikacje muszą korzystać z procedur systemowych przy odwołaniach do urządzeń. Rozdział pomiędzy trybem jądra a trybem użytkownika jest stworzony w celu zapewnienia stabilności systemu Windows XP i niezależnienia działania systemu od błędów wadliwie działających aplikacji.

Na rysunku 1.1 pokazano podsystem *Windows on Windows* (WOW) wraz z podsystemem trybu zgodności aplikacji (z ang. *Application Compatibility Mode*). Starsze aplikacje napisane na procesory 16-bitowe mogą pracować w trybie emulacji podsystemu WOW. Chociaż większość 32-bitowych aplikacji napisanych dla wersji systemu operacyjnego od Windows 95 do Windows 2000 powinna działać na komputerze z systemem Windows XP, zdarzają się i takie, które nie działają. Dla takich programów element systemu

zwany trybem zgodności aplikacji pozwala na emulację wcześniejszych wersji systemu operacyjnego Windows. Tryb zgodności aplikacji jest szczegółowo opisany w rozdziale 12. „Instalowanie i usuwanie aplikacji”.

Uwaga

Może się zdarzyć, że niektóre aplikacje będą działały niestabilnie w systemie Windows XP. Mogą to być programy antywirusowe, archiwizacyjne i niektóre narzędzia systemowe. Jednak producenci najlepiej sprzedających się programów zapewnią nowe wersje tych programów działających w systemie Windows XP. Sprawdź witrynę Windows Catalog (<http://www.microsoft.com/windowsxp/partners/catalog.asp>) w celu uzyskania dostępu do aplikacji zgodnych z Windows XP (dostęp do witryny można uzyskać również wybierając z menu *Start/Wszystkie programy/Windows Catalog*).

Oprócz umożliwienia działania aplikacji napisanych dla wcześniejszych wersji systemu Windows (nawet niektórych programów działających w trybie MS-DOS), 64-bitowa wersja systemu Windows XP pozwala uruchamiać aplikacje 64-bitowe. Wersja ta zawiera podsystem zwany *Windows on Windows 64 (WOW64)* umożliwiający wspólne działanie zarówno aplikacji 32-bitowych, jak i aplikacji napisanych specjalnie dla wersji 64-bitowej. Chociaż większość z nas nie potrzebuje rekordowych osiągnięć najnowocześniejszych 64-bitowych stacji roboczych, ta wersja systemu operacyjnego Windows XP będzie niezwykle użyteczna dla takich środowisk, jak systemy projektowania wspomaganego komputerowo (CAD) oraz aplikacje do obróbki obrazów 3D. Ponieważ zrezygnowano z obsługi systemów działających na podstawie 64-bitowych procesorów Alpha, 64-bitowa wersja systemu Windows XP jest przeznaczona dla procesorów Intel Itanium.

Pamięć chroniona i jądro

Ponieważ korzystanie z pamięci fizycznej jest możliwe tylko w trybie jądra systemu operacyjnego, jest ona również zwana *pamięcią chronioną* (z ang. *protected memory*). We wczesnych systemach operacyjnych, takich jak MS-DOS, źle napisany program mógł łatwo uzyskać dostęp do obszaru pamięci, w którym nie powinien niczego zapisywać. Skutki takiego działania były nieprzewidywalne, ale zwykle kończyło się awarią systemu. Przez podzielenie systemu operacyjnego na tryb użytkownika i tryb jądra, a co za tym idzie, ochronę obszarów pamięci zawierających kod chroniony, aplikacje użytkowe nie mają już bezpośredniego dostępu do obszarów pamięci, z których nie powinny korzystać.

Usługi rezydentne

Usługi rezydentne (z ang. *executive services*) to nazwa nadana zestawowi różnorodnych elementów stanowiących podstawy systemu operacyjnego. Podstawowe usługi oferowane przez jądro obejmują:

- Menedżer pamięci wirtualnej (z ang. *Virtual Memory Manager*) — element odpowiedzialny za zarządzanie wirtualną przestrzenią adresów o rozmiarze 4 GB dostępną dla każdego procesu, jak również adresowanie w pamięci fizycznej wszystkich obszarów pamięci potrzebnych w dowolnej chwili systemowi operacyjnemu lub aplikacji.

- Narzędzie uruchamiania procedur lokalnych (z ang. *Local Procedure Call Facility*) — element odpowiedzialny za komunikację pomiędzy procesami. Jest on potrzebny, gdyż każdy proces korzysta z tej samej przestrzeni adresów wirtualnych.
- Menedżer procesów (z ang. *Process Manager*) — gdy pojawia się potrzeba uruchomienia lub zakończenia procesu, ten element jest odpowiedzialny za utworzenie odpowiednich struktur pamięci oraz utworzenie wątków procesu. Od tej chwili inne elementy usług rezydentnych, takie jak menedżer pamięci wirtualnej, biorą udział w zarządzaniu działaniem procesu i dostępem do zasobów systemowych.
- Menedżer obiektów (z ang. *Object Manager*) — system Windows XP wykorzystuje koncepcję obiektu do wielu różnych celów. Uznajmy, że obiekt to jednostka podstawowa, która może być zarządzana przez usługi rezydentne. Na przykład takie zasoby systemowe, jak porty, są traktowane jako obiekty. Obiekt zapewnia łączność pomiędzy aplikacją a zasobem systemowym.
- Menedżer zabezpieczeń (z ang. *Security Reference Monitor*) — jest niezwykle istotną częścią usług rezydentnych. Menedżer zabezpieczeń to element kontrolujący wszystkie mechanizmy bezpieczeństwa istniejące w systemie operacyjnym. Obejmuje to wstępne logowanie, sprawdzanie hasła użytkownika oraz kontrolę, czy użytkownikowi można dać dostęp do zasobu takiego jak plik.

Jak działa pamięć wirtualna

Element jądra zwany *menedżerem pamięci wirtualnej* (z ang. *virtual memory manager* — *VMM*) jest odpowiedzialny za zarządzanie pamięcią fizyczną komputera oraz przydzielanie jej zasobów poszczególnym procesom. Jest to jedna z najbardziej istotnych cech nie tylko Windows XP, ale również wszystkich nowoczesnych systemów operacyjnych, jak Unix czy Open VMS. 32-bitowy system Windows XP może obsługiwać adresowanie w zakresie do 4 GB pamięci, chociaż jest mało prawdopodobne, abyś akurat w tej chwili miał taką ilość pamięci w swoim komputerze.



64-bitowa wersja systemu Windows XP, która ma działać na procesorach Intel Itanium, będzie potrafiła zaadresować obszar pamięci wirtualnej o rozmiarze 16 terabajtów. To ogromny postęp w zakresie dostępu do pamięci wirtualnej w porównaniu ze zwykłym systemem 32-bitowym Windows XP Professional. Podwojenie wartości numerycznej obsługiwanej przez procesor będzie miało ogromny wpływ na sukces systemu Windows XP. Użytkownicy oczekujący większych możliwości przetwarzania będą zadowoleni z systemu Windows XP umożliwiającego aplikacjom obsługę większych struktur danych i szybsze przetwarzanie danych niż w systemach 32-bitowych.

Z 4 GB dostępnego zakresu pamięci, 2 GB są zarezerwowane dla samego systemu operacyjnego, podczas gdy pozostałe 2 GB są przeznaczone dla aplikacji. Oznacza to, że aplikacje mogą być pisane w taki sposób, jakby miały być uruchamiane na komputerach z zainstalowaną pamięcią o rozmiarze 2 GB. Jednak należy wziąć pod uwagę jedną bardzo istotną cechę pamięci wirtualnej — a mianowicie fakt, iż jest ona wirtualna. Ważne jest również, że obszar pamięci, jaką może zaadresować system Windows XP, jest znacznie większy niż aktualnie zainstalowana pamięć w danym komputerze. Dwa istotne pojęcia, na które trzeba tutaj zwrócić uwagę, to *adresy pamięci wirtualnej* (z ang. *virtual memory*

addresses) i *pamięć fizyczna* (z ang. *physical memory*). Adresy wirtualne to po prostu ponumerowane lokalizacje w pamięci adresowej dostępne dla procesu. Pamięć fizyczna to ilość pamięci zainstalowanej w komputerze użytkownika. VMM zajmuje się odwzorowywaniem adresów wirtualnych w fizycznej przestrzeni adresowej.

Zadaniem VMM jest śledzenie przestrzeni adresowej procesu i koordynacja tych adresów z adresami pamięci fizycznej komputera. Zakres adresów pamięci rozciąga się od 0000000016 do 7FFFFFFF16 (w zapisie szesnastkowym — to ulubiony typ zapisu programistów). Ponadto należy pamiętać, że każdy proces korzysta z tego samego zakresu adresów. Mogłoby się wydawać, że dostępny zakres adresów powinien być podzielony pomiędzy współdziałające procesy i każdy z nich powinien otrzymać tylko fragment dostępnego zakresu, ale tak nie jest. Każdy z procesów może korzystać z całego dostępnego zakresu adresów. VMM śledzi, jakie adresy proces wykorzystuje i robi to dla wszystkich procesów działających w systemie.

Niewątpliwie większość użytkowników zna historię pierwszego twórcy systemu Windows NT Davida Cutlera, który był również autorem systemu operacyjnego Virtual Memory System (VMS) firmy Digital Equipment Corporation, zwanym teraz OpenVMS. Dlatego nie powinno nikogo zdziwić wykorzystywanie przez system Windows XP metod obsługi pamięci wirtualnej podobnych do metod wykorzystywanych przez OpenVMS.

Stronicowanie pamięci

Sposoby zarządzania pamięcią wirtualną w systemie Windows XP mogą początkowo wydawać się skomplikowane, ale w rzeczywistości jest to prosta sztuczka pozorująca dostępności pamięci o rozmiarze 4 GB, podczas gdy fizycznie komputer ma zainstalowane dużo mniej pamięci fizycznej. Należy pamiętać o kilku bardzo istotnych rozwiązaniach używanych przy obsłudze pamięci wirtualnej na zasadzie stronicowania:

- 32-bitowe adresowanie pamięci wirtualnej,
- katalog stronic,
- tablica stronic,
- ramka stronic,
- Translation Lookaside Buffers,
- pliki stronicowania.

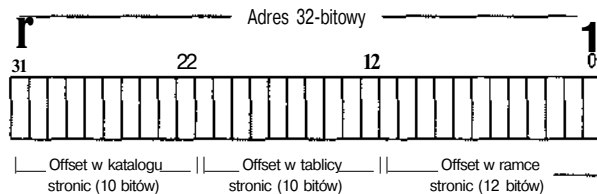
32-bitowe adresowanie pamięci wirtualnej

System Windows XP używa 32-bitowego adresu do lokalizacji danych w pamięci. Cała pamięć jest podzielona na segmenty o rozmiarze 4 096 bajtów, zwanych *stronicami* (z ang. *pages*). 32-bitowy adres składa się z trzech segmentów, każdy z nich używany jest do indeksowania tablicy, której elementy wskazują na obszary w pamięci zawierające dane potrzebne aplikacji. Należy pamiętać, że cały czas mowa o adresach pamięci wirtualnej, a nie pamięci fizycznej. Po odnalezieniu poszukiwanego bajtu przy użyciu adresu wirtualnego może się okazać, że bajt może być zapisany w pamięci fizycznej lub strona zawierająca poszukiwany bajt może znajdować się w pliku stronicowania. W takim wypadku

strona musi zostać wczytana do pamięci i dopiero wtedy dane będą dostępne dla aplikacji. Najpierw przyjrzymy się, w jaki sposób adres 32-bitowy jest wykorzystywany do odszukiwania jednego bajtu danych, a następnie jak strony pamięci są wczytywane do pamięci z pliku stronicowania.

Na rysunku 1.2 pokazano, w jaki sposób adres jest dzielony na trzy osobne segmenty, z których każdy ma swoje przeznaczenie.

Rysunek 1.2.
32-bitowy adres składa się z trzech segmentów



Rysunek przedstawia trzy segmenty adresu, które odpowiadają offsetom w *katalogu stronic* (z ang. *page directory*), *tablicy stronic* (z ang. *page table*) i *ramce stronic* (z ang. *page frame*). VMM wykonuje kilka kroków w celu zlokalizowania bieżącej ramki stronic, która jest bieżącą stroną danych aplikacji.

Katalog stronic

Pierwsze 10 bitów adresu określanych jest jako *offset w katalogu stronic* (z ang. *page directory offset*). Katalog stronic to stronica pamięci, która — jak wiadomo — ma rozmiar 4 096 bajtów. Każdy wpis w katalogu stronic ma rozmiar czterech bajtów i jest określany jako *wpis w katalogu stronic* (z ang. *page directory entry* — *PDE*). Łatwo obliczyć, że pojedynczy katalog stronic może przechowywać do 1 024 PDEs ($1\ 024 \times 4 = 4\ 096$ bajtów).

Każdy wpis jest offsetem w kolejnej strukturze zwanej *tablicą stronic* (z ang. *page table*).

Każdy proces uruchomiony w systemie operacyjnym Windows XP posiada swój własny katalog stronic. Może to się wydawać ogromnym marnotrawstwem pamięci, ale jak się zaraz przekonamy, sposób, w jaki VMM zarządza pamięcią, potrafi obsłużyć ogromną ilość katalogów pamięci. Ponieważ każdy proces ma własny katalog stronic, to wszystkie procesy mogą używać tego samego zakresu adresów. Ponieważ adresy są wirtualnymi wskaźnikami i VMM zajmuje się przepisywaniem bieżących danych pomiędzy plikiem stronicowania a pamięcią fizyczną, konkretny adres 32-bitowy wykorzystywany przez proces nie wskazuje na ten sam bajt danych co identyczny adres wykorzystywany przez inny proces. Każdy proces ma własny zbiór danych, do których odwołuje się, wykorzystując zakres adresów zarządzanych przez VMM systemu Windows XP.

Tablica stronic

Po tym, jak pierwsze 10 bitów 32-bitowego adresu zostało wykorzystanych do znalezienia wpisu w katalogu stronic, wartość odczytana z tego miejsca jest wykorzystywana do znalezienia kolejnej struktury zwanej *tablicą stronic* (z ang. *page table*). Ponownie jest to struktura stronicy o rozmiarze 4 096 bajtów składająca się z wpisów zwanych *wpisami w tablicy stronic* (z ang. *page table entries* — *PTEs*), z których każdy ma rozmiar czterech bajtów.

Kolejne 10 bitów 32-bitowego adresu jest wykorzystywanych przez VMM do znalezienia wpisu w tablicy stron wskazanej przez wpis w katalogu stron.

Po wykorzystaniu przez VMM wpisu PDE do zlokalizowania tablicy stron i wykorzystaniu drugiej 10-bitowej porcji adresu do odnalezienia wpisu w tablicy stron, VMM korzysta z wartości tam zapisanej do zlokalizowania kolejnej struktury zwanej *ramką stron* (z ang. *page frame*).

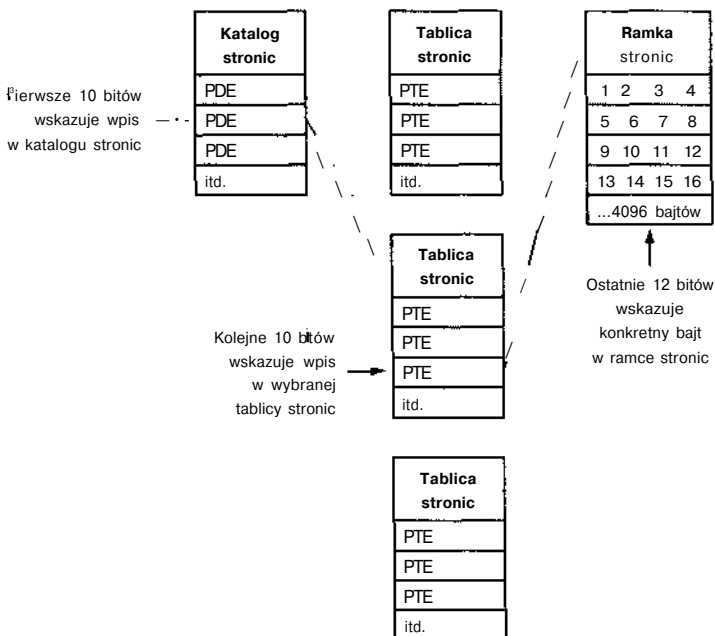
Ramka tablic

Nareszcie dotarliśmy do miejsca w pamięci, gdzie przechowywane są rzeczywiste dane. Należy pamiętać, że cały czas mówimy o pamięci wirtualnej, a nie fizycznej. Kiedy VMM znalazł w końcu ramkę stron, wykorzystuje ostatnie 12 bitów 32-bitowego adresu do wyszukania bajtu danych w ramce stron, na który wskazuje adres. Ponieważ ostatnia część adresu ma rozmiar 12 bitów, może być wykorzystana do wskazania konkretnych bajtów w ramce stron. Ostatnie 12 bitów adresu nie wskazują już na kolejną strukturę procesu stronicowania, ale są offsetem w bieżącej stronie pamięci zawierającej rzeczywiste dane użytkownika lub aplikacji. Dlatego istnieje możliwość zaadresowania każdego bajtu w ramce stron osobno. Pamiętaj, że wpisy PDE w katalogu stron (z ang. *page directory entry*) i PTE w tablicy stron (z ang. *page table entry*) mają długość 10 bitów i są używane do wskazywania czterobajtowych wpisów w kolejnej strukturze procesu stronicowania, natomiast 12-bitowy fragment adresu pozwala VMM adresować poszczególne bajty w ramce stron.

Rysunek 1.3 przedstawia proces wykorzystywania wskaźników do znalezienia wpisu w tablicy stron, ramce stron i bieżącego bajtu danych w ramce stron.

Rysunek 1.3.

Wpis w katalogu stron wskazuje tablicę stron, a wpis w tablicy stron wskazuje ramkę stron



Jak widać na rysunku 1.3, wpisy w katalogu stronnic nie wskazują na wpisy w tablicy stronnic, ale na konkretną tablicę stronnic, których każdy proces może utworzyć bardzo dużo, w zależności od przydzielonego mu obszaru pamięci. Wpis w tablicy stronnic nie wskazuje na konkretny wpis w ramce stronnic, ale na ramkę stronnic zawierającą poszukiwany wpis. Pozostałe 12 bitów 32-bitowego adresu pozwala wskazać konkretny bajt danych w ramce stronnic.

Czyż to nie jest proste? Ale poczekaj, to jeszcze nie wszystko...

Plik stronicowania

Większość komputerów nie jest wyposażona w 4 GB pamięci fizycznej. Dlatego też VMM musi stwierdzić, czy stronica zawierająca dane jest w danej chwili dostępna w pamięci, czy też jest przechowywana w pliku stronicowania na dysku. Dzieje się tak, gdy PTE wskazuje na ramkę stronnic, która w danej chwili nie jest obecna w pamięci fizycznej. W takiej sytuacji pojawia się tak zwany *błąd stronicowania* (z ang. *page-fault*). Oznacza to, że stronica pamięci, na którą wskazuje PTE, musi dopiero zostać wczytana do pamięci fizycznej. Jeśli cała dostępna pamięć jest w danej chwili zajęta danymi, to VMM musi wybrać inną stronicę pamięci, którą przepisze do pliku stronicowania, aby w to miejsce wczytać stronicę, do której wystąpiło odwołanie. W całej operacji bierze udział procesor. Po wczytaniu do pamięci strony, do której odwoływała się aplikacja, procesor kontynuuje proces stronicowania i dane zwracane są do aplikacji.

Uwaga

Zarówno rozwiązanie dzielenia 32-bitowego adresu w celu wyszukiwania danych, jak i rozwiązanie przepisywania danych pomiędzy plikiem stronicowania a pamięcią fizyczną, są rozwiązaniami łatwymi do zrozumienia. Jest jednak jeszcze jedna rzecz w całym procesie, na którą powinieneś zwrócić uwagę. Istnieje możliwość, że wszystkie bieżące stronicie znajdują się w pliku stronicowania. W takiej sytuacji pojawią się trzy błędy stronicowania, zanim uda się dotrzeć do bieżących danych: jeden przy wczytywaniu strony katalogu stronnic, drugi przy wczytywaniu tablicy stronnic i ostatni przy wczytywaniu ramki stronnic zawierającej potrzebne dane.

Co to jest Translation Lookaside Buffer?

W opisanym wcześniej procesie wyszukiwania danych w pamięci VMM korzystał z trzech stronnic pamięci, aby odszukać konkretny bajt danych. Istnieje jednak metoda, która może pomóc w szybszym odszukaniu tego bajtu. Jest to sprzętowa część samego procesora zwana *Translation Lookaside Buffer* (*TLB*). Jest to pamięć buforowa będąca częścią kości procesora, która przechowuje najczęściej używane adresy fizyczne. Należy zwrócić uwagę na to, że są to wskaźniki do adresów w pamięci fizycznej, nie wskaźniki do danych przechowywanych w pliku stronnic. Ponieważ łatwiej procesorowi przeszukać pamięć podręczną niż czekać, aż napęd dysku wczyta stronicę pamięci, można szybko odwołać się do TLB, aby ustalić, czy potrzebne dane są dostępne w pamięci fizycznej. Jeśli nie, uruchamiany jest standardowy proces stronicowania w celu załadowania danych do pamięci fizycznej. Czas potrzebny do przeszukania TLB jest minimalny w porównaniu z czasem przepisywania danych pomiędzy systemem stronnic a pamięcią fizyczną.

Baza danych ramek stronic

Ponieważ proces uruchamiany w systemie Windows XP ma własny katalog stronic, a wszystkie procesy korzystają z tego samego zakresu adresów, niezwykle istotne jest zrozumienie, że nie wszystkie adresy wirtualne są takie same. Inaczej mówiąc, adres wirtualny dla jednego procesu nie zwraca tych samych danych co identyczny adres wirtualny innego procesu. Aby sprawować kontrolę nad tym, co jest w danej chwili zapisane w pamięci fizycznej, VMM korzysta z bazy danych zwanej *bazą danych ramek stronic* (z ang. *page-frame database*), która przechowuje wpis dla każdej stronicy pamięci w systemie i korzysta z tej bazy, aby śledzić zmiany statusu tych stronic.

W tabeli 1.1 przedstawiono sześć kategorii, według których klasyfikowany jest status stronic pamięci fizycznej.

Tabela 1.1. Status stronic pamięci fizycznej

Status stronicy	Opis
Aktualny (z ang. <i>valid</i>)	Stronica jest w użyciu przez aktywny proces
Zmieniony (z ang. <i>modified</i>)	Stronica była zmieniana, ale dane nie zostały przepisane do pliku stronic
Oczekujący (z ang. <i>standby</i>)	Stronica została usunięta z aktualnego zestawu dla procesu
Wolny (z ang. <i>free</i>)	Stronica jest dostępna do zapisu po wyczyszczeniu (chyba że jest to stronica przeznaczona tylko do odczytu)
Wyczyszczony (z ang. <i>zeroed</i>)	Wolna stronica jest dostępna dla dowolnego procesu
Błędny (z ang. <i>bad</i>)	Stronica, która była przyczyną awarii sprzętowej; nie jest dostępna do użytku dla procesów

Aby zachować kontrolę nad zmianami statusu stronic pamięci fizycznej, baza danych ramek stronic jest częścią obszaru pamięci kontrolowanego przez jądro systemu zwanego *obszarem niestronicowanym* (z ang. *nonpaged pool*). Katalogi stronic, tablice stronic i stronicie danych mogą być przesyłane do pliku stronic w celu zwolnienia miejsca dla innych stronic, ale stronicie wchodzące w skład obszaru niestronicowanego zawsze są obecne w pamięci fizycznej.

Bieżąca organizacja bazy danych ramek stronic jest zestawem powiązanych list, które grupują stronicie w oparciu o ich status. Oznacza to, że jeśli VMM ma odszukać trzy kolejne wolne stronicie potrzebne dla procesu, może to zrobić korzystając z listy wolnych stronic. Jeśli VMM chce zapisać stronicę danych do pliku stronic, aby zrobić miejsce do przepisania innej stronicy, może poszukać stronic zmienionych (takich, których zawartość była modyfikowana od przepisania z pliku stronic) a następnie przerzucić takie stronicie do pliku stronic.

Innym istotnym rozwiązaniem wynikającym z bazy danych ramek stron jest wskaźnik do PTE, który odzwierciedla aktualną lokalizację stronicy w pamięci fizycznej. W ten sposób VMM może stwierdzić, który proces korzysta z danej stronicy pamięci fizycznej. Każdy program ma dostęp do ogromnej przestrzeni adresowej, w której proces może być wykonywany. W rzeczywistości to VMM zajmuje się tłumaczeniem wirtualnych adresów procesu na adresy fizycznej pamięci komputera.

Wielozadaniowość i wieloprocusowość równoległa

MS-DOS i inne proste systemy operacyjne pozwalają na uruchomienie tylko jednego programu w danej chwili. Wielozadaniowe systemy operacyjne (z ang. *multitasking operating systems*) pozwalają na jednoczesne uruchamianie wielu programów. W systemie z jedną jednostką centralną tylko jeden proces jest aktywny w danej jednostce czasu. Procesy mogą być dzielone na sekwencje instrukcji zwane *wątkami* (z ang. *thread*), które następnie mogą być uruchamiane równolegle z innymi zadaniami potrzebnymi do działania aplikacji. Każdy wątek ma przydzielony z pozoru bardzo krótki odcinek czasu mierzony w milisekundach, w którym to czasie wątek jest wykonywany. Należy jednak uzmysłowić sobie, że dzisiejsze jednostki centralne mogą wykonywać miliony operacji na sekundę, więc nawet bardzo mały fragment czasu przydzielony wątkowi danego programu jest więcej niż wystarczający do wykonania zleconego zadania.

W systemie jednoprocusorowym tylko jeden wątek może być wykonywany w danej chwili przez jednostkę centralną. Niezwykle istotna jest metoda zastosowana przy podziale czasu procesora pomiędzy wątki różnych procesów. Na przykład w systemie Windows 3.1 zastosowano *wielozadaniowość bez wywłaszczania* (z ang. *nonpreemptive multitasking*). Metoda ta polega na przydzieleniu czasu procesora dla danego zadania na czas potrzebny do całkowitego wykonania zadania, następnie zadanie oddaje zasoby procesora następnemu zadaniu oczekującemu w kolejce. Oczywiście jeśli działający program nie chce przekazać procesora następnemu programowi oczekującemu w kolejce, to komputer zawiesi się. W systemie korzystającym z wielozadaniowości z wywłaszczaniem, jakim to systemem jest Windows XP, system operacyjny decyduje o tym, w jaki sposób czas procesora jest przekazywany kolejnym zadaniom.

System operacyjny przydziela każdemu wątkowi procesu określony przedział czasu, w którym może być uruchomiony i po którym musi przekazać kontrolę nad jednostką centralną. W tym momencie system operacyjny zapisuje dane potrzebne do kontynuowania działania procesu w przyszłości. Proces przełączania pomiędzy wątkami procesów nazywa się *przełączaniem kontekstów* (z ang. *context switching*) i ma miejsce miliony razy w ciągu sekundy w dzisiejszych szybkich jednostkach centralnych.

Ponieważ działaniem procesora steruje system operacyjny, a nie aplikacja, wielozadaniowość z wywłaszczaniem pozwala uniknąć sytuacji, kiedy błąd jednej aplikacji powodował zawieszenie się całego systemu. Na przykład jeśli w Windows XP aplikacja z takiego lub innego powodu zawiesi się, zawsze można przywołać *Menedżera zadań* i „zabić” taki proces. Dzieje się tak, ponieważ Windows XP korzysta z technologii zwanej *priorytetyzacją* (z ang. *prioritization*), która polega na przydzielaniu niektórym procesom wyższego priorytetu w dostępie do jednostki centralnej. *Menedżer zadań* ma wysoki priorytet, dlatego może być przywołany bez względu na to, jakie zadanie jest w danej chwili wykonywane.

Jak widać, przez użycie koncepcji wątków podczas pisania aplikacji można przyspieszyć jej wykonywanie. Zamiast czekać na pojawienie się jakiegoś wydarzenia zewnętrznego.

proces może zaplanować, aby wątek czekał na nadejście zdarzenia i kontynuować wykonywanie innych wątków, których działanie nie jest zależne od wątku czekającego na nadejście zdarzenia. Jest to szczególnie przydatne podczas wykonywania operacji wejścia-wyjścia czy też operacji sieciowych, kiedy to działanie procesu może być kontynuowane, podczas kiedy wątek oczekuje na przesłanie do buforu danych z podsystemu wejścia-wyjścia.

W systemie wieloprocesorowym wyposażonym w więcej niż jedną jednostkę centralną rozdzielenie wątków jednego procesu pomiędzy kilka procesorów jest zadaniem bardzo prostym, przez co można uzyskać jeszcze większą szybkość pracy. Chociaż systemy wieloprocesorowe są najczęściej używane jako serwery w środowiskach sieciowych, również wysokiej jakości stacje robocze mogą być wyposażone w więcej niż jeden procesor w sytuacjach, gdy wykonywany jest bardzo dużo obliczeń.

Co to jest DirectX?

Kiedy system Windows NT po raz pierwszy pojawił się na rynku, większość urządzeń graficznych, podobnie jak innych elementów sprzętowych, takich jak pamięć, karty sieciowe itd., było obsługiwanych przez jądro systemu, a czas odpowiedzi był znacznie dłuższy niż w komputerze z systemem MS-DOS, gdzie aplikacje mogły bezpośrednio obsługiwać na przykład sterowniki karty graficznej. Aby poprawić tę niedogodność i umożliwić działanie szybkich urządzeń graficznych, do systemu operacyjnego dołączono DirectX. Windows XP obsługuje DirectX w wersji 8, który zbudowany jest z następujących elementów:

- **DirectX Graphics** — łączy w sobie elementy wcześniej znane jako *Microsoft DirectDraw* i *Microsoft Direct3D*. Ułatwia to pracę programisty przy tworzeniu niezwykłych gier i grafik, które sprawiają, że Windows XP jest wspaniałą platformą dla programów zorientowanych graficznie.
- **Direct Audio** — ten element pakietu DirectX również jest połączeniem elementów znanych wcześniej jako *Microsoft DirectSound* i *Microsoft DirectMusic* w złącze programowe aplikacji (z ang. *applications programming interface* — *API*), przez co pisanie aplikacji audio stało się łatwiejsze dla platformy Windows XP.
- **Microsoft DirectInput** — funkcja zapewniająca obsługę dużej ilości urządzeń wejściowych, obejmująca obsługę technologii *force-feedback*.
- **Microsoft DirectPlay** — wielobicielom gier internetowych ten element DirectX w wersji 8 pozwoli tworzyć wieloużytkownikowe gry działające z dużą szybkością.
- **Microsoft DirectShow** — Windows XP posiada wiele cech multimedialnych, takich jak *MorieMaker*. Ten element DirectX umożliwia systemowi Windows XP zapisywanie wysokiej jakości obrazów wideo i dźwięków audio, jak również posiada funkcje potrzebne do odtwarzania plików audio i wideo wysokiej jakości.
- **Microsoft DirectSetup** — i w końcu DirectX umożliwia łatwą instalację wszystkich elementów pakietu DirectX. Instalowanie aplikacji multimedialnych nigdy dotąd nie było takie proste.

Aktualny układ elementów pakietu DirectX jest przedmiotem prac programistów i nie należy do rzeczy, które łatwo wyjaśnić w takiej książce jak ta, przeznaczonej bardziej dla użytkowników systemu. Można znaleźć książki poświęcone programowaniu z wykorzystaniem technologii DirectX w wersji 8. Należy pamiętać, że aplikacje pisane z wykorzystaniem technologii DirectX dla systemu Windows XP w ogromnym stopniu poprawiają wrażenia użytkownika z korzystania z aplikacji multimedialnych. W dalszej części tej książki opiszemy niektóre z tych aplikacji będące częścią systemu operacyjnego.