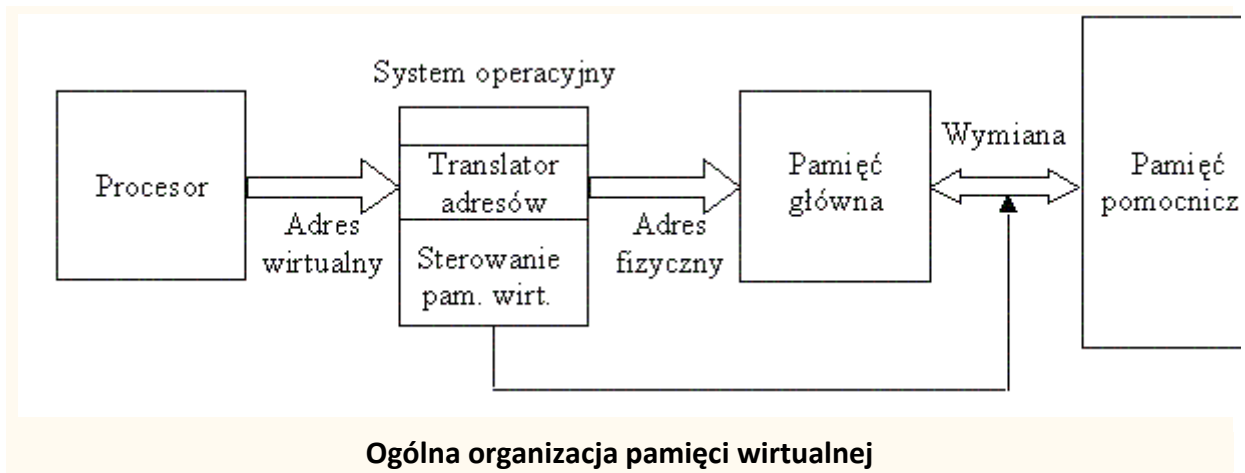


## Wykład 7. Zarządzanie pamięcią. Pamięć wirtualna.

Pamięć operacyjna, obok procesora, jest kluczowym zasobem systemu komputerowego dla wykonywania programów. Zarządzanie pamięcią jest jednak dość skomplikowane, ponieważ jej poszczególne części są w tym samym czasie wykorzystywane przez wiele procesów oraz przez jądro systemu operacyjnego. Stabilność pracy systemu komputerowego wymaga odpowiedniej ochrony przestrzeni użytkowników i jądra systemu.

**Podstawowe zadania, realizowane w ramach zarządzania pamięcią operacyjną obejmują** przydział pamięci i jej odzyskiwanie, ochronę, udostępnianie w celu współdzielenia, transformację adresów oraz transfer danych. Zadania te podzielone są między układy sprzętowe na poziomie architektury komputera, a system operacyjny. Ze względu na efektywność realizacji, niektóre zadania są realizowane przez odpowiednie układy sprzętowe. Zadaniem systemu jest dostarczenie tym układom odpowiednich danych, które wynikają z wcześniejszych decyzji o przydziale pamięci, co należy do kompetencji systemu operacyjnego.

**Pamięć wirtualna** jest techniką pozwalającą na sztuczne zwiększenie ilości pamięci RAM poprzez użycie w jej charakterze pamięci masowej (np. twardego dysku).



**Pamięć wirtualna** jest techniką programową a także sprzętową gospodarowania pamięcią operacyjną RAM pozwalającą na przydzielanie pamięci dla wielu procesów, zwalnianie jej i powtórne przydzielanie, w ilości większej niż rzeczywista ilość pamięci fizycznej zainstalowanej w komputerze poprzez przeniesienie danych z ostatnio nie używanej pamięci do pamięci masowej (np. twardego dysku); w sytuacji gdy procesor odwołuje się do danych z pamięci przeniesionej na dysk przesuwa się te dane do pamięci w wolne miejsce, a gdy brak wolnej pamięci - zwalnia się ją przez wyżej opisane przerzucenie jej na dysk.

Najczęściej spotykane są dwa sposoby przechowywania danych zrzucanych z pamięci fizycznej na dysk. Pierwszy, stosowany w systemach rodziny Windows polega na zapisie pamięci w pliku znajdującym się na ustalonej partycji komputera. Drugi, stosowany w systemach z rodziny UNIX to utworzenie osobnej partycji wymiany (*partycji swap*) przeznaczonych wyłącznie na pamięć wirtualną. Zapewnia to szybszy dostęp do danych niż pierwsze rozwiązanie (głównie ze względu na ominięcie obsługi systemu plików).

Wyróżniamy kilka sposobów zarządzania pamięcią operacyjną w systemie: segmentacja, stronicowanie czy segmentacja ze stronicowaniem.

## SEGMENTACJA

Segmentacja pamięci – jedna z metod ochrony pamięci, używana przy wielozadaniowości. Każdy proces otrzymuje swój własny obszar pamięci, realizowany poprzez rejestry segmentowe.

Segmentacja pamięci polega na podzieleniu przez procesor pamięci fizycznej na ciągłe bloki nazywane segmentami, które opisane są przez deskryptory zawierające:

- adres bazowy – fizyczny adres początku segmentu w pamięci,
- rozmiar – długość segmentu w ustalonych jednostkach (np. w bajtach, paragrafach),
- atrybuty określające rodzaj zawartości i dostępność (np. kod programu, dane tylko do odczytu, stos itp., pierścień ochrony) – na potrzeby weryfikacji poprawności odniesień,
- identyfikator (określany też jako nazwa lub numer) – wartość wskazująca na opis segmentu w tablicy segmentów (najczęściej indeks w tablicy segmentów). Jeśli identyfikatorem segmentu jest indeks, to jego wartość nie jest przechowywana w deskrytorze, ale wynika z lokalizacji deskryptora w tablicy.

System tworzy takie segmenty na żądanie aplikacji, przekazując jej jedynie identyfikatory nie pozwalające na odczytanie parametrów segmentów. Programy odwołują się zatem do kolejnych komórek pamięci w ramach należących do nich segmentów, nie wiedząc nic o tym, w jakie miejsca pamięci fizycznej trafiają odwołania do nich. Procesy nie mają też prawa „wiedzieć” segmentów należących do innych programów – w czasie przekazywania kontroli procesowi system musi zablokować definicje segmentów należących do pozostałych procesów, przy każdym przełączeniu blokując segmenty wyłączanego programu i na nowo uaktywniając segmenty programu aktywowanego.

Parametry segmentu przechowywane są w 8 bajtowym rekordzie nazywanym deskryptorem segmentu.

adres bazowy 31..24	G	X	0	A	limit 19.16	P	DP L	1	typ	A	B adres bazowy 23..16
B adres bazowy segmentu 15..0						L limit segmentu 15..0					

**Deskryptor segmentu**

B - adres bazowy segmentu

L - długość segmentu

G - sposób interpretacji limitu segmentu (0 – bajty, 1 – strony 4KB),

DPL - poziom uprzywilejowania segmentu (0-najwyższy, 3-najniższy),

P - bit obecności segmentu (używany w pamięci wirtualnej),

AV - nie używany,

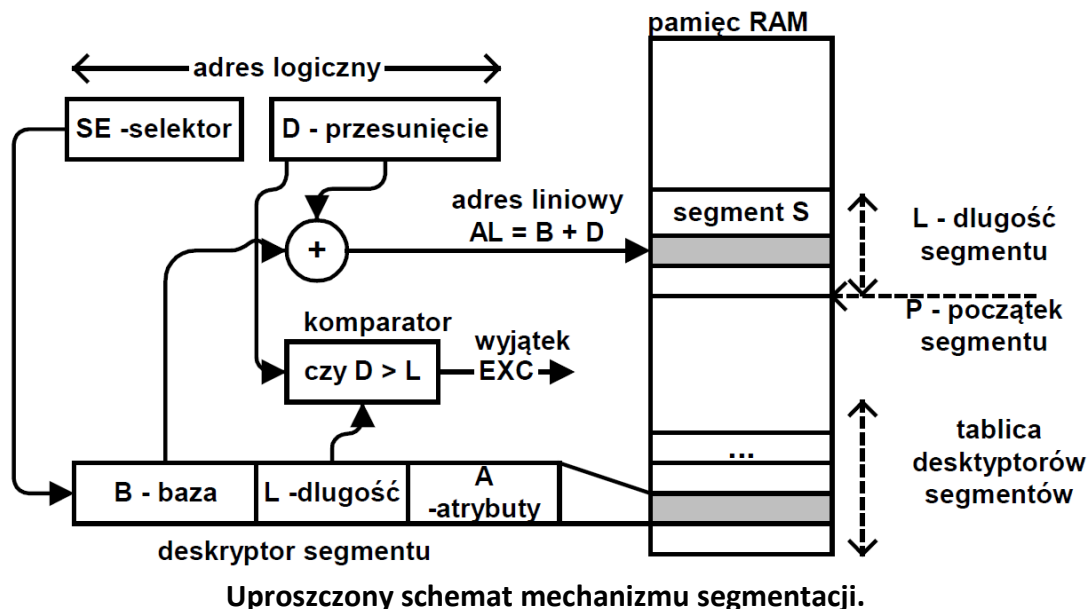
A - mówi czy deskryptor jest używany.

Deskryptory są przechowywane w dwóch rodzajach tablic:

- **globalnej tablicy deskryptorów GDT** (ang. *Global Descriptor Table*)
- **lokalnej tablicy deskryptorów LDT** (ang. *Local Descriptor Table*).

W systemie istnieje:

- jedna tablica GDT - opisuje segmenty widoczne dla wszystkich procesów
- wiele lokalnych tablic deskryptorów LDT (ang. *Local Descriptor Table*), opisujących prywatne segmenty procesów

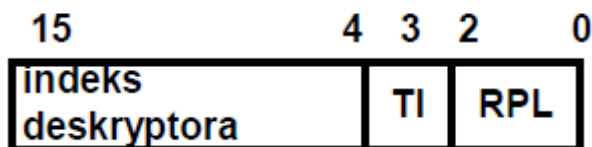


Adres logiczny składa się z:

- selektora segmentu SE
- przesunięcia D.

Funkcje selektora pełni jeden z rejestrów segmentowych:

- dla kodu selektorem jest rejestr CS.,
- dla danych rejestr DS., ES,
- dla stosu SS.



RPL - requested privilege level  
TI - table indicator

#### Zawartość selektora segmentu

Selektor zawiera:

- indeks deskryptora - położenie segmentu znajdującego się w tablicy deskryptorów,
- TI - określa o którą tablicę chodzi (0 – GDT, 1 - LDT)
- RPL - żądany poziom uprzywilejowania – określa poziom uprzywilejowania procesu.

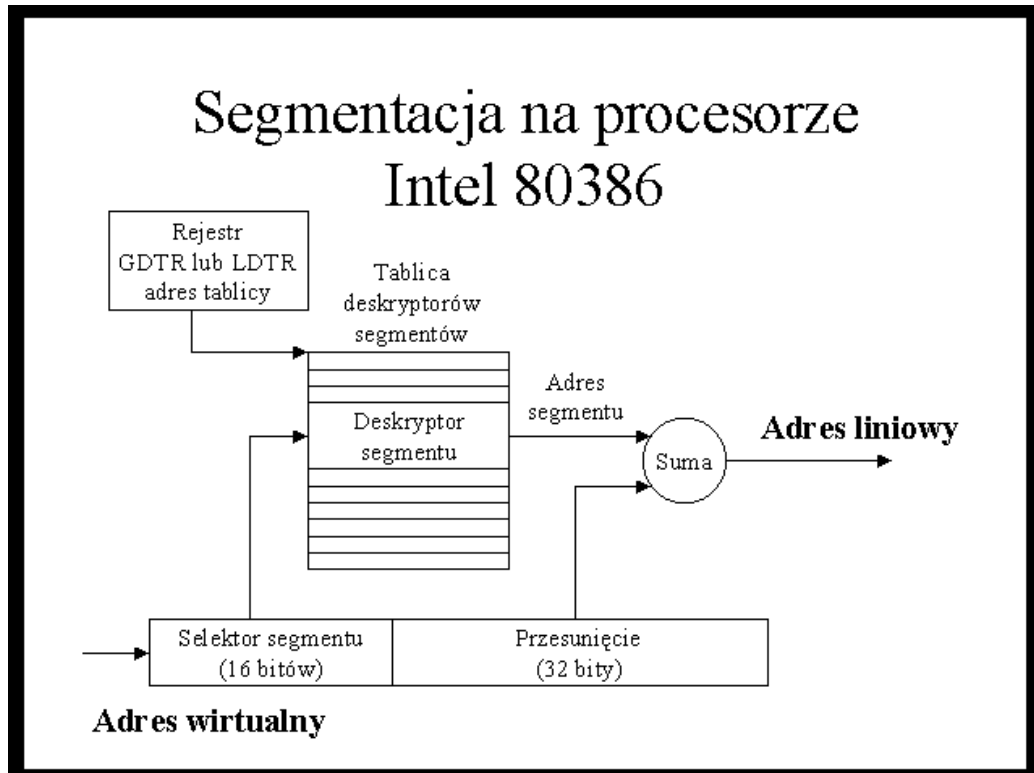
Adres liniowy - suma pobieranego z pola adresowego rozkazu przesunięcia D i adresu początku segmentu B pobieranego z deskryptora.

Komparator sprawdza czy przesunięcie D nie wykracza poza długość segmentu L zapisanego w deskryptorze. Gdy tak się zdarzy generowany jest wyjątek EXC który powoduje wywołanie

systemu operacyjnego. System operacyjny podejmuje decyzję, co zrobić z naruszającym przydzielony segment procesem.

Adres liniowy może być poddany przetwarzaniu przez mechanizm stronicowania.

## SEGMENTACJA NA PROCESORZE INTEL 80386



### PROCES TŁUMACZENIA ADRESU WIRTUALNEGO NA LINIOWY

Kiedy proces odwołuje się do pamięci z odpowiedniego rejestru segmentowego (CS, SS, DS, ED, FS, GS) pobierany jest selektor segmentu. Na jego podstawie odnajdowany jest odpowiadający mu deskryptor: wartość w polu TI (wskaźnik tablicy - w związku z istnieniem dwóch tablic deskryptorów: lokalnej i globalnej - bit ten wskazuje o którą tablicę chodzi: 0 - globalna, 1 - lokalna) określa tablicę, numer deskryptora jest indeksem w tej tablicy, a wartość RPL (poziom ochrony zadania żądającego dostępu) musi być mniejsza lub równa od numeru uprawnienia wskazywanego przez selektor (w przeciwnym przypadku generowane jest przerwanie wewnętrzne). Z deskryptora pobierany jest adres bazowy i dodawany do przesunięcia - w ten sposób wyliczony zostaje adres liniowy.

### ZALETY SEGMENTACJI

- Skuteczna, prosta relokacja kodu i danych – nieważne jest, gdzie w pamięci fizycznej znajduje się segment, program może odwoływać się do kolejnych słów pamięci w ramach segmentu, licząc od zera do końca segmentu
- Bardzo dobra ochrona, wynikająca ze struktury logicznej przestrzeni adresowej
- Możliwość łatwego współdzielenia kodu oraz danych
- Brak wewnętrznej fragmentacji
- Wyższy stopień wieloprogramowości

## WADY SEGMENTACJI

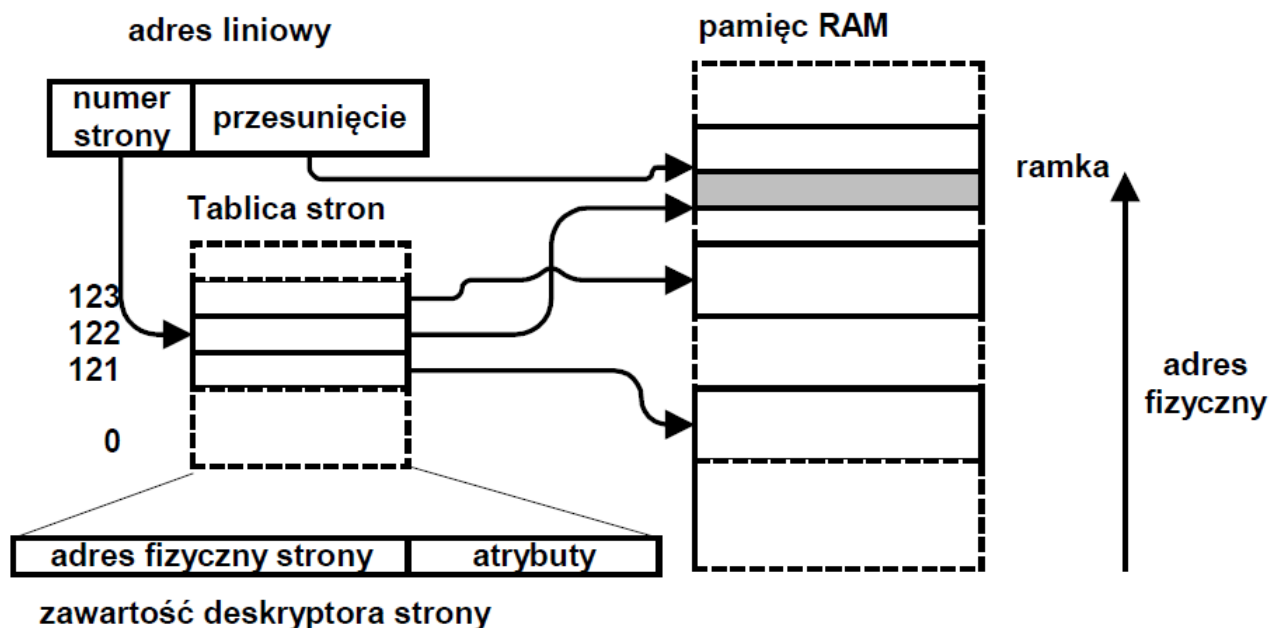
- Komplikacja modelu programowego (dwuelementowy adres)
- Problematiczna i niewydajna dynamiczna alokacja
- Praktycznie nierealizowalna pamięć wirtualna
- Duże ograniczenia na rozmiary segmentów – co zmuszało do dzielenia kodu programów oraz bloków danych w sposób nienaturalny, utrudniając tworzenie wielkich struktur danych (te ograniczenie zostało później zniesione, ale dzielenie programu na logiczne części jest dalej skomplikowane)

## STRONICOWANIE

Stronicowanie – jeden ze sposobów rozwiązania problemu zewnętrznej fragmentacji, polegający na dopuszczeniu nieciągłości rozmieszczenia logicznej przestrzeni adresowej procesu w pamięci fizycznej. Jest to podział pamięci na mniejsze obszary o ustalonej lub zmiennej wielkości i przydzielanie tym blokom adresów fizycznych lub logicznych.

Podstawowa filozofia stronicowania:

- Pamięć fizyczna dzielona jest na bloki stałej długości zwane ramkami.
- Pamięć logiczna dzielona jest na bloki stałej długości zwane stronami.
- Rozmiary stron i ramek są identyczne.
- Przy wykonywaniu procesu, strony z pamięci pomocniczej wprowadzane są w odpowiednie ramki pamięci operacyjnej.



### Schemat stronicowania

W celu odróżnienia stron z obrazem procesu od stron pamięci fizycznej, strony fizyczne nazywa się ramkami (ang. frames). Pamięć procesu tworzą kolejne strony, natomiast ramki mogą być porzucane gdziekolwiek w pamięci. Ich fizyczna lokalizacja zawarta jest w tablicy stron.

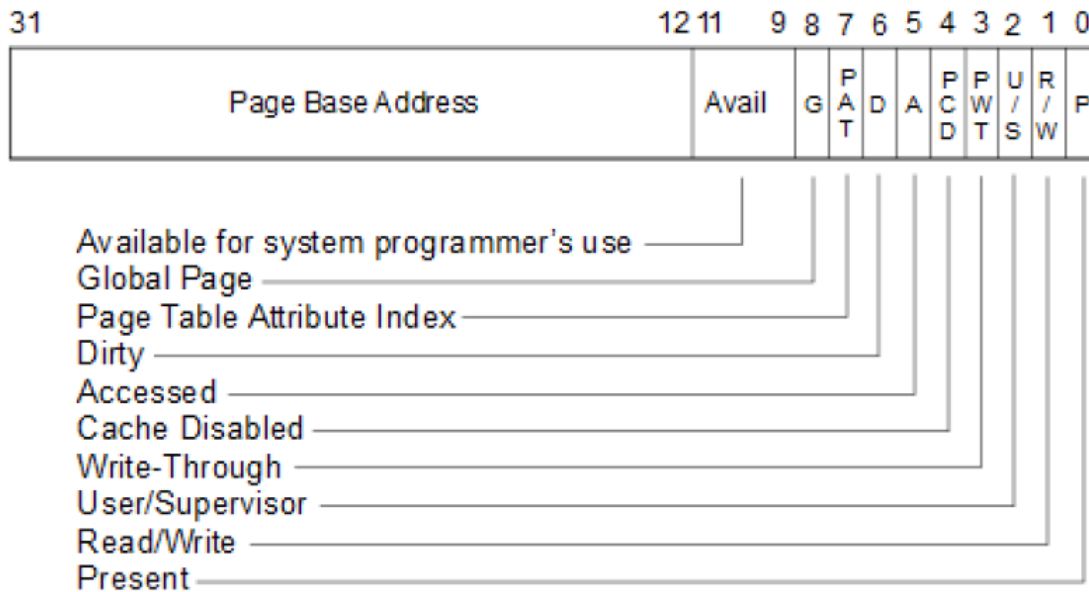
Na powyższym rysunku proces zawiera strony 121,122,123 które w adresie liniowym są po kolei, natomiast w pamięci fizycznej mogą być umieszczone gdziekolwiek.

## Tablica stron

Tablica stron składa się z 32 bitowych słów, każde z nich opisuje pojedynczą stronę.

Opis strony zawiera:

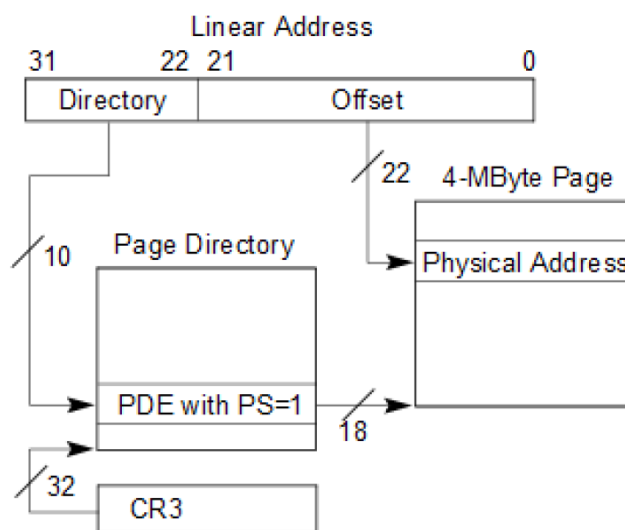
- Adres fizyczny strony
- Atrybuty strony: bit obecności, bit dostępu, możliwość zapisu, przywileje administratora, czy był zapis.



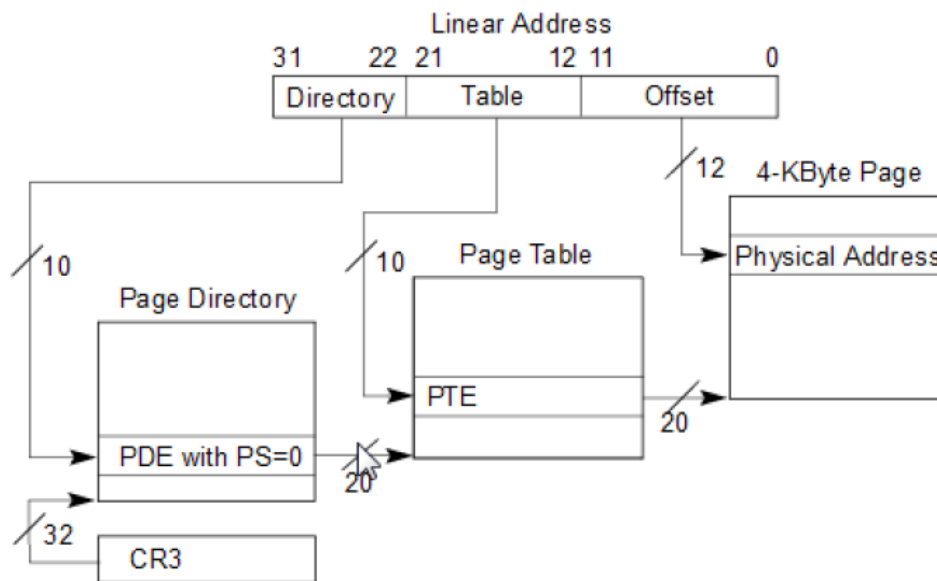
**Zawartość deskryptora strony**

W architekturze IA-32 stosuje się dwa schematy stronicowania:

- Stronicowanie jednopoziomowe (strona ma 4MB)
- Stronicowania dwupoziomowe (strona 4 KB)



**Stronicowanie jednopoziomowe, strona 4MB**



**Stronicowanie dwupoziomowe, strona 4kB**

Stronicowanie pozwala na rozwiązanie problemu fragmentacji pamięci kosztem dodatkowego obszaru pamięci potrzebnego na tablicę stron.

Stronicowanie 2 poziomowe pozwala na oszczędność tej pamięci.

Co tracimy:

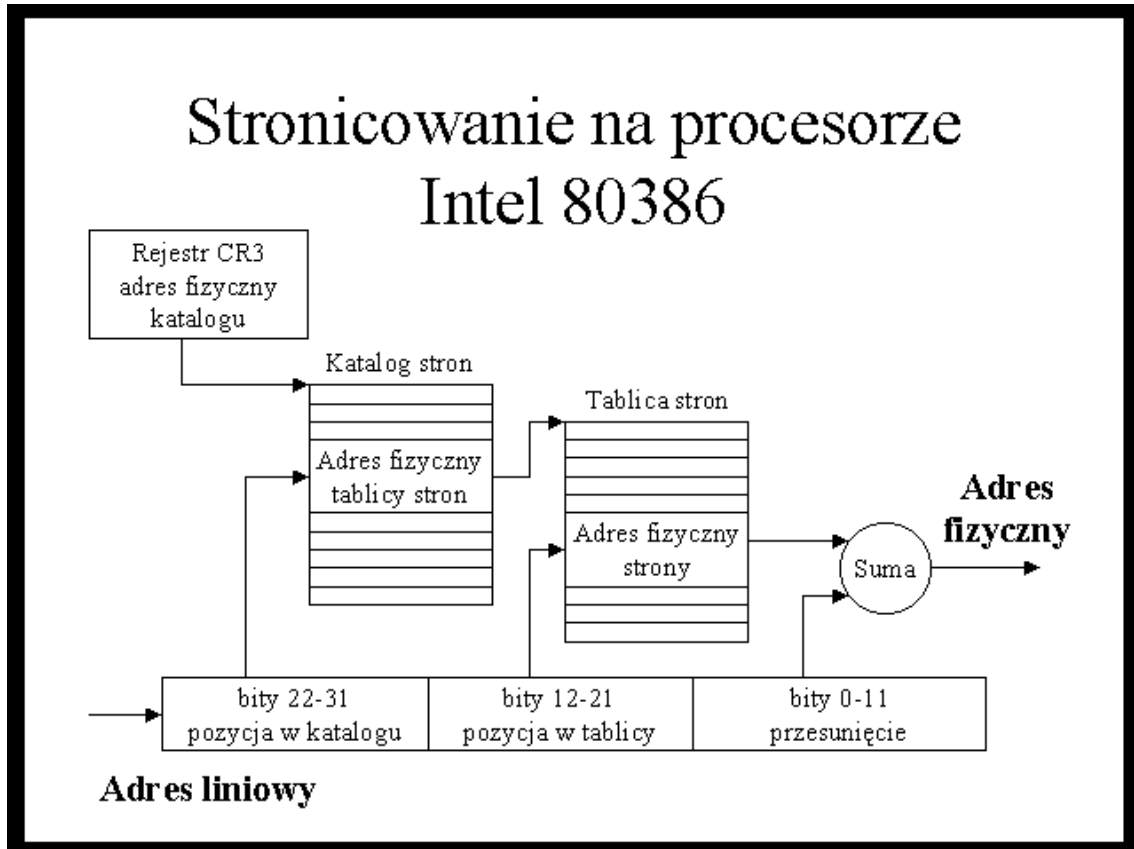
- Zwiększony czas dostępu do komórki pamięci (adresacja pośrednia)
- Tablica stron zajmuje pamięć
- Fragmentacja wewnętrzna

<b>Segmentacja</b>	Ochrona obszarów pamięci używanych przez procesy przed dostępem przez inny proces
	Zapewnienie przemieszczalności programów
<b>Stronicowanie</b>	Realizacja pamięci wirtualnej większej niż fizyczna
	Rozwiązanie problemu fragmentacji
	Ochrona obszarów pamięci

**Porównanie: Mechanizmy sprzętowe zarządzania pamięcią**

### STRONICOWANIE PAMIĘCI FIZYCZNEJ

Stronicowanie pamięci fizycznej wykonywane było z powodu ograniczenia przestrzeni adresowej procesora (stronicowanie fizyczne). Stronicowanie takie stosowano w systemach 8- i 16-bitowych, w których ilość zainstalowanej pamięci przekraczała możliwości adresowania pamięci przez procesor.



Procesory 80386 i nowsze pracujące w trybie chronionym umożliwiają dowolne mapowanie adresów logicznych na adresy fizyczne. Adresy logiczne obejmują całą przestrzeń adresową procesora, czyli 4 GB, niezależnie od tego, ile w rzeczywistości w komputerze zainstalowano pamięci. Zadaniem systemu operacyjnego jest odpowiednie mapowanie adresów logicznych na adresy pamięci fizycznej, co pozwala zwykłym programom użytkowym przez cały czas działania odwoływać się do tych samych adresów logicznych.

Jeśli włączone jest stronicowanie, wówczas cała pamięć (4GB) dzielona jest na bloki – *strony* o rozmiarach 4kB; w procesorach Pentium i nowszych możliwe jest także używanie stron o rozmiarach 4MB. Gdy program odwołuje się do pamięci, a więc podaje 32-bitową liczbę – numer komórki pamięci – jest ona rozbijana na trzy części:

1. indeks w **katalogu stron** (liczba 10-bitowa),
2. indeks w **tablicy stron** (liczba 10-bitowa),
3. **przesunięcie** w obrębie strony (liczba 12-bitowa).

**Katalog stron** zawiera wskaźniki do **tablic stron**, **tablice stron** przechowują adresy fizyczne stron. (System operacyjny może zarządzać wieloma katalogami i tablicami stron).

Zatem pierwsza część adresu wybiera z katalogu stron tablicę stron. Druga część adresu wybiera pozycję z tablicy stron, która wyznacza fizyczny adres konkretnej strony. **Przesunięcie** jest adresem lokalnym w obrębie wybranej strony. Zatem adres fizyczny na który zamapowano adres logiczny jest wyznaczany z dwóch składników: adresu fizycznego strony i przesunięcia.



Ponieważ jest to dość skomplikowana operacja, dlatego procesory posiadają pamięć podręczną (**TLB** — **T**ranslation **L**ookaside **B**uffer), w której pamiętane są ostatnio wyznaczone adresy fizyczne stron. TLB przechowuje 32, 64, 256 lub więcej pozycji.

Każda pozycja w tablicy stron przechowuje pewne dodatkowe informacje dotyczące strony pamięci, także te przydatne w realizacji pamięci wirtualnej:

- **Bit obecności** – mówi czy strona znajduje się w pamięci fizycznej. Gdy system operacyjny usuwa stronę z pamięci, tzn. zapisuje ją na dysk, zeruje ten bit. Jeśli program odwoła się do strony nieobecnej w pamięci procesor generuje wyjątek i wówczas system operacyjny wczytuje z dysku uprzednio zapisaną tam zawartość strony.
- **Bit użycia** – jest ustawiany przez procesor, gdy nastąpi odwołanie do danej strony. System operacyjny zwalnia pamięć zapisując nieużywane strony pamięci na dysk. Wartość zerowa bitu użycia oznacza brak odwołań do strony i jest sygnałem dla systemu, że może być przeniesiona na dysk. Ustawiony bit użycia może w pewnym stopniu zabezpieczać stronę przed przeniesieniem na dysk.
- **Bit modyfikacji** – jest ustawiany podczas zapisu danych. Jeżeli dana strona jest usuwana z pamięci, to w przypadku ustawienia bitu modyfikacji musi być zapisana na dysk. Wartość zerowa bitu modyfikacji oznacza, że strona w pamięci oraz strona na dysku jest taka sama, więc podczas usuwania strony z pamięci nie ma konieczności ponownego zapisywania jej na dysk.

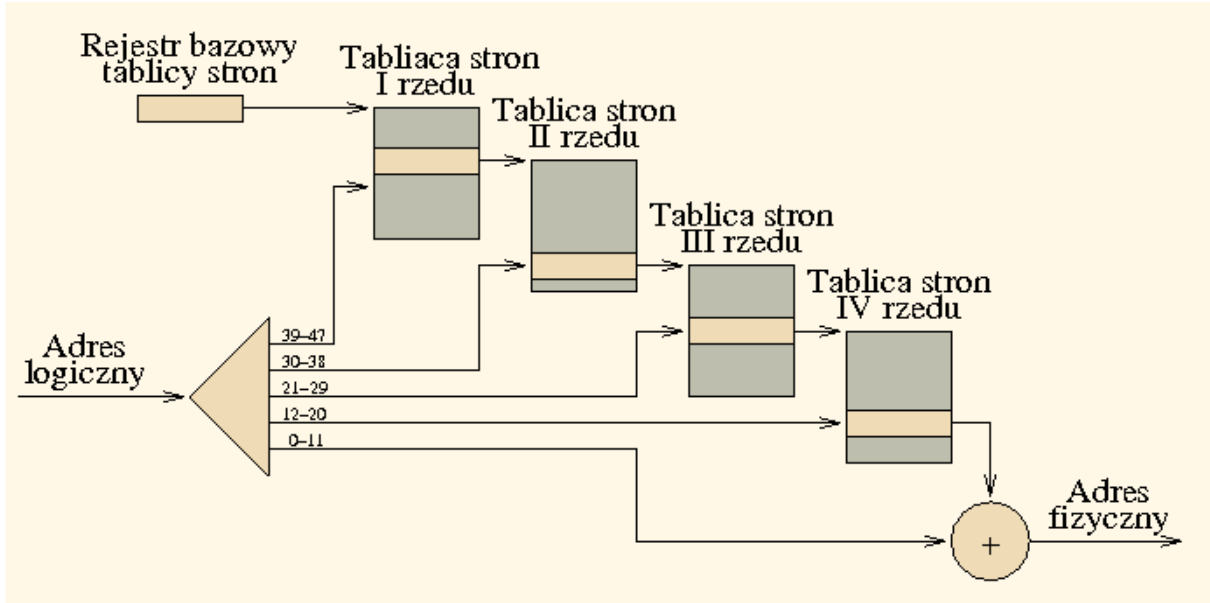
## **STRONICOWANIE WIELOPOZIOMOWE**

Logiczna przestrzeń adresowa na 32-bitowej maszynie z rozmiarem strony 4KB powoduje, że tablica stron może zawierać do miliona wpisów. Ponieważ każda pozycja w tablicy stron ma 4B, więc każdy proces może wymagać do 4MB fizycznej przestrzeni adresowej na samą tylko tablicę stron. Jednym z rozwiązań jest stronicowanie wielopoziomowe.

Idea stronicowania dwupoziomowego:

- ✓ Tablica stron podlega stronicowaniu
- ✓ Tablica stron pierwszego poziomu przechowuje numery stron w tablicy drugiego poziomu
- ✓ Każdej pozycji w tablicy stron pierwszego poziomu odpowiada tablica stron drugiego poziomu
- ✓ Tablica stron drugiego poziomu przechowuje numery stron

## PRZYKŁAD STRONICOWANIA WIELOPOZIOMOWEGO



### ZALETY STRONICOWANIA

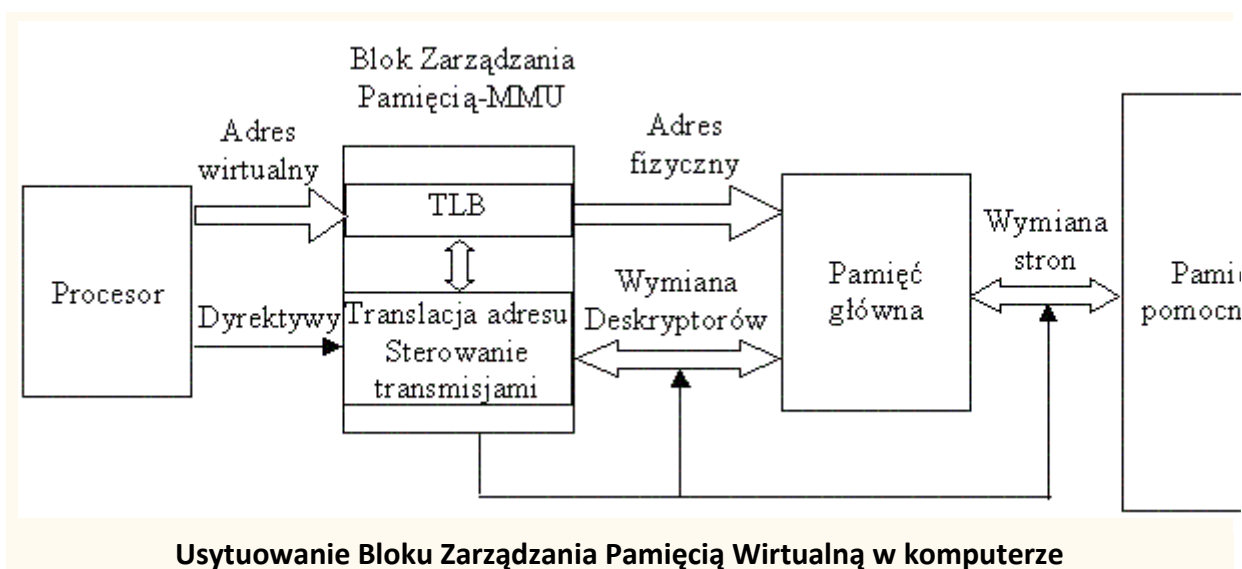
- Brak fragmentacji zewnętrznej
- Wspomaganie dla współdzielenia i ochrony pamięci
- Wyższy stopień wieloprogramowości
- Duża przestrzeń adresów wirtualnych

### WADY STRONICOWANIA

- Narzut czasowy przy transformacji adresu
- Narzut pamięciowy (na potrzeby tablicy stron)
- Fragmentacja wewnętrzna (niewielka)

Obsługa wirtualnej pamięci stronicowanej wymaga wykonywania dużej liczby operacji. Szczególnie niekorzystne jest tutaj obciążanie czasu dostępu do danych i rozkazów czasem na dodatkowy dostęp do pamięci operacyjnej, związany z odczytem deskryptorów stron. Aby zredukować te dodatkowe nakłady czasu, w nowoczesnych mikroprocesorach wprowadzono odpowiednik pamięci podręcznej na deskryptory ostatnio wykorzystywanych stron. Jest to bardzo szybka pamięć nazywana **buforem translacji adresów** (ang. **translation look-aside buffer - TLB**). W czasie translacji adresów sprawdzana jest najpierw obecność deskryptora stron w buforze translacji adresów a dopiero przy negatywnym wyniku testu, podejmowane jest odczytanie i sprowadzenie do TLB deskryptora z pamięci operacyjnej.

Dalszym krokiem w kierunku usprawnienia translacji adresów jest wprowadzenie w nowoczesnych mikroprocesorach wydzielonego sprzętowego **bloku zarządzania pamięcią** (ang. **memory management unit - MMU**). Blok zarządzania pamięcią działa w oparciu o bufor translacji adresów oraz rejestry, w których są przechowywane bieżące wskaźniki do wszelkich tablic wykorzystywanych przy konwersji adresów wirtualnych na fizyczne.



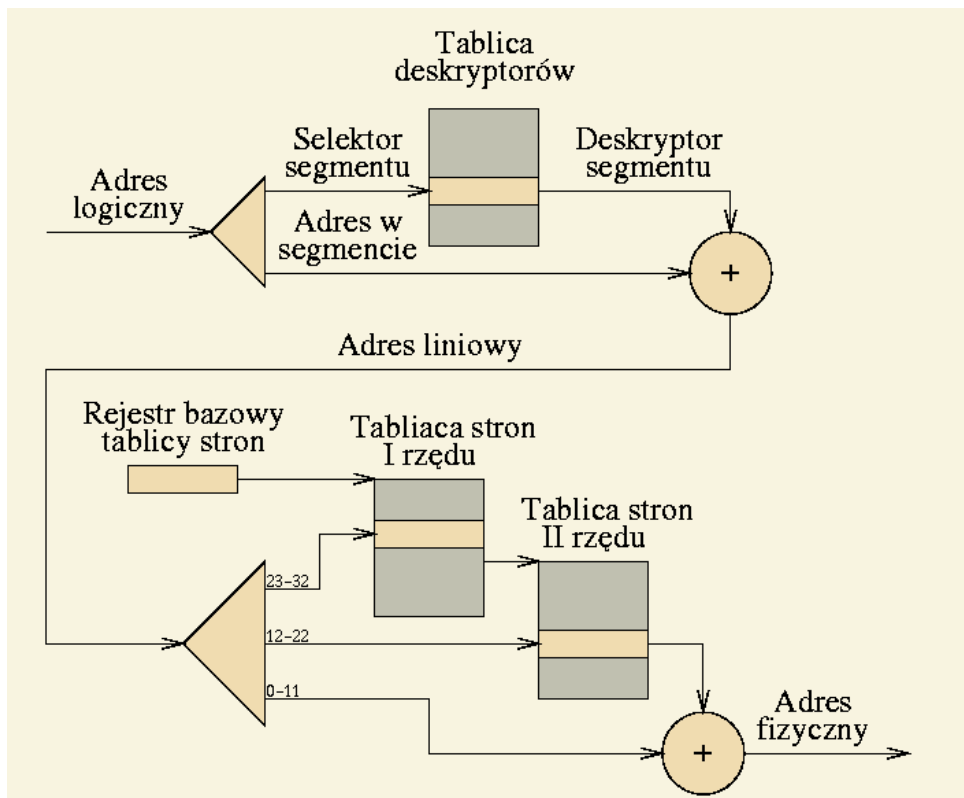
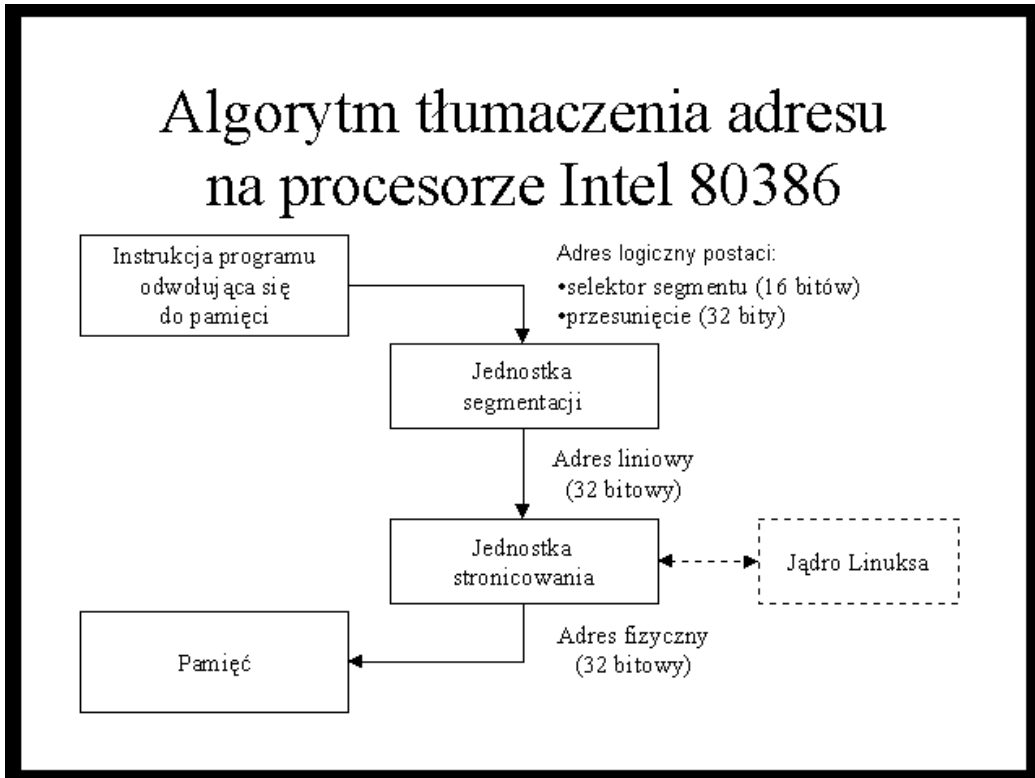
Blok MMU sprawdza czy deskryptor strony dla danego adresu wirtualnego jest w TLB. Jeśli tak, to MMU generuje adres fizyczny słowa dla pamięci operacyjnej. Jeśli brak deskryptora w TLB, to MMU sprowadza deskryptor z pamięci operacyjnej, zarządzając przy tym szyną tej pamięci, i uaktualnia tablicę TLB. Następnie w zależności od tego czy dana strona rezyduje w pamięci operacyjnej czy nie, MMU wykonuje translację adresu na adres rzeczywisty albo zarządza wczytanie strony do pamięci operacyjnej z pamięci pomocniczej.

### SEGMENTACJA ZE STRONICOWANIEM

Segmentacja ze stronicowaniem stanowi połączenie technik segmentacji i stronicowania. Z punktu widzenia procesu, pamięć logiczna składa się z szeregu segmentów. Jednak poszczególne segmenty nie są rozmieszczone w pamięci fizycznej w spójny sposób. Są one dzielone na strony i każda strona może być pamiętana w dowolnej ramce. Tak więc adres logiczny składa się z numeru segmentu i adresu w obrębie segmentu. Na podstawie numeru segmentu można wyznaczyć tablicę stron określającą rozmieszczenie stron tworzących segment.

Podobnie jak w przypadku stronicowania, możemy mieć do czynienia ze stronicowaniem wielopoziomowym, jak również z podręczną tablicą stron.

Dzięki połączeniu technik segmentacji i stronicowania, możemy z jednej strony przydzielać procesom wiele segmentów, czy współdzielić segmenty, a z drugiej strony unikamy fragmentacji zewnętrznej.



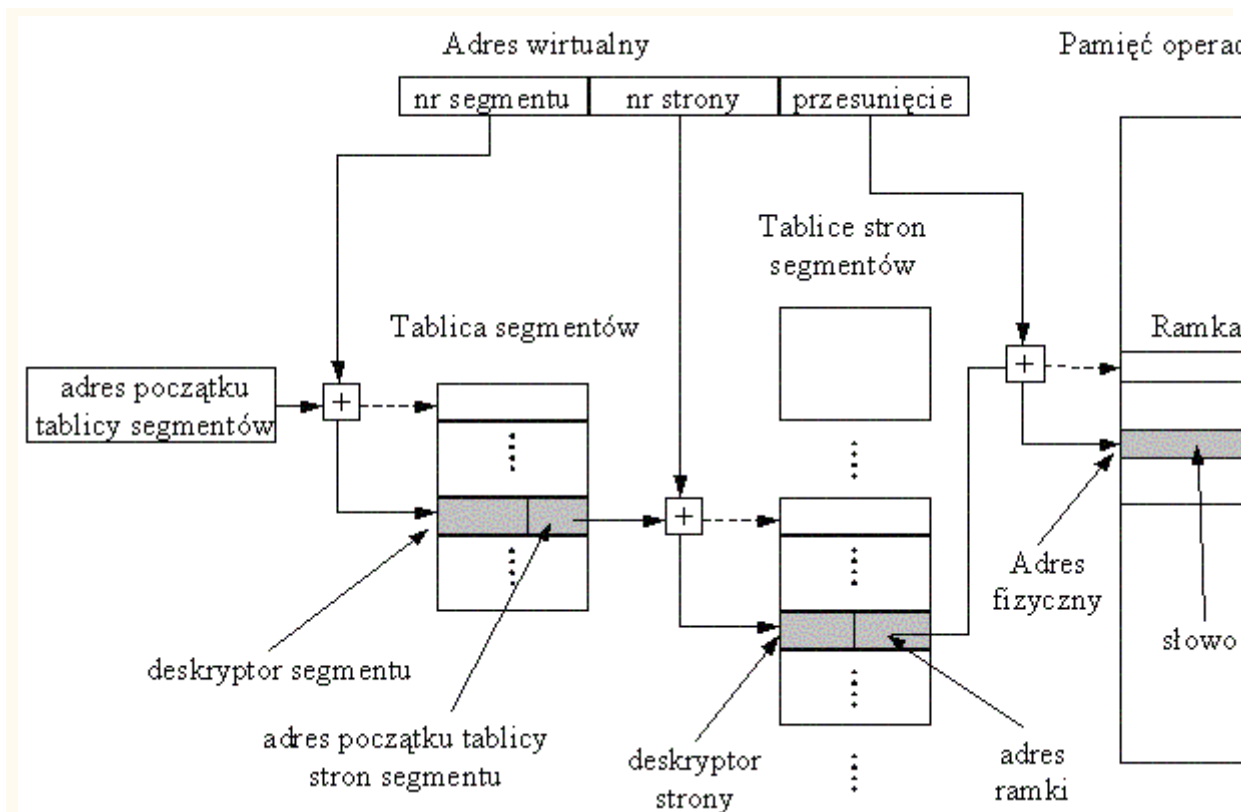
W przypadku tego procesora, segment jest identyfikowany przez tzw. selektor. Selektor jest tłumaczony na tzw. deskryptor, który po dodaniu do adresu w obrębie segmentu tworzy logiczny adres liniowy. Adres ten jest następnie tłumaczony na adres fizyczny poprzez dwustopniową tablicę stron. Adres tablicy stron pierwszego rzędu jest przechowywany w specjalnym rejestrze procesora, tzw. rejestrze bazowym tablicy stron.

## ZALETY SEGMENTACJI ZE STRONICOWANIEM

- Eliminacja fragmentacji – rozmieszczenie segmentów w różnych obszarach pamięci operacyjnej oraz fakt, iż segment nie musi być ciągły
- Wirtualność pamięci – całkowita długość programu może przekraczać długość pamięci fizycznej
- Możliwość współdzielenia obszarów pamięci (wykorzystanie segmentów przez różnych użytkowników)
- Możliwość ochrony danych

## WADY SEGMENTACJI ZE STRONICOWANIEM

- Wzrost kosztów sprzętu (np. dodatkowe rejestry)
- Zużycie czasu procesora na obsługę tablic oraz odwzorowania adresów logicznych na fizyczne
- Trudniejsza kontrola poprawności kodu



### Translacja adresu pamięci wirtualnej z segmentacją i stronicowaniem

Bity sterujące	Bity ochronv	Długość segmentu (w stronach)	Adres tablicy stron segmentu
----------------	--------------	-------------------------------	------------------------------

**Budowa deskryptora segmentu w pamięci segmentowanej ze stronicowaniem.**

## Ochrona procesora

Aby system operacyjny mógł wykonywać swe funkcje powinien mieć on dostęp do wszystkich istotnych zasobów procesora, mechanizmu zarządzania pamięcią, kontrolera przerwań i kontrolerów wejścia wyjścia.

Procesy aplikacyjne nie mogą mieć dostępu do tego typu zasobów, gdyż czy to na skutek błędów czy intencjonalnie mogłyby zdestabilizować pracę systemu.

We współczesnych mikroprocesorach wprowadza się dwa (lub więcej) **tryby pracy procesora**:

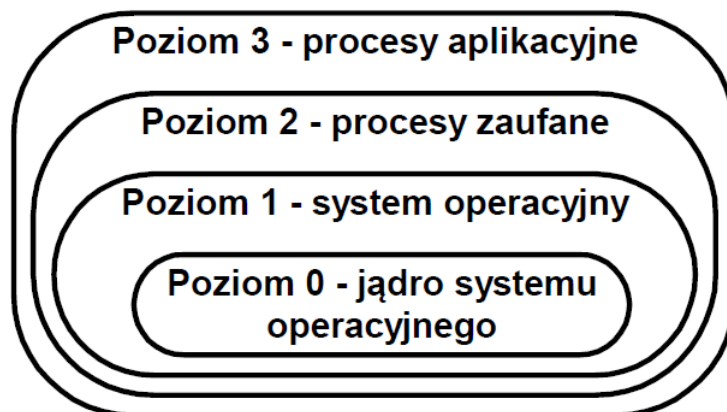
- **tryb użytkownika** (ang. *User Mode*)
- **tryb systemowy** (ang. *System Mode*).

W trybie systemowym proces może wykonywać wszystkie instrukcje procesora, sięgać do wszystkich obszarów pamięci i przestrzeni wejścia wyjścia.

W trybie użytkownika nie jest dozwolony dostęp do rejestrów związanych z zarządzaniem pamięcią, obsługą przerwań zarządzaniem pracą procesora.

## Poziomy uprzywilejowania

W mikroprocesorach o architekturze IA-32 ochrona procesora oparta jest o koncepcję poziomów ochrony (ang. *Privilege Level*).



**Poziomy ochrony w mikroprocesorach Intel**

Wykonywany proces pobiera instrukcje z segmentu wskazywanego przez rejestr CS. Rejestru CS nie da się przeładować inaczej niż przez instrukcję ICALL (IRET) lub przerwanie INT (IRET).

Używa on danych wskazywanych przez DS lub stosu wskazywanego przez SS.

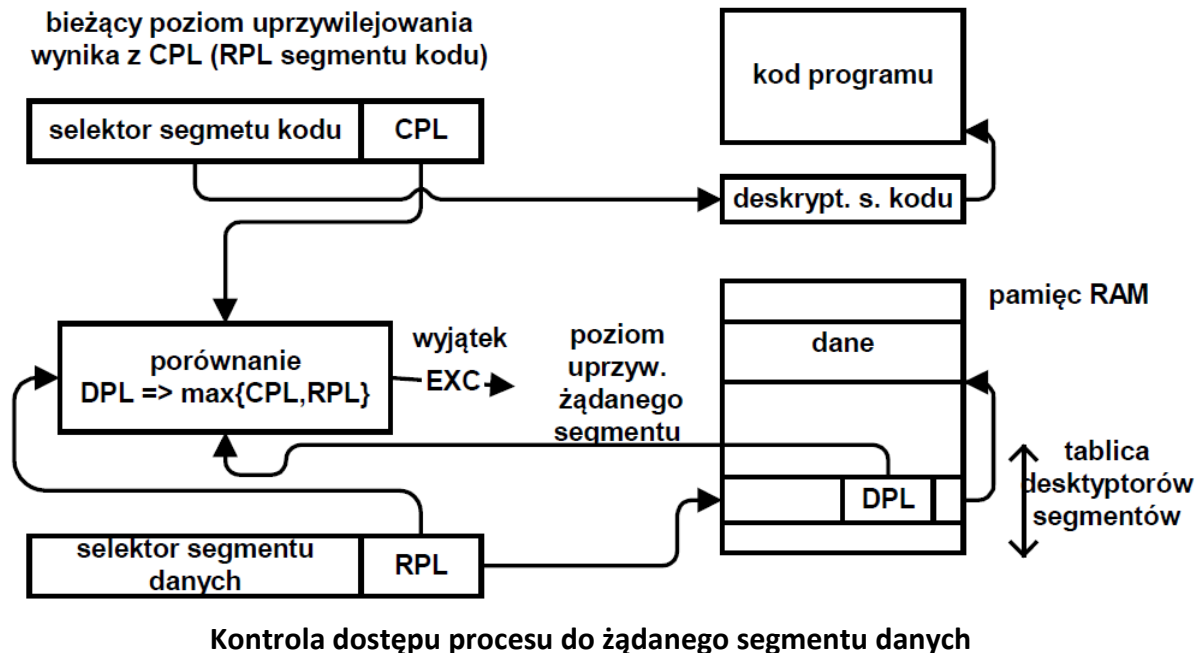
Wykonywany w danej chwili proces charakteryzuje się aktualnym poziomem ochrony CPL (ang. *Current Privilege Level*) uzyskiwanym z dwóch najmłodszych bitów rejestru CS.

Operacja dostępu do danych używa jakiegoś selektora (np. DS.) który zawiera pole RPL (ang. *Requested Privilege Level*), są to 2 najmłodsze bity.

Każdy segment pamięci (w tym danych) ma przyporządkowany poziom ochrony zapamiętany w polu DPL (ang. *Descriptor Privilege Level*) deskryptora segmentu do którego odnosi się żądanie.

Procesor ocenia prawa dostępu do segmentu poprzez porównanie obecnego poziomu uprzywilejowania CPL (wynika on z pola CPL), pola RPL selektora segmentu operandu i poziomu uprzywilejowania jaki ma deskryptor segmentu do którego przyznany ma być dostęp. Wartość CPL równa jest RPL z rejestru CS (selektor segmentu kodu).

- Przyznawany jest dostęp do danych o tym samym lub niższym poziomie ochrony (mniej ważnych).
- Dopuszczane jest wywoływanie procedur o tym samym lub wyższym poziomie ochrony (bardziej godnych zaufania)



## Instrukcje systemowe

Wśród wszystkich instrukcji procesora wyróżnia się instrukcje zarezerwowane dla systemu. Są to:

- instrukcje dostępu do rejestrów systemowych,
- ładowania tablic deskryptorów,
- zatrzymywanie procesora,
- zmiany niektórych flag.

Instrukcje systemowe mogą być wykonywane tylko w trybie uprzywilejowania  $CPL = 0$ . Gdy  $CPL$  procesu wykonującego instrukcję systemową jest różny od zera zostanie wygenerowany wyjątek.