

Wpływ architektury komputera na system operacyjny



Architektura procesora



Plan

- Podstawowe cechy współczesnych procesorów
- Klasyfikacja procesorów
 - Architektury 32 i 64 bitowe
 - RISC, CISC, VLIW i EPIC
- CISC i RISC w nowoczesnych procesorach.
- Potokowanie – wady, zalety
 - Optymalizacje, idee
- SMT, Hyper-Threading
- Mały przegląd co ciekawszych procesorów i bajerów.



Podstawowe cechy

- Do czego dążymy?
 - Zwiększenie prędkości
 - Zwiększenie współczynnika ILP (Instruction Level parallelism)
 - Superskalarność – możliwość ukończenia wielu instrukcji w jednym cyklu procesora, wiele jednostek wykonawczych (SMT, HT) Wiele ALU.
 - Potokowanie – przetwarzanie blokowe



I klasyfikacja

- Podział ze względu na rozmiar słowa maszynowego. Co od tego zależy?
(rejestry bazowe, jednostki arytmetyczne, szyna adresowa, cena, więcej tranzystorów).
 - Architektury 32 lub 64 bitowe
- 32 bitowe: Procesory Intela od 80386 aż do Pentium 4, AMD zgodne z x86-32 (Athlon), ARM-y
- 64 bitowe: IBM PowerPc 9xx, Intel Core 2, Itanium (!), Sun UltraSPARC, AMD Opteron, Athlon 64 (wsteczna kompatybilność)

Windows wsparcie

- Jak sprawdzić w Windows czy mamy wersje 64 bitową:
 - 1. Otwórz aplet System, klikając przycisk **Start** , klikając kolejno polecenia **Panel Sterowania, System i konserwacja**, a następnie klikając polecenie **System**.2.
 - W aplecie **System** można sprawdzić typ systemu.



RISC, CISC i inni

- *Podział ze względu na sposób traktowania instrukcji mikroprocesora.*
- *RISC = Reduced Instruction Set Computers*
- *CISC = Complex Instruction Set Computers*
- *Złożoność operacji (ilość cykli zegara).*
- *Inni: VLIW, EPIC*

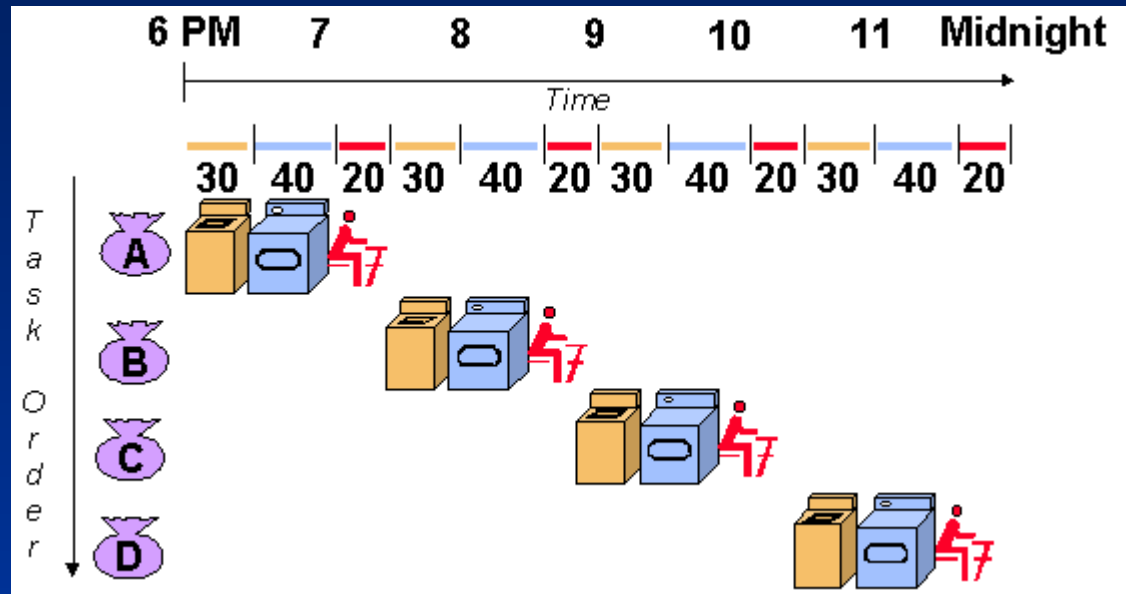


RISC

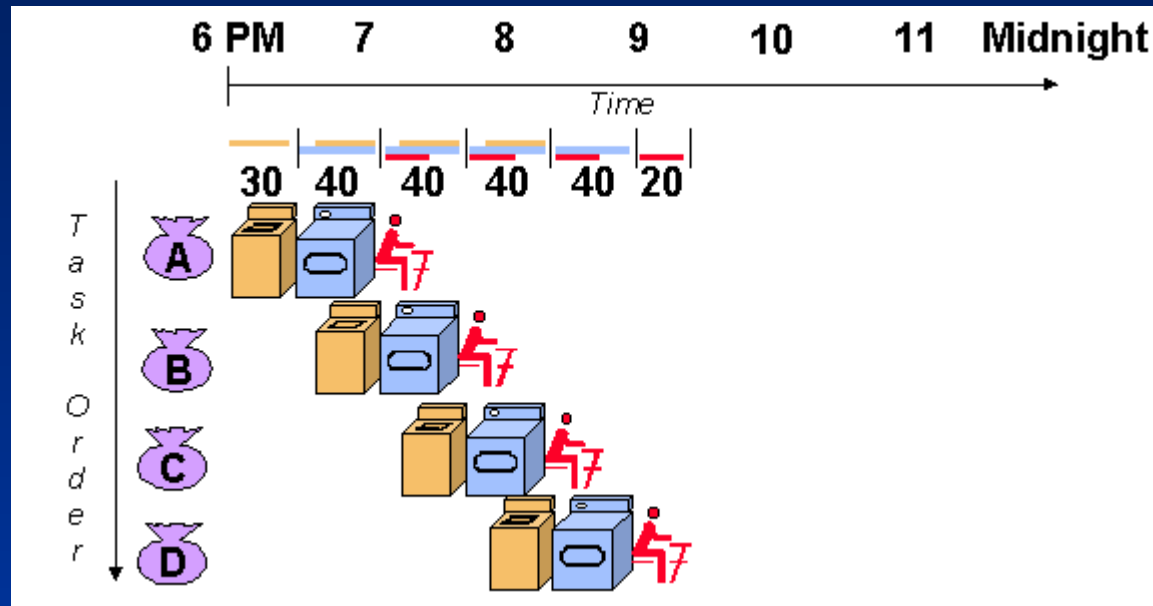
- Mała ilość instrukcji procesora, uproszczenie dekodera rozkazów.
- Instrukcje trwające jeden cykl zegara procesora, co umożliwia efektywne wykorzystanie potokowania.
- Operacje na pamięci przez mechanizm load/store
- Zwiększenie ilość rejestrów.
 - Operacje na 3 rejestrach bez zamazywania



Pipelining, przykład życiowy



Pipelining, przykład życiowy



Potokowanie

- Podział logiki przetwarzania rozkazów procesora na niezależne fazy.
- W MIPS (prekursor) podział na 5 faz:
 - Fetch, read and decode, execute/calculate, acces memory, write result.
- Inne procesory i realizacje: zmienione fazy.



Problemy z potokowaniem

- Zależności danych (hazardy)
 - Hazardy danych.
 - Hazardy strukturalne (odwołania do tych samych układów obliczeniowych ALU)
(branch,komputacja).
- Konieczność odwołania do pamięci niepodręcznej. Zastoje
- Skoki warunkowe
- Metody rozwiązywania:



Zmiana nazw rejestrów

- Dzięki zmianie nazwy R1 na R2, możemy wykonać operacje 4,5,6 równoległe z 1,2,3. Pozorna wspólnota danych.

```
1. R1 = MEM[1024]
2. R1 = R1 + 2
3. MEM[1032] = R1
4. R1 = MEM[2048]
5. R1 = R1 + 4
6. MEM[2056] = R1
```

Out-of-order Execution

- Pobieramy i dekodujemy instrukcje
- Dodajemy ją do kolejki zadań. Jeśli argumenty instrukcji są dostępne, pozwalamy na opuszczenie kolejki przed poprzednikami.
- Wykonanie w odpowiedniej jednostce wykonawczej.
- Kolejujemy wyniki
- Wpisanie wyników po wynikach poprzedników



Branch prediction

- Przewidywanie wyniku skoku warunkowego. Czasami potrzebne wycofanie wyników obliczeń (dezaktualizacja jeśli przewidywania okazały się błędne).



Branch predication

- Opatrzenie niektórych instrukcji predykatami, które powodują warunkowe wykonanie instrukcji.
 - loop CMP Ri, Rj // ustawia flagi predykatów
SUBGT Ri, Ri, Rj // jeśli GT wykonaj $i := i - j$;
SUBLT Rj, Rj, Ri // jeśli LT, $j := j - i$
BNE loop // jeśli NE, jmp loop



Z serii ciekawostek

- W procesorach ARM, technologia Jazelle DBX (Direct Bytecode Execution).
 - Do faz przetwarzania potokowego dodana jest jeszcze jedna faza, w której rozpoznawane są rozkazy bytecode'u.

Są następnie tłumaczone (sprzętowo) na instrukcje natywne ARM.

Implementacja tylko niektórych operacji JVM, ale sprzętowo rozwiązany powrót do interpretacji Software'owej.



CISC

- Zakłada odpowiedniość między operacjami języka wysokiego poziomu, a rozkazami mikroprocesora.
- Złożone operacje, wymagające wielu cykli zegara
- Operowanie bezpośrednio na pamięci
- Złożone dekodowanie rozkazów
- Jeden rozkaz mikroprocesora odpowiada wielu operacjom niskiego poziomu.



CISC przykłady

- x386.



CISC rzeczywistość

- Zamiana rozkazów zmiennej długości na miopsy, które łatwiej poddają się potokowaniu.
- Czyli tak naprawdę:
 - Programujemy stosując rozkazy CISC
 - Późniejsza translacja na mniejsze operacje
- W rzeczywistości CISC podąża drogą ku RISC.



Inni (potokowe)

- EPIC = Explicitly Parallel Instruction Computing
 - Zamiast na złożonym układem wbudowanym w procesor, odpowiedzialność za odpowiednie strumieniowanie danych do potoków przekazuje się kompilatorom.
- VLIW = Very Long Instruction Word
 - Podwaliny pod EPIC



EPIC

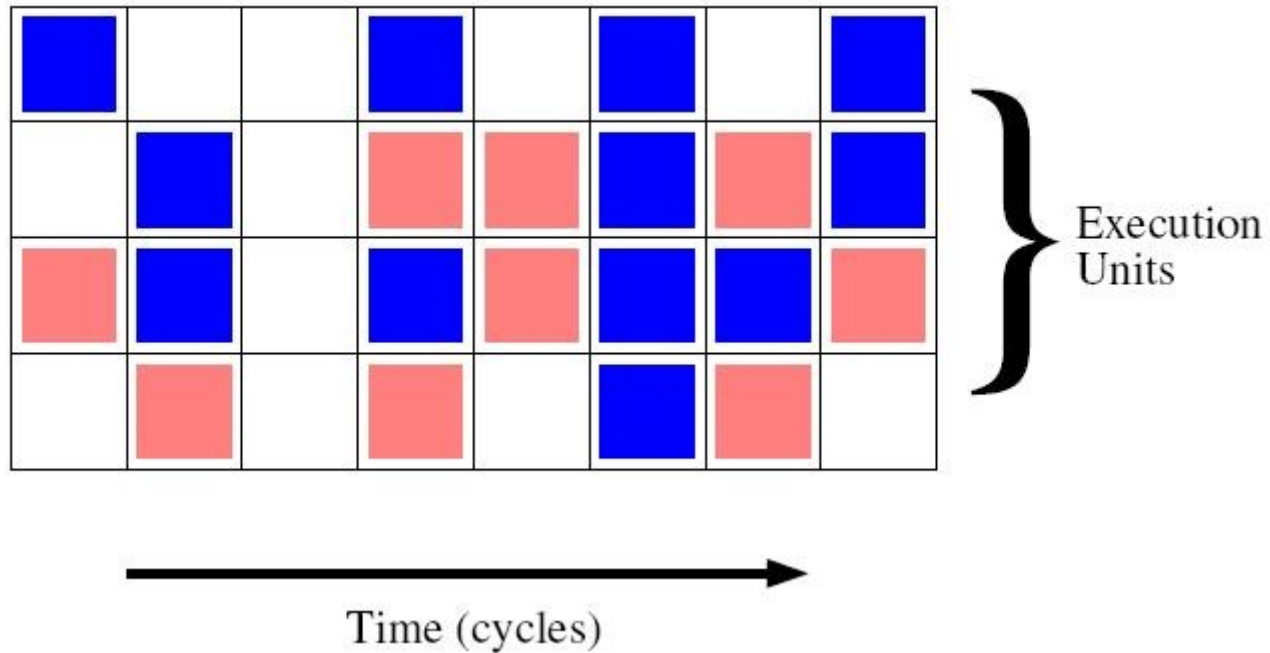
- Uproszczenie budowy procesora.
 - Usunięcie modułów odpowiedzialnych za out-of-order execution.
- Na przykład nowa serwerowa architektura Intel, Itanium 2.
- System operacyjny obsługujące (udostępniające odpowiednie kompilatory):
 - Windows Server 2008
 - Linux

SMT

- SMT, Simultaneous multithreading
- Jeśli mamy wiele rdzeni/jednostek wykonawczych inne operacje wykonują się na nich
- Motywacja – opóźnienia w dostępie do pamięci. Zależność od poziomu ILP. Chcemy wykonywać więcej równoległych instrukcji, np. w przypadku cache miss.
- Przykłady komer. implementacji: Intel, IBM



SMT



Architektura SMT dwuwątkowa

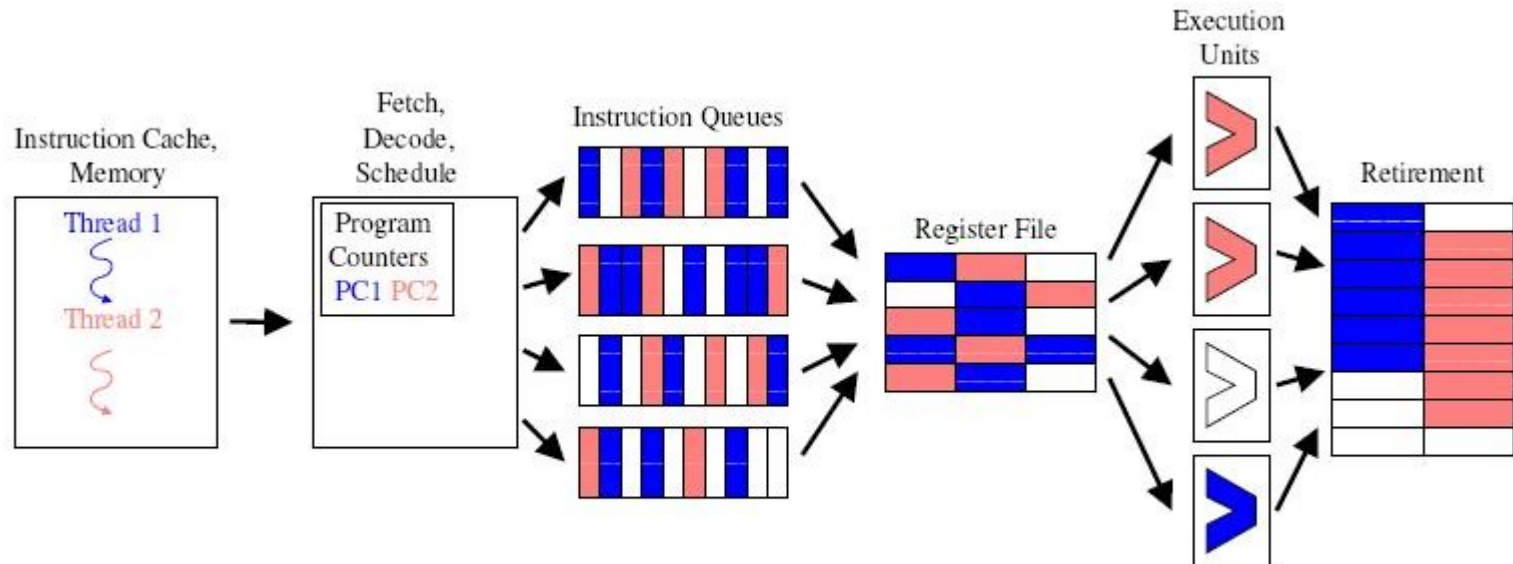


Figure 2.2: A simplified 2-thread dynamic SMT architecture.

Hyper-Threading

- Konkretna realizacja SMT
 - Procesory Pentium 4, albo Pentium EE.
- Dodanie dodatkowych jednostek do odczytywania instrukcji z różnych wątków, dodatkowe rejestry dla wątków.
- Dla systemu widoczny dodatkowy procesor logiczny
- Wzrost wydajność w granicach 20-30%.



Hyper-Threading podział zasobów

Area	Duplicated	Dynamically Shared	Tagged or Partitioned
Fetch	ITLB Streaming buffers	Microcode ROM	Trace cache
Branch prediction	Return stack buffer Branch history buffer		Global history array
Decode	State	Logic	uop queue (partitioned)
Execute	Register rename	Instruction schedulers	Retirement Reorder buffer ($\leq 50\%$ per thread)
Memory		Caches	DTLB

Table 2.1: Resource division on Hyper-Threaded P4 processors.

SMT Fajne Problemy

- Wywołania bezproduktywne
- Problem spin locków / przepowiadanie zamknięcia pętli. Rozwiązania:
 - Unikanie spin locków
 - Zmniejszenie priorytetu procesu piruetującego
// metoda zastosowana przez IBM
- Współdzielenia cache - częstsze wymiany stron w cache'u - częstsze cache missy.



SMT + OS

- Scheduler i priorytety
- Proces High priority i Low priority podzielone po równo na dwóch procesorach. Hyper-Threading nie wspiera priorytetowania procesów. Rozwiązania:
 - Software'owe: część czasu dla low.
 - Hardware'owe: W Ibm Power jest możliwość ustalenia stosunku fetchowanych instrukcji z procesów.



RISC przykład, z serii ciekawostek

- Procesory CELL opracowane przez IBM, Sony i Toshiba.
- Składają się ze zubożonego rdzenia PowerPC, wspomaganego przez 8 specjalizowanych podjednostek zmiennoprzecinkowych.
- [http://en.wikipedia.org/wiki/Cell_\(microprocessor\)](http://en.wikipedia.org/wiki/Cell_(microprocessor))

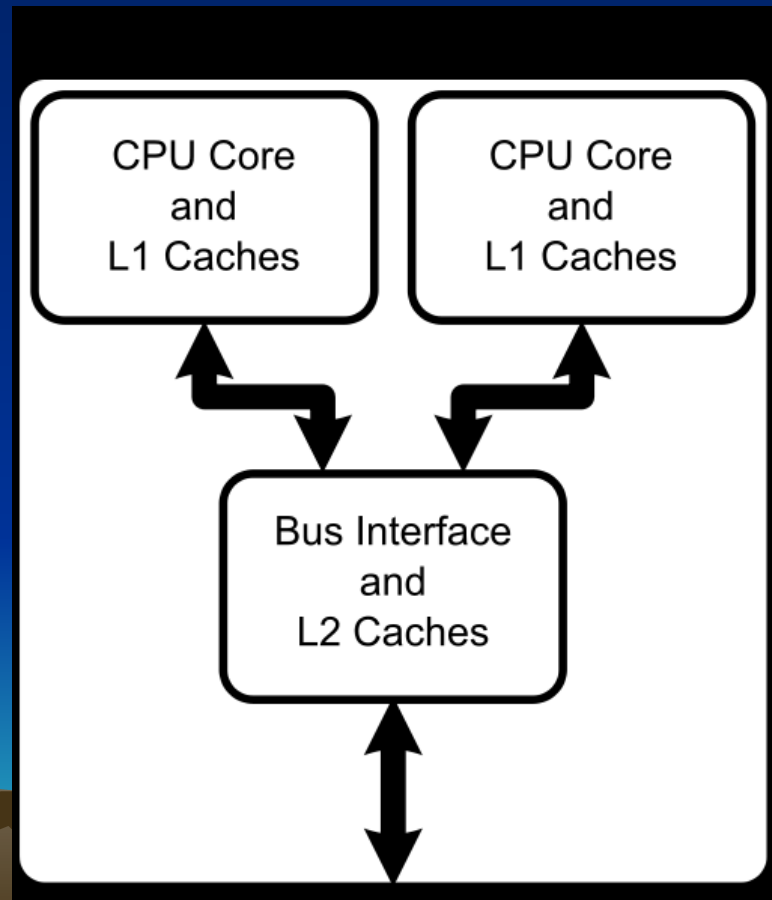


Czym są procesory wielordzeniowe (multicore)?

- Procesory wielordzeniowe jak sama nazwa wskazuje zawierają w sobie dwa lub więcej rdzeni. Pozwalają na równoległe wykonywanie obliczeń.



Czym są procesory wielordzeniowe (multicore)?



Jakie są zalety procesorów wielordzeniowych?

- Zmniejszenie kosztów produkcji (mniej połączonych nóżek na rdzeń).
- Możliwość połączenia wielu rdzeni o różnym przeznaczeniu (np. CPU + GPU + PPU).
- Mniejsze zużycie energii a przez to także mniej wydalanego ciepła.
- Prostota. Nie trzeba zmieniać architektury. Wystarczy dodać kolejny stary procesor w roli kolejnego rdzenia.



Jakie są wady procesorów wielordzeniowych?

- Dwa rdzenie o taktowaniu X są nie szybsze niż jeden rdzeń o taktowaniu $2X$. (przy tej samej architekturze).
- Wymaga od programistów umiejętności programowania współbieżnego.
- Stare programy nie są w stanie wykorzystać dodatkowych rdzeni procesora.



Jakie są wady procesorów wielordzeniowych?

- Wszystkie rdzenie dzielą tę samą szynę systemową. Jeżeli algorytm jest współbieżny ale źle napisany (pod numerycznym względem) nawet jeden rdzeń nie zostanie do końca wykorzystany.
- Istnieją problemy nie dające się zrównoległać (algorytmy numeryczne).



Komercyjne wykorzystanie

- Serwery (np. WWW).
- Algorytmy bardzo równoległe ("embarrassingly parallel").
- Komputery domowe.
- Gry (PS3 ma 8 rdzeni Xbox 360 ma 3 rdzenie).



Ile na tym zyskamy?

- Dla zadań "embarrassingly parallel" zyskamy proporcjonalnie do ilości rdzeni.
- W rzeczywistości nie jest tak różowo. Nawet teraz większość oprogramowania korzysta tylko z jednego rdzenia. Korzyści jednak będą zawsze. W systemie w każdej chwili działa więcej niż jeden proces (np. antywirus).
- Programy które są równoległe najczesciej nie są takie w 100% (prawo Amdahla).



Ile na tym zyskamy?

- Mamy większą szansę wyłapać błąd pisząc zadanie na PW albo SO ;)
- Statystycznemu informatykowi w dzisiejszych czasach semafor się kojarzy z pociągami.



Misc

- Wiele dostępnego komercyjnie oprogramowania (MS Windows Server, VMware Workstation) ma licencję per procesor. Może się okazać, że trzeba kupić tyle licencji ile się ma rdzeni.



Przyszłość

- Rdzenie do konkretnych specyficznych zadań (Heterogeneous).
- Dużo więcej rdzeni. Nie ma co liczyć na zwiększenie taktowania.
- Komputery kwantowe.



Architektura pamięci



Architektura pamięci

- Większość nowoczesnych systemów operacyjnych wykorzystuje do zarządzania pamięcią mechanizm stronicowania.
- Niektóre systemy, takie jak MULTICS wykorzystują segmentację połączoną ze stronicowaniem.
- Pokażemy w jaki sposób segmentacja może posłużyć do rozszerzenia obsługi pamięci fizycznej ponad 4 GB.



Przypomnienie



Przypomnienie - nazewnictwo

- W ramach programu korzystamy z *adresów logicznych*.
- Do komunikacji z pamięcią podręczną procesor wykorzystuje *adresy fizyczne*.
- Adresy logiczne są tłumaczone za pomocą wybranego algorytmu na *adresy fizyczne*.



Przypomnienie

- Istnieją dwie podstawowe techniki zarządzania pamięcią: *segmentację* oraz *stronicowanie*.
- Segmentacja i stronicowanie są procesami niezależnymi, dlatego możliwe jest ich łączenie.
- W takim przypadku poprzez segmentację przekształcamy adres logiczny na *adres liniowy*. Następnie adres liniowy poddajemy stronicowaniu otrzymując adres fizyczny.

Przypomnienie – segmentacja



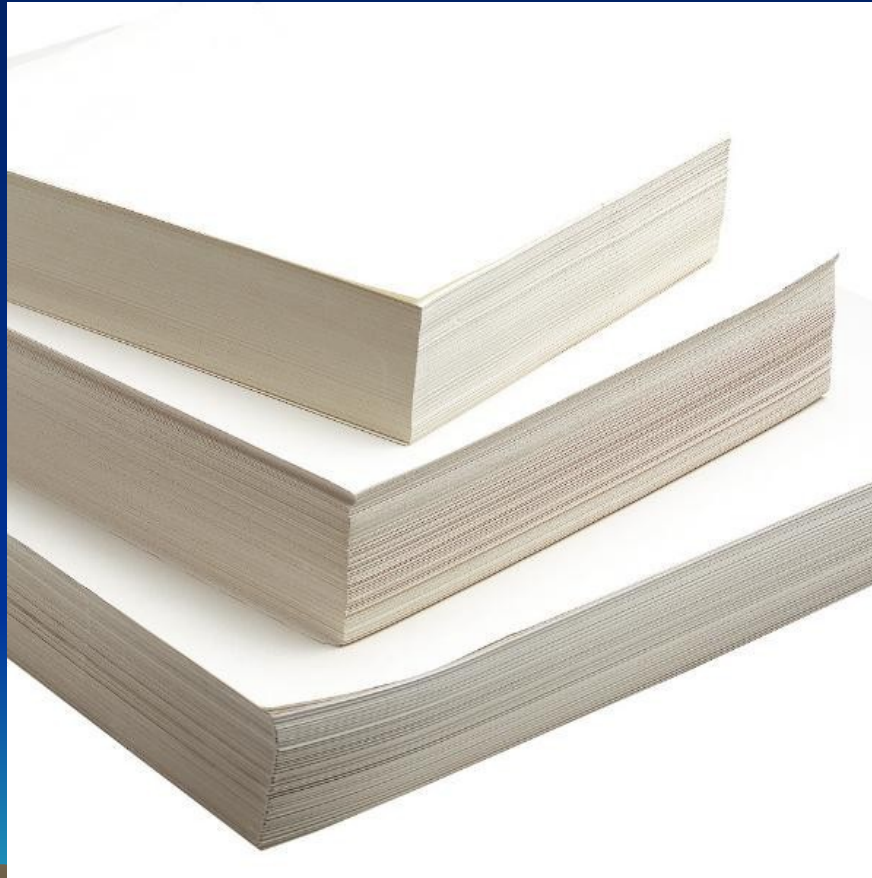
Przypomnienie – segmentacja

- Segmentacja polega na podzieleniu programu na logiczne fragmenty, które mogą być następnie przechowywane w dowolnym porządku w *przestrzeni adresów liniowych*.
- Dzięki segmentacji system operacyjny może zapewniać współdzielenie kodu oraz danych między procesami.
- Segmentacja wspomaga ochronę pamięci.
- Segmentacja zmniejsza fragmentację zewnętrzną *przestrzeni adresów liniowych*.

Przypomnienie – segmentacja

- Segmentacja jest realizowana poprzez zastąpienie pojedynczego *płaskiego* adresu logicznego parą: <nr segmentu, przesunięcie>
 - Wydajna segmentacja wymaga obecności w procesorze wbudowanych mechanizmów wyliczania adresu liniowego.
 - W takim modelu numer segmentu jest przekazywany poprzez specjalnie do tego wyznaczone rejestry.
- 

Przypomnienie – stronicowanie



Przypomnienie – stronicowanie

- Stronicowanie polega na:
 - Podziale pamięci fizycznej na *ramki* równej wielkości.
 - Podziale przestrzeni adresów liniowych na *strony* (tej samej wielkości co ramki).
 - Utrzymywaniu funkcji *mapującej* strony na ramki.
- Stronicowanie pozwala na ukrycie różnic między pamięcią fizyczną a logiczną.
- Stronicowanie rozwiązuje problem *fragmentacji zewnętrznej* pamięci fizycznej.

Przypomnienie – stronicowanie

- Dzięki stronicowaniu możliwe jest odsyłanie niewykorzystywanych zasobów na dysk (tzw. wymiana) bez wiedzy procesów.
- Stanowi szczególnie dobre rozwiązanie gdy pamięć logiczna jest znacznie większa niż pamięć fizyczna.



Zarządzanie pamięcią - przykłady

- Przedstawimy wsparcie dla zarządzania pamięcią w procesorach i386 i późniejszych z serii x86.
- Dla kontrastu omówimy również działanie MMU w architekturze PowerPC.
- Opiszemy również inne często stosowane rozwiązania.



Zarządzanie pamięcią w



MMU w i386

- Procesory serii x86 począwszy od i386 posiadają wsparcie dla segmentacji oraz stronicowania.
- Segmentacja jest wspierana we wszystkich trybach pracy procesora, jednak działa różnie w każdym z nich.
- Mechanizm stronicowania jest dostępny tylko w *trybie chronionym*.



Tryb rzeczywisty



Tryb rzeczywisty – segmentacja

- Procesor dysponuje 6 *rejestrami segmentowymi*.
- Segmentacja polega po prostu na dodaniu do adresu logicznego *adresu bazowego*.
- Adres bazowy jest uzyskiwany przez przemnożenie wartości wybranego rejestru segmentowego przez 16 (przesunięcie o 4 bity).
- Uzyskujemy 20-bitową liczbę (gdy jest 21-bitowa, obcinamy najbardziej znaczący bit). Stanowi ona adres fizyczny.

Tryb rzeczywisty – segmentacja

- W trybie rzeczywistym procesor jest w stanie zaadresować trochę ponad 1 MB pamięci.
- Dzięki segmentacji możliwe było adresowanie większej ilości pamięci niż wynikało to z 16-bitowego trybu pracy.
- Segmentacja dostarczała również mechanizmów ochrony pamięci przed zapisem i wykonywaniem.



Tryb chroniony



Tryb chroniony – segmentacja

- W trybie chronionym również dostępna jest segmentacja.
- Tym razem jednak rejestry segmentowe pełnią rolę *deskryptorów*. Ich wartości służą jako indeksy w *lokalnej* lub *globalnej* *tablicy deskryptorów*.
- Każdy proces posiada swoją własną *lokalną* *tablicę deskryptorów*.
- Każda z tych tablic zawiera po 8192 wpisy.

Tryb chroniony – segmentacja

- Aby przyspieszyć działanie segmentacji, procesor dysponuje specjalnymi, niewidocznymi dla programisty *rejestrami deskryptorów*.
- W momencie przypisywania wartości do rejestru segmentowego, automatycznie aktualizowany jest odpowiadający mu rejestr deskryptora.
- Procesor i386 nie posiada opcji wyłączenia segmentacji. Można jednak utworzyć pojedynczy segment długości 4GB.

Tryb chroniony - stronicowanie

- W trybie chronionym procesora i386 wspierane jest stronicowanie.
- Procesor dostarcza tablicę *TLB* (ang. *Translation Lookaside Buffer*), która automatycznie potrafi skojarzyć podany adres logiczny z jednym ze wpisów w tablicy.
- W przypadku nieobecności takiego wpisu generowany jest *page fault* a adres brakującej strony zostaje przekazany w specjalnym rejestrze procesora (*CR2*).

Tryb chroniony - stronicowanie

- Strony mają wielkość 4 KB.
- Wykorzystywane są tablice stron oraz katalogi tablic stron, dzięki czemu obydwa rodzaje tablic same mieszczą się na pojedynczych stronach.
- Podobnie jak w przypadku segmentacji mamy specjalny rejestr procesora służący do przechowywania adresu katalogu tablic stron (część rejestru *CR3*).



Zarządzanie pamięcią w



Pentium - adresowanie

- W odpowiedzi na zwiększające się zapotrzebowanie na pamięć operacyjną (szczególnie w maszynach serwerowych), projektanci Intel'a wprowadzili nowe tryby adresowania.
- Nowe tryby adresowania to:
 - PSE-36 – zwiększa wielkość strony z 4 KB do 4 MB (lub 2 MB, gdy włączone jest również PAE).
 - PAE – dodaje do katalogu stron dodatkowy poziom, nazywany *tablicą wskaźników do katalogów*.

Pentium - adresowanie

- W obydwu tryby pozwalają na adresowanie przestrzeni 36-bitowej.
- Jednak dla pojedynczej aplikacji nadal dostępne jest 4 GB pamięci.
- Z tego powodu budowanie aplikacji potrafiących wykorzystać całą dostępną pamięć jest zadaniem skomplikowanym – muszą to być aplikacje złożone z wielu procesów.



Zarządzanie pamięcią w x86-64



X86-64 – adresowanie

- Zwiększenie pamięci adresowej do 48 bitów.
- Perspektywa rozszerzenia adresowania do 64-bitowego w dalszej perspektywie.
- Obecne implementacje wykorzystują 40 bitów.
- Stronicowanie dodaje dwa nowe poziomy stronicowania w stosunku do domyślnego (tablicę wskaźników do katalogów oraz *poziom PML4*).

Wnioski

- Przejście na adresowanie 64-bitowe (w praktyce 48-bitowe) jest nieuchronne.
- 64 GB pamięci dostępne w 36-bitowych trybach już jest niewystarczające dla niektórych zastosowań serwerowych.
- Dopiero tryby adresowania oferowane przez x86-64 pozwalają na *jednoczesny* dostęp do więcej niż 4 GB pamięci.



Działanie TLB

- Zazwyczaj do tablicy TLB dodawana jest dodatkowa *tablica PTPC* przechowująca adresy tablic stron.
- W większości przypadków możemy obsłużyć *page miss* za pomocą pojedynczego sięgnięcia do pamięci.
- W architekturze PowerPC adresy stron i tablic stron są trzymane w TLB. Procesor wykonuje dwa przeszukania *jednocześnie*.



OS a architektura



Wykorzystanie SMT

- Domyślnie *SMT* widoczne jako oddzielne procesory logiczne.
- Problemy
 - Średnio tylko paręnaście procent przyspieszenia w stosunku do braku *SMT*.
 - W przypadku obecności wielu procesorów istotne jest równoważenie obciążenia, system musi być świadomy architektury.



Wykorzystanie SMT

- Przy ponownym przydzielaniu procesora temu samemu wątkowi systemy operacyjne wykorzystują zjawisko *ciepłego cache'u*
- W przypadku SMT system powinien być świadomy, że wybór rdzenia nie ma znaczenia dla trafień w cache (obydwa poziomy są współdzielone).



Wykorzystanie Multi-core

- W przypadku współdzielenia cache'u drugiego poziomu dwa programy wykonywane na tym samym rdzeniu zazwyczaj będą miały gorszą wydajność (np. Intel Core 2 Duo).
- W wyjątkowych sytuacjach mogą mieć wyższą wydajność. Wspólny cache może być zaletą.
- Rdzenie w AMD 64 X2 nie współdzielą cache'u.

Wykorzystanie Multi-core

- Analogiczna sytuacja ze zjawiskiem *ciepłego cache'u*, jak w SMT.
- System może optymalizować wykorzystanie energii. Wówczas jak najwięcej procesów powinno być wykonywanych na rdzeniach tego samego procesora, umożliwiając obniżenie taktowania pozostałych procesorów.



Architektury wieloprocessorowe

- Najprostszą architekturą jest SMP (Symmetric Multiple Processors). Dostęp do pamięci jest taki sam dla wszystkich procesorów.
- NUMA – Non Uniform Memory Access.
 - Dostęp do pamięci związanej z danym procesorem szybki.
 - Dostęp do pamięci związanej z innymi procesorami wolniejszy.
- System powinien ograniczać migrację zadań między procesorami z różnych domen.

Przykłady systemów

- Skoncentrujemy się na edycjach 64-bitowych i na problemach z nimi związanych.
- Przytoczymy wyniki testów porównujących niektóre edycje 64-bitowe Linuxa z Windowsem.



Windows Vista x86-64

- System Windows Vista działający w architekturze x86-64.
- System nie obsługuje 32-bitowych sterowników.
- Problem z dostępnością 64-bitowych edycji aplikacji.
- Aplikacje 32-bitowe są obsługiwane poprzez WOW64 (Windows-on-Windows 64-bit).
- Ze względu na kompatybilność x86 oraz x86-64 różnice wydajnościowe nie są duże.

Windows Vista x64

- Systemy Windows Vista przeznaczone do działania w architekturze IA-64.
- Podobnie jak w Vista x86-64 brak obsługi 32-bitowych sterowników oraz brak 64-bitowych edycji wielu popularnych aplikacji.



Windows Vista x64

- IA-64 posiada wsparcie dla IA-32, jednak jego osiągi były tak słabe, że w zamian wykorzystywany został WOW.
- WOW64 tworzy 32-bitowe środowisko zgodne z x86. Różnice w obydwu architekturach powodują duże straty wydajnościowe.



64-bitowe dystrybucje systemu Linux

- System Linux znacznie wyprzedził Windows w obsłudze procesorów 64-bitowych.
- Przyczyną jest przede wszystkim portowalna architektura systemu oraz opóźnienie Microsoftu w pracy nad własną edycją 64-bitową.
- Zachowanie kompatybilności z 32-bitowymi aplikacjami.



64-bitowe dystrybucje systemu Linux

- Dodatkową zaletą Linux'a jest dużo większa dostępność 64-bitowych odpowiedników programów (praktycznie każdy szanujący się projekt dostarcza taką wersję).
- Możliwość samodzielnej kompilacji programów z otwartymi źródłami pozwala na lepsze wykorzystanie możliwości konkretnego procesora.



Problemy

- Bardzo podobne problemy bez względu na dystrybucję (OpenSUSE, Fedora Core, Ubuntu)
- Występowały problemy z działaniem wtyczek flash'owych oraz Javy. Podobno zostały usunięte.
- W wielu edycjach występowały problemy ze wtyczkami do programów takich jak Firefox (Potrzeba użycia nspluginwrapper).
- Problemy z dostępnością 64-bitowych wersji programów z zamkniętymi źródłami – np. Opery (Jest już dostępna).

Wydajność

- Brak zestawień Vista 64-bit vs Linux 64-bit.
- Testy 64-bitowego Windows XP pokazywały jego znaczącą stratę do Linux'a (rzędu parunastu procent).



Źródła

- Mistrzostwo świata:
<http://www.cl.cam.ac.uk/techreports/UCAM>
- Ciekawostki:
http://en.wikipedia.org/wiki/ARM_architectu
- <http://en.wikipedia.org/wiki/Multi-core>

