Wydział Elektryczny
Katedra Elektrotechniki Teoretycznej i Metrologii

Materiały do wykładu z przedmiotu:
Informatyka
Kod: EDS1B1007

WYKŁAD NR 14 (pozostałe slajdy)

Opracował: dr inż. Jarosław Forenc

Białystok 2020

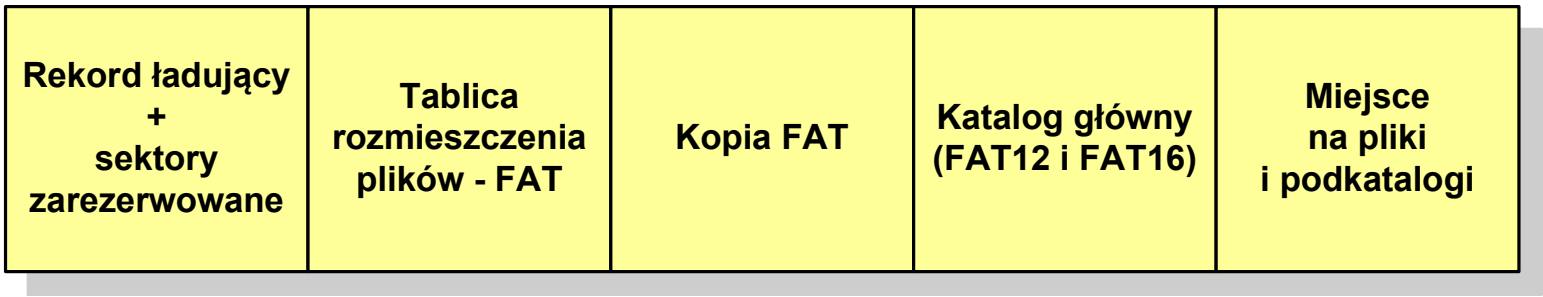
Materiały zostały opracowane w ramach projektu „PB2020 - Zintegrowany Program Rozwoju Politechniki Białostockiej” realizowanego w ramach Działania 3.5 Programu Operacyjnego Wiedza, Edukacja, Rozwój 2014-2020 współfinansowanego ze środków Europejskiego Funduszu Społecznego.

Plan wykładu nr 14

- Zarządzanie dyskowymi operacjami we-wy
  - systemy plików (FAT, NTFS, ext2)
- Zarządzanie pamięcią operacyjną
  - proste stronicowanie, prosta segmentacja
  - pamięć wirtualna, stronicowanie i segmentacja pamięci wirtualnej

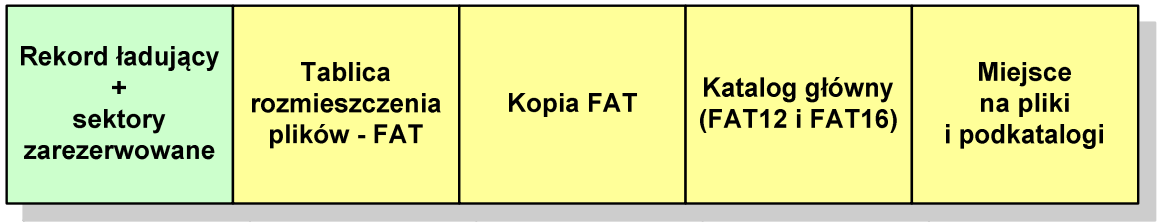
System plików FAT (File Allocation Table)

- opracowany na przełomie lat 70. i 80. dla systemu MS-DOS
- występuje w czterech wersjach: FAT12, FAT16, FAT32 i exFAT (FAT64)
- numer występujący po słowie FAT oznacza liczbę bitów przeznaczonych do kodowania (numeracji) jednostek alokacji pliku (JAP), tzw. klastrów (ang. cluster) w tablicy alokacji plików
  - 12 bitów w systemie FAT12
  - 16 bitów w systemie FAT16
  - 32 bity w systemie FAT32
  - 64 bity w systemie exFAT (FAT64)
- ogólna struktura dysku logicznego / dyskietki w systemie FAT:



FAT12

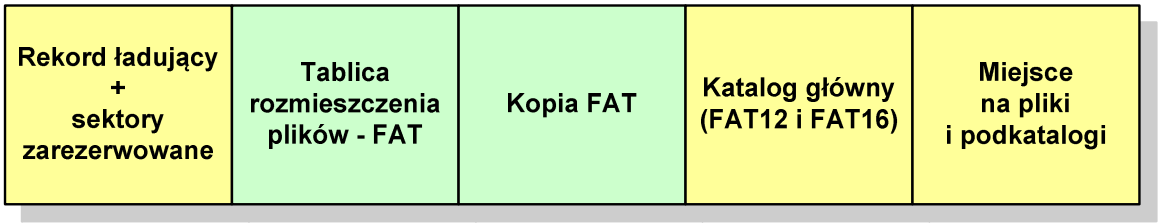
- system plików FAT12 przeznaczony jest dla nośników o małej pojemności
- rekord ładujący zajmuje pierwszy sektor dyskietki lub dysku logicznego



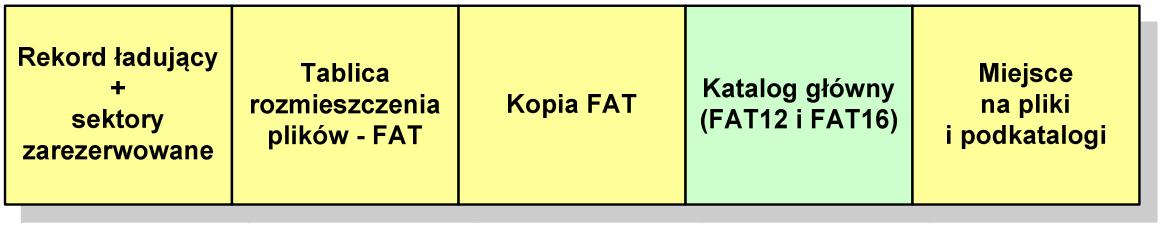
- rekord ładujący zawiera następujące dane:
  - instrukcja skoku do początku programu ładującego (3 bajty)
  - nazwa wersji systemu operacyjnego (8 bajtów)
  - struktura BPB (ang. BIOS Parametr Block) - blok parametrów BIOS (25 bajtów)
  - rozszerzony BPB (ang. Extended BPB, 26 bajtów)
  - wykonywalny kod startowy uruchamiający system operacyjny (448 bajtów)
  - znacznik końca sektora - 55AAH (2 bajty)

FAT12

- tablica rozmieszczenia plików FAT tworzy swego rodzaju „mapę” plików zapisanych na dysku
- za tablicą FAT znajduje się jej kopia, która nie jest wykorzystywana



- za kopią tablicy FAT znajduje się katalog główny zajmujący określoną dla danego typu dysku liczbę sektorów

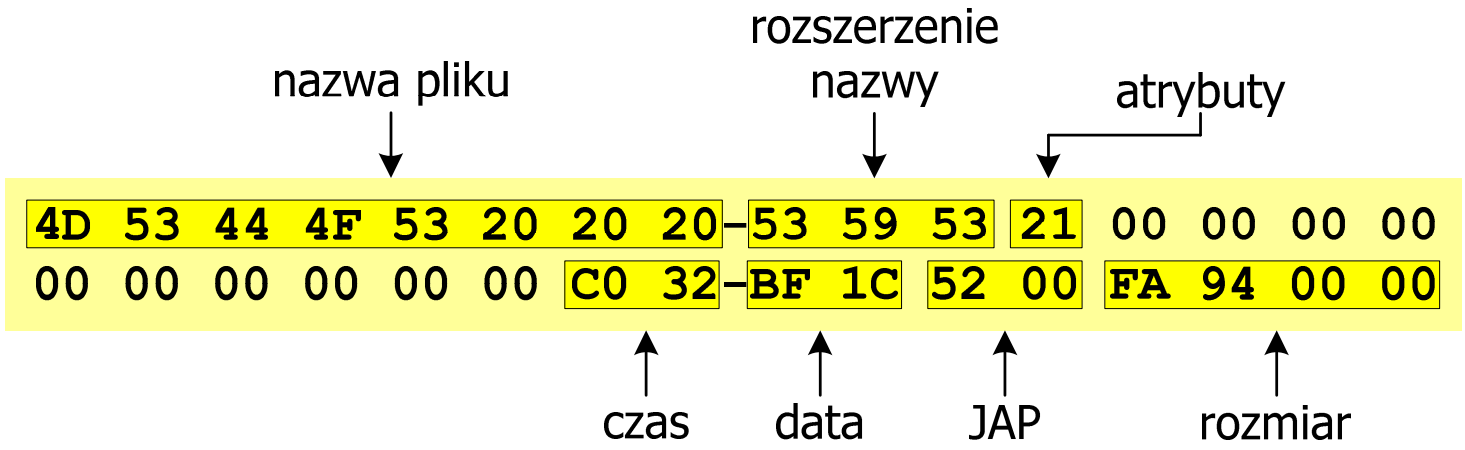


- katalog główny zawiera 32-bajtowe pola mogące opisywać pliki, podkatalogi lub etykietę dysku

FAT12

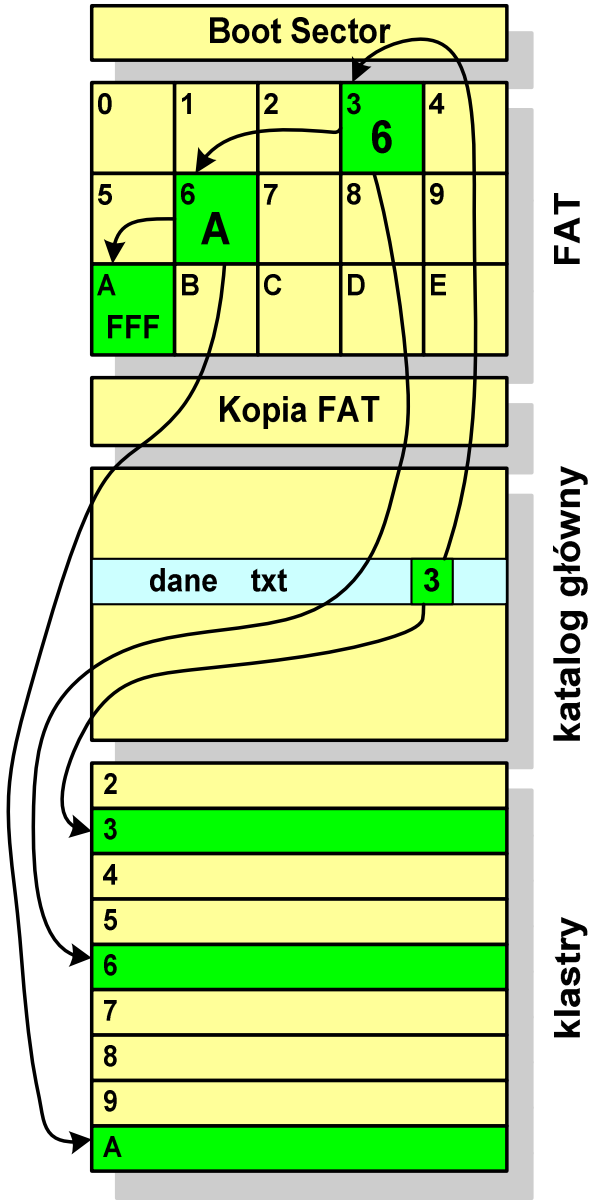
- przykładowa zawartość katalogu głównego:

```
0000  49 4F 20 20 20 20 20 20-53 59 53 21 00 00 00 00   IO      SYS!....
0010  00 00 00 00 00 00 C0 32-BF 1C 02 00 46 9F 00 00   .......2....F...
0020  4D 53 44 4F 53 20 20 20-53 59 53 21 00 00 00 00   MSDOS   SYS!....
0030  00 00 00 00 00 00 C0 32-BF 1C 52 00 FA 94 00 00   .......2..R.....
0040  43 4F 4D 4D 41 4E 44 20-43 4F 4D 20 00 00 00 00   COMMAND COM ....
0050  00 00 00 00 00 00 C0 32-BF 1C 9D 00 75 D5 00 00   .......2....u...
0060  41 54 54 52 49 42 20 20-45 58 45 20 00 00 00 00   ATTRIB  EXE ....
0070  00 00 00 00 00 00 C0 32-BF 1C 08 01 C8 2B 00 00   .......2.....+..
```



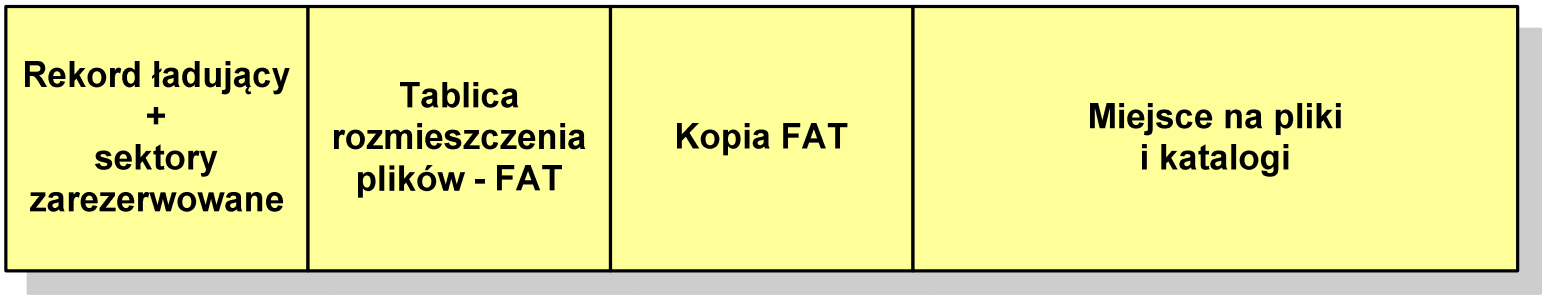
FAT12 - położenie pliku na dysku

- w katalogu, w 32-bajtowym polu każdego pliku wpisany jest początkowy numer JAP
- numer ten określa logiczny numer sektora, w którym znajduje się początek pliku
- ten sam numer JAP jest jednocześnie indeksem do miejsca w tablicy FAT, w którym wpisany jest numer kolejnej JAP
- numer wpisany we wskazanym miejscu tablicy rozmieszczenia plików wskazuje pierwszy sektor następnej części pliku i równocześnie położenie w tablicy FAT numeru następnej JAP
- w ten sposób tworzy się łańcuch, określający położenie całego pliku
- jeśli numer JAP składa się z samych FFF, to oznacza to koniec pliku



FAT32

- po raz pierwszy wprowadzony w systemie Windows 95 OSR2
- ogólna struktura systemu FAT32 jest taka sama jak w FAT12/FAT16 - nie ma tylko miejsca przeznaczonego na katalog główny
- w systemie FAT32 katalog główny może znajdować się w dowolnym miejscu na dysku i może zawierać maksymalnie 65 532 pliki i katalogi



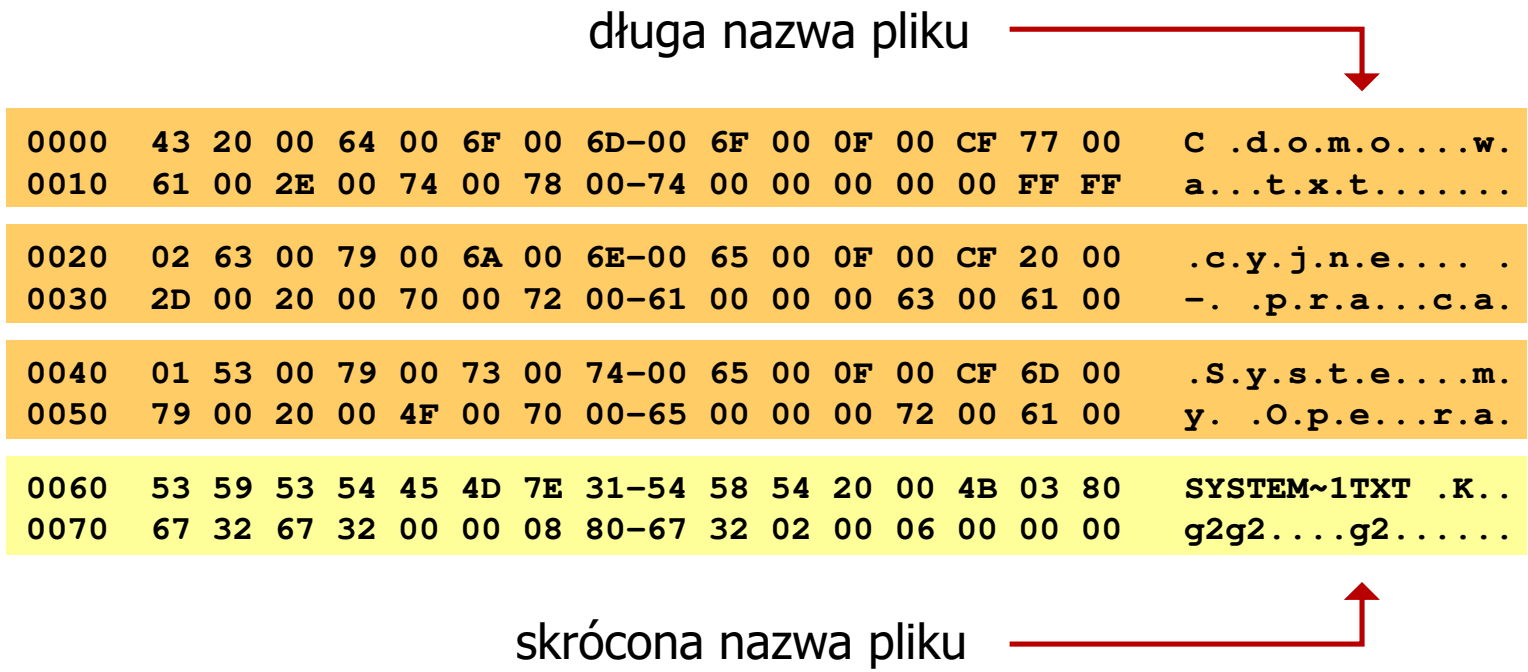
- do adresowania JAP stosuje się, obcięty o 4 najstarsze bity, adres 32-bitowy i dlatego dysk z FAT32 może zawierać maksymalnie 2^{28} JAP
- w systemie FAT32 można formatować tylko dyski, nie można natomiast zainstalować go na dyskietkach

FAT32 - długie nazwy plików

- wprowadzone w systemie Windows 95
- informacje o nazwie pliku zapamiętywane są jako:
  - długa nazwa
  - skrócona nazwa (tzw. alias długiej nazwy)
- skrócona nazwa pliku przechowywana jest w identycznej, 32-bajtowej, strukturze jak w przypadku plików w starym formacie 8+3
- długie nazwy plików zapisywane są także w 32-bajtowych strukturach, przy czym jedna nazwa zajmuje kilka struktur (w jednej strukturze umieszczonych jest 13 kolejnych znaków w formacie Unicode)

FAT32 - długie nazwy plików

- Nazwa pliku: Systemy Operacyjne - praca domowa.txt

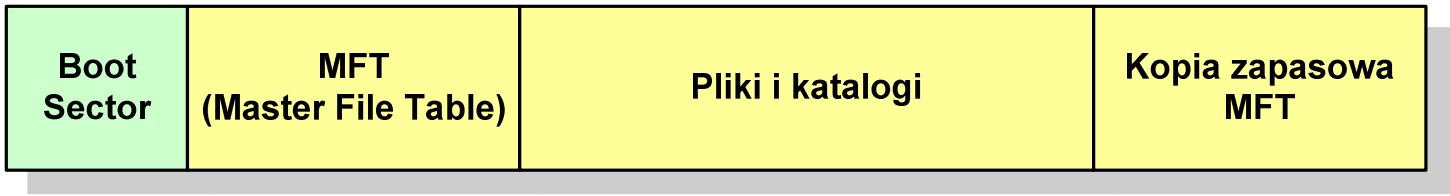


exFAT (FAT64)

- po raz pierwszy pojawił się w listopadzie 2006 roku w Windows Embedded CE 6.0 i Windows Vista SP1
- obsługiwany także przez Windows 7/8/10, Windows Server 2003/2008, Windows XP SP2/SP3, Linux
- stworzony przez Microsoft na potrzeby pamięci Flash
- podstawowe cechy:
  - maksymalna wielkość pliku to 2^{64} = 16 EB
  - maksymalna wielkość klastra - do 32 MB
  - nieograniczona liczba plików w pojedynczym katalogu
  - prawa dostępu do plików i katalogów

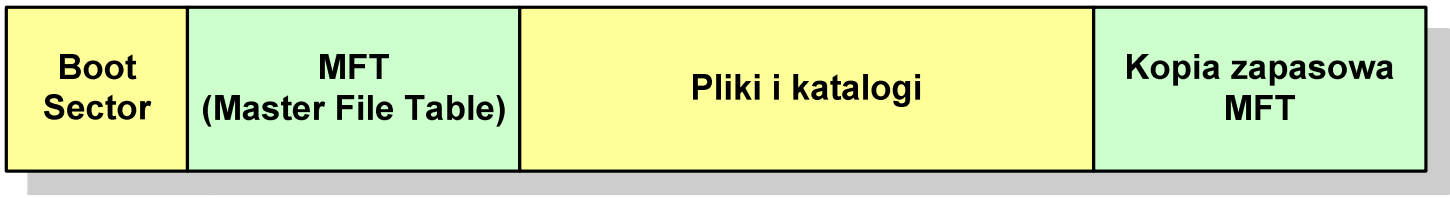
NTFS (New Technology File System)

- wersja 1.0 (połowa 1993 r.) - Windows NT 3.1
- wersja 3.1 (NTFS 5.1) - Windows XP/Server 2003/Vista/7/8/10
- struktura wolumenu (dysku) NTFS:



- Boot Sector rozpoczyna się od zerowego sektora partycji, może zajmować 16 kolejnych sektorów, zawiera podobne dane jak w systemie FAT

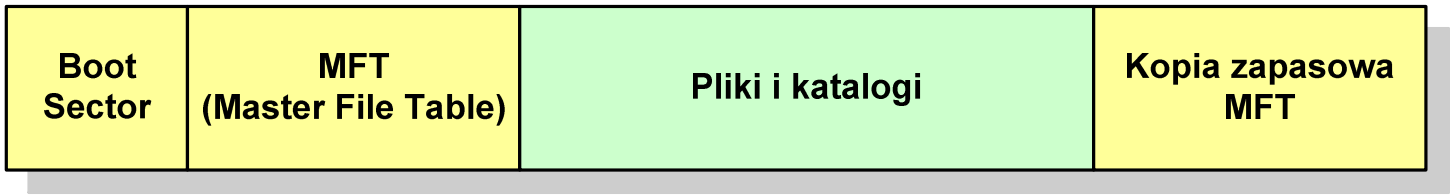
NTFS



- MFT (Master File Table) - specjalny plik, niewidoczny dla użytkownika, zawiera wszystkie dane niezbędne do odczytania pliku z dysku, składa się z rekordów o stałej długości (1 kB - 4 kB)
- pierwsze 16 (NTFS 4) lub 26 (NTFS 5) rekordów jest zarezerwowane dla tzw. metaplików, np.
  - rekord nr: 0 plik: $Mft (główna tablica plików)
  - rekord nr: 1 plik: $MftMirr (główna tablica plików 2)
  - rekord nr: 5 plik: $ (indeks katalogu głównego)
- pozostała część pliku MFT przeznaczona jest na rekordy wszystkich plików i katalogów umieszczonych na dysku

NTFS

- struktura wolumenu (dysku) NTFS:



- plik w NTFS to zbiór atrybutów
- wszystkie atrybuty mają dwie części składowe: nagłówek i blok danych
- nagłówek opisuje atrybut, np. liczbę bajtów zajmowanych przez atrybut, rozmiar bloku danych, położenie bloku danych, znacznik czasu
- bloku danych zawiera informacje zgodne z przeznaczeniem atrybutu

NTFS - Pliki

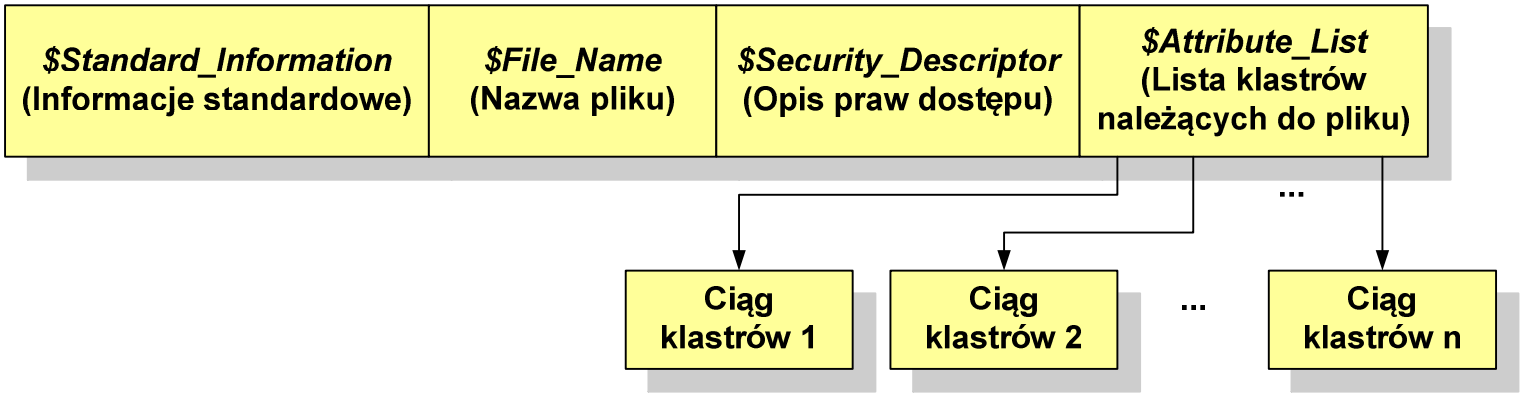
- pliki w systemie NTFS są reprezentowane w MFT przez rekord zawierający atrybuty:
  - $Standard_Information
  - $File_Name
  - $Security_Descriptor
  - $Data

$Standard_Information **(Informacje standardowe)**	***$File_Name*** **(Nazwa pliku)**	***$Security_Descriptor*** **(Opis praw dostępu)**	***$Data*** **(Dane)**

- w przypadku małych plików wszystkie jego atrybuty zapisywane są bezpośrednio w MFT (atrybuty rezydentne)

NTFS - Pliki

- jeśli atrybuty pliku są duże (najczęściej dotyczy to atrybutu $Data), to w rekordzie w MFT umieszczany jest tylko nagłówek atrybutu oraz wskaźnik do jego bloku danych, a sam blok danych przenoszony jest na dysk poza MFT (atrybuty nierezydentne)
- blok danych atrybutu nierezydentnego zapisywany jest w przyległych klastrach
- jeśli nie jest to możliwe, to dane zapisywane są w kilku ciągach jednostek alokacji i wtedy każdemu ciągowi odpowiada wskaźnik w rekordzie MFT



NTFS - Katalogi

- katalogi reprezentowane są przez rekordy zawierające trzy takie same atrybuty jak pliki:
  - $Standard_Information
  - $File_Name
  - $Security_Descriptor

$Standard_Information (Informacje standardowe)	*$File_Name* (Nazwa pliku)	*$Security_Descriptor* (Opis praw dostępu)	*$Index_Root*	*$Index_Allocation*	*$Bitmap*

- zamiast atrybutu $Data umieszczone są trzy atrybuty przeznaczone do tworzenia list, sortowania oraz lokalizowania plików i podkatalogów
  - $Index_Root
  - $Index_Allocation
  - $Bitmap

ext2

- pierwszy system plików w Linuxie: Minix (14-znakowe nazwy plików i maksymalny rozmiar wynoszący 64 MB)
- system Minix zastąpiono nowym systemem nazwanym rozszerzonym systemem plików - ext (ang. extended file system), a ten, w styczniu 1993 r., systemem ext2 (ang. second extended file system)
- w systemie ext2 podstawowym elementem podziału dysku jest blok
- wielkość bloku jest stała w ramach całego systemu plików, określana na etapie jego tworzenia i może wynosić 1024, 2048 lub 4096 bajtów
- w celu zwiększenia bezpieczeństwa i optymalizacji zapisu na dysku posługujemy się nie pojedynczymi blokami, a grupami bloków

ext2

- w każdej grupie bloków znajduje się kopia tego samego bloku identyfikacyjnego oraz kopia bloków z deskryptorami wszystkich grup
- blok identyfikacyjny zawiera informacje na temat systemu plików (rodzaj systemu plików, rozmiar bloku, czas dokonanej ostatnio zmiany , …)
- w deskryptorach grupy znajdują się informacje na temat grupy bloków (numer bloku z bitmapą zajętości bloków grupy, numer bloku z bitmapą zajętości i-węzłów, numer pierwszego bloku z tablicą i-węzłów, liczba wolnych bloków, liczba katalogów w grupie)

ext2

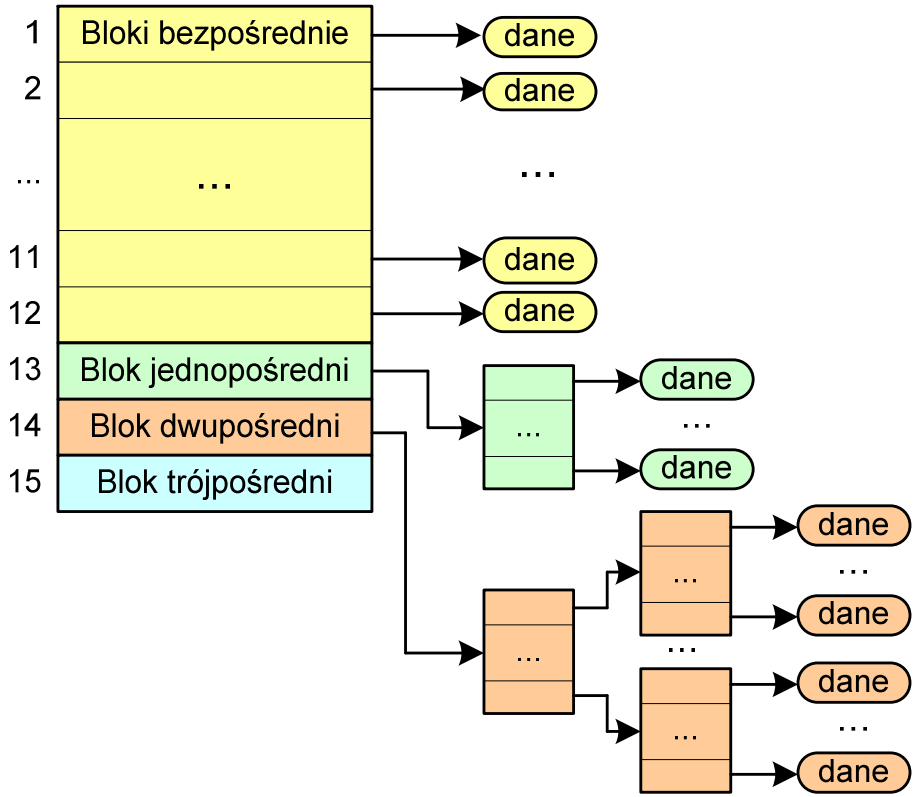
- blok z mapą bitową zajętości bloków danej grupy jest tablicą bitów o rozmiarze jednego bloku
  - jeśli blok ma rozmiar 1 kB to pojedynczą mapą można opisać fizyczna grupę 8096 bloków czyli 8 MB danych
  - jeśli natomiast blok ma rozmiar 4 kB, to fizyczna grupa bloków zajmuje 128 MB danych
- przed tablicą i-węzłów znajduje się blok z mapa bitową zajętości i-węzłów danej grupy - jest to tablica bitów, z których każdy zawiera informację czy dany i-węzeł jest wolny czy zajęty

ext2 - i-węzeł

- pliki na dysku reprezentowane są przez i-węzły (ang. i-node)
- każdemu plikowi odpowiada dokładnie jeden i-węzeł, który jest strukturą zawierającą m.in. następujące pola:
  - numer i-węzła w dyskowej tablicy i-węzłów
  - typ pliku: zwykły, katalog, łącze nazwane, specjalny, znakowy
  - prawa dostępu do pliku: dla wszystkich, grupy, użytkownika
  - liczba dowiązań do pliku
  - identyfikator właściciela pliku
  - identyfikator grupy właściciela pliku
  - rozmiar pliku w bajtach (max. 4 GB)
  - czas utworzenia pliku
  - czas ostatniego dostępu do pliku
  - czas ostatniej modyfikacji pliku
  - liczba bloków dyskowych zajmowanych przez plik

ext2 - i-węzeł

- położenie pliku na dysku określają w i-węźle pola:
  - 12 adresów bloków zawierających dane (w systemie Unix jest ich 10) - bloki bezpośrednie
  - 1 adres bloku zawierającego adresy bloków zawierających dane - blok jednopośredni (ang. single indirect block)
  - 1 adres bloku zawierającego adresy bloków jednopośrednich - blok dwupośredni (ang. double indirect block)
  - 1 adres bloku zawierającego adresy bloków dwupośrednich - blok trójpośredni (ang. triple indirect block)



ext2

- nazwy plików przechowywane są w katalogach, które w systemie Linux są plikami, ale o specjalnej strukturze
- katalogi składają się z ciągu tzw. pozycji katalogowych o nieustalonej z góry długości
- każda pozycja opisuje dowiązanie do jednego pliku i zawiera:
  - numer i-węzła (4 bajty)
  - rozmiar pozycji katalogowej (2 bajty)
  - długość nazwy (2 bajty)
  - nazwa (od 1 do 255 znaków)

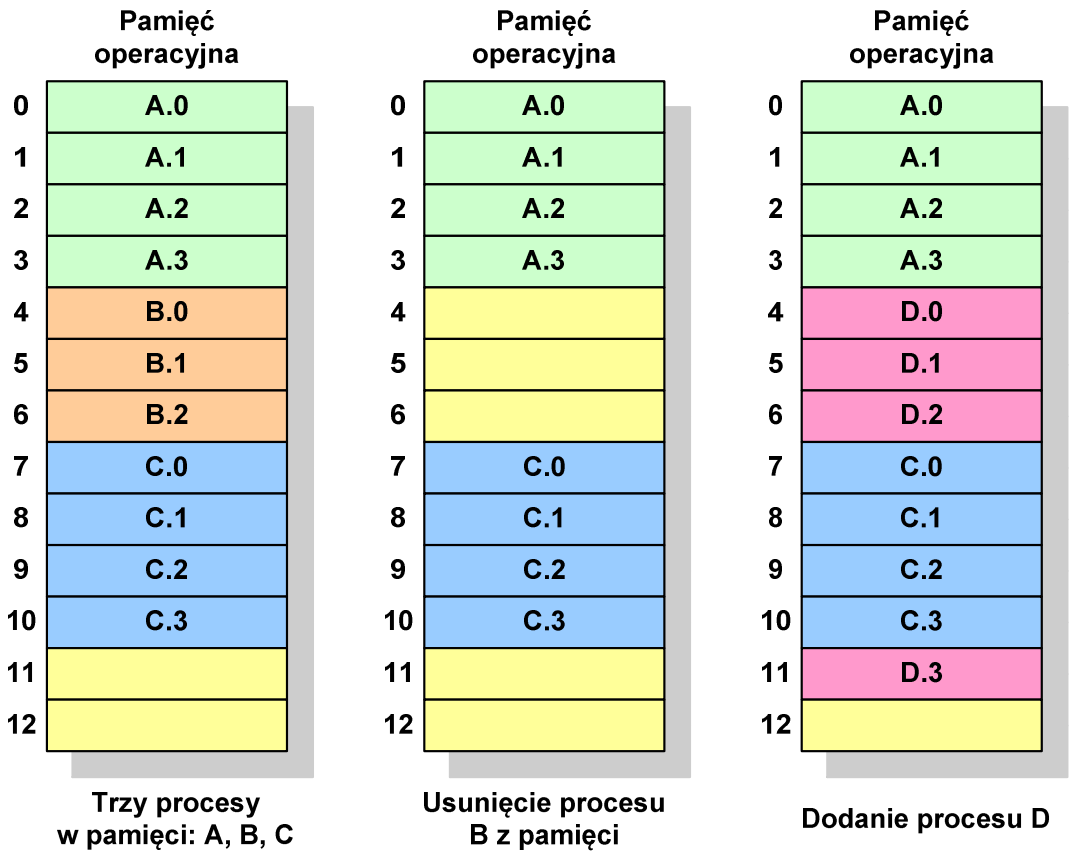
```
struct ext2_dir_entry
{
   _u32  inode                  /*  numer i-wezla                */
   _u16  rec_len                /*  dlugosc pozycji katalogowej  */
   _u16  name_len               /*  dlugosc nazwy                */
   char  name[EXT2_NAME_LEN]    /*  nazwa                        */
}
```

Zarządzanie pamięcią

- zarządzanie pamięcią polega na wydajnym przenoszeniu programów i danych do i z pamięci operacyjnej
- w nowoczesnych wieloprogramowych systemach operacyjnych zarządzanie pamięcią opiera się na pamięci wirtualnej
- pamięć wirtualna bazuje na wykorzystaniu segmentacji i stronicowania
- z historycznego punktu widzenia w systemach komputerowych stosowane były/są następujące metody zarządzania pamięcią:
  - proste stronicowanie, prosta segmentacja
  - stronicowanie pamięci wirtualnej, segmentacja pamięci wirtualnej
  - stronicowanie i segmentacja pamięci wirtualnej

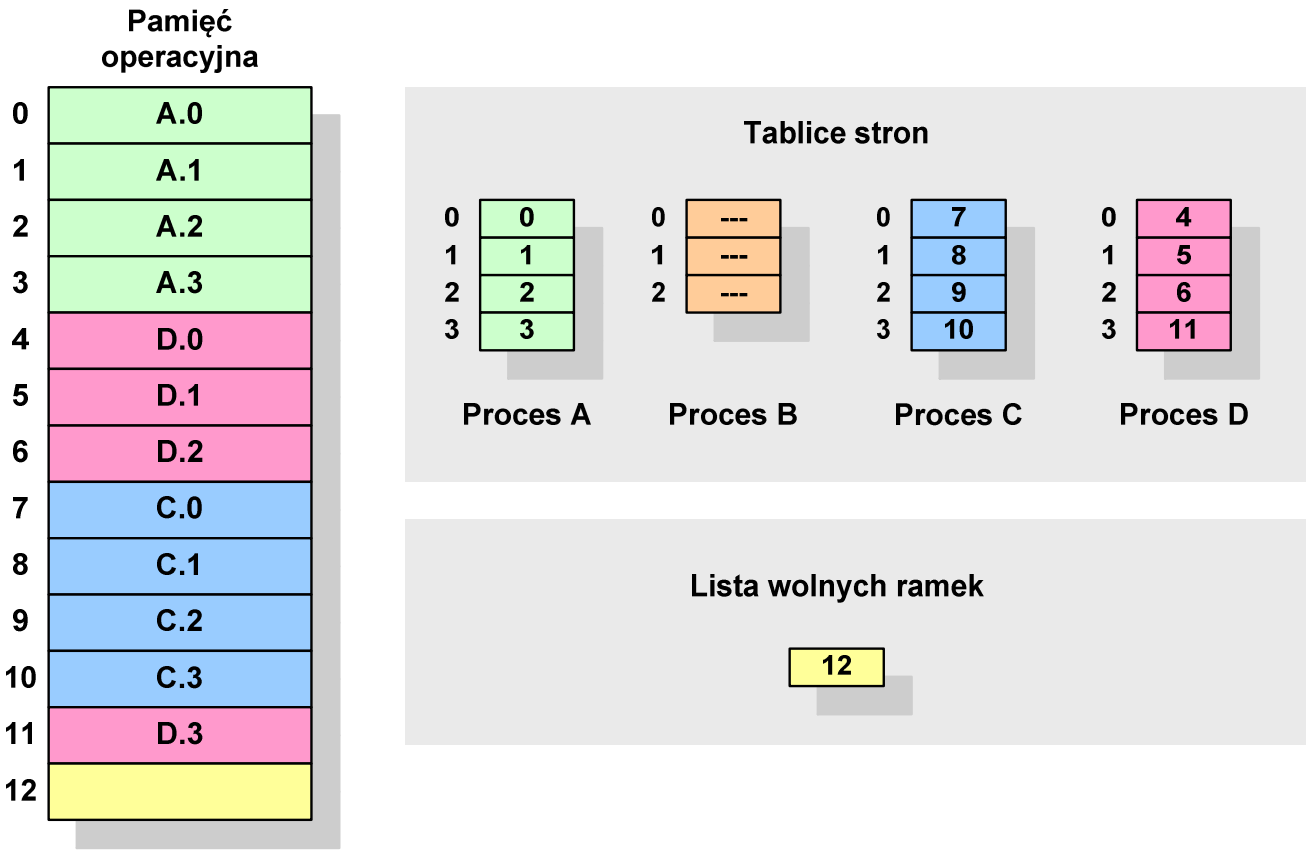
Proste stronicowanie

- pamięć operacyjna podzielona jest na jednakowe bloki o stałym niewielkim rozmiarze nazywane ramkami lub ramkami stron (page frames)
- do tych ramek wstawiane są fragmenty procesu zwane stronami (pages)
- aby proces mógł zostać uruchomiony wszystkie jego strony muszą znajdować się w pamięci operacyjnej



Proste stronicowanie

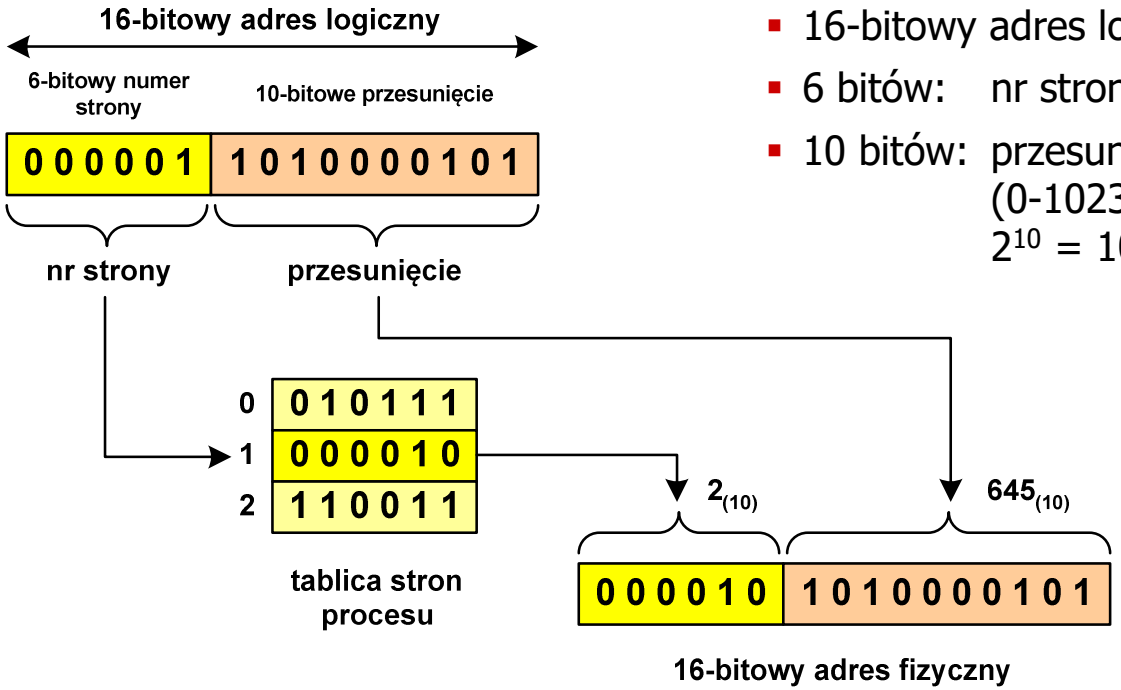
- dla każdego procesu przechowywana jest tablica strony (page table) zawierająca lokalizację ramki dla każdej strony procesu



Proste stronicowanie

- aby mechanizm stronicowania był wygodny ustala się, że rozmiar strony jest liczbą podniesioną do potęgi drugiej - dzięki temu adres względny oraz adres logiczny (numer strony + jej przesunięcie) są takie same

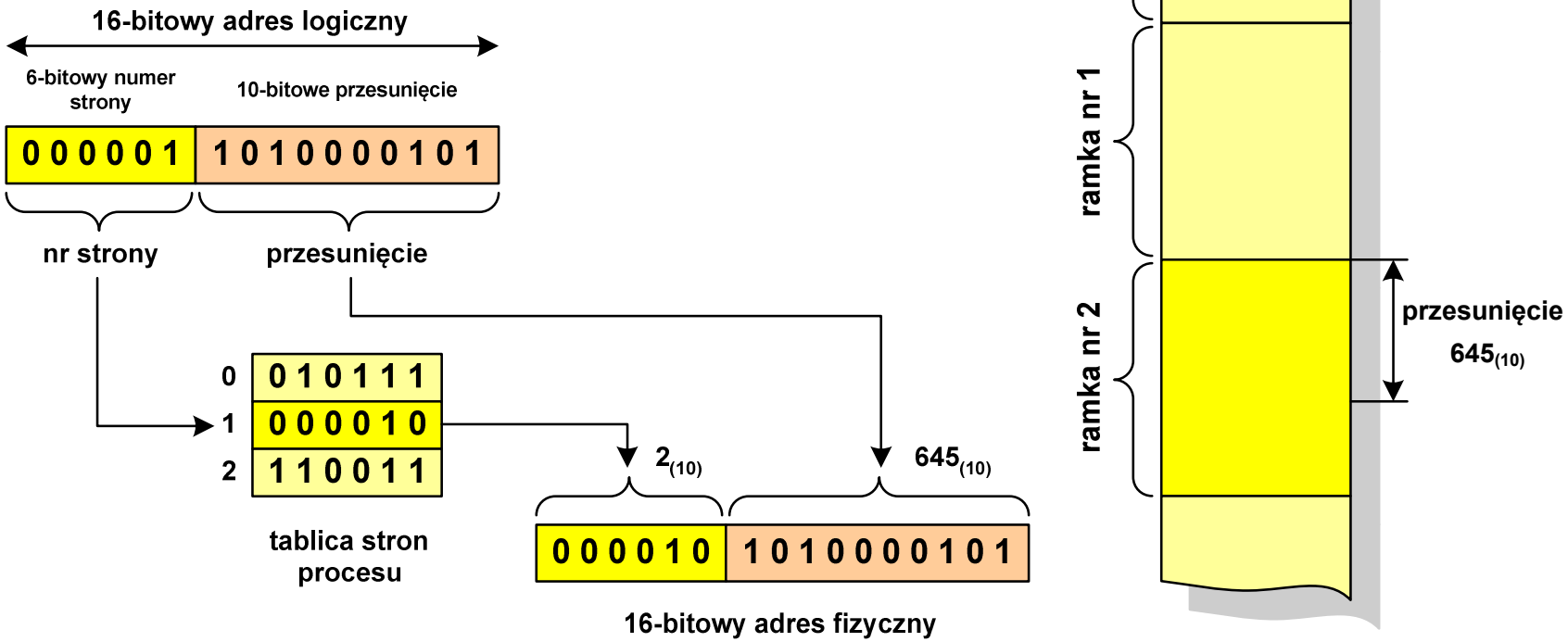
Przykład:



- 16-bitowy adres logiczny
- 6 bitów: nr strony (0-63), max. 2^6 = 64 strony
- 10 bitów: przesunięcie w ramach strony (0-1023), rozmiar strony wynosi: 2^{10} = 1024 bajty = 1 kB

Proste stronicowanie

- zalety: brak fragmentacji zewnętrznej, stronicowanie nie jest widoczne dla programisty
- wady: niewielki stopień fragmentacji wewnętrznej



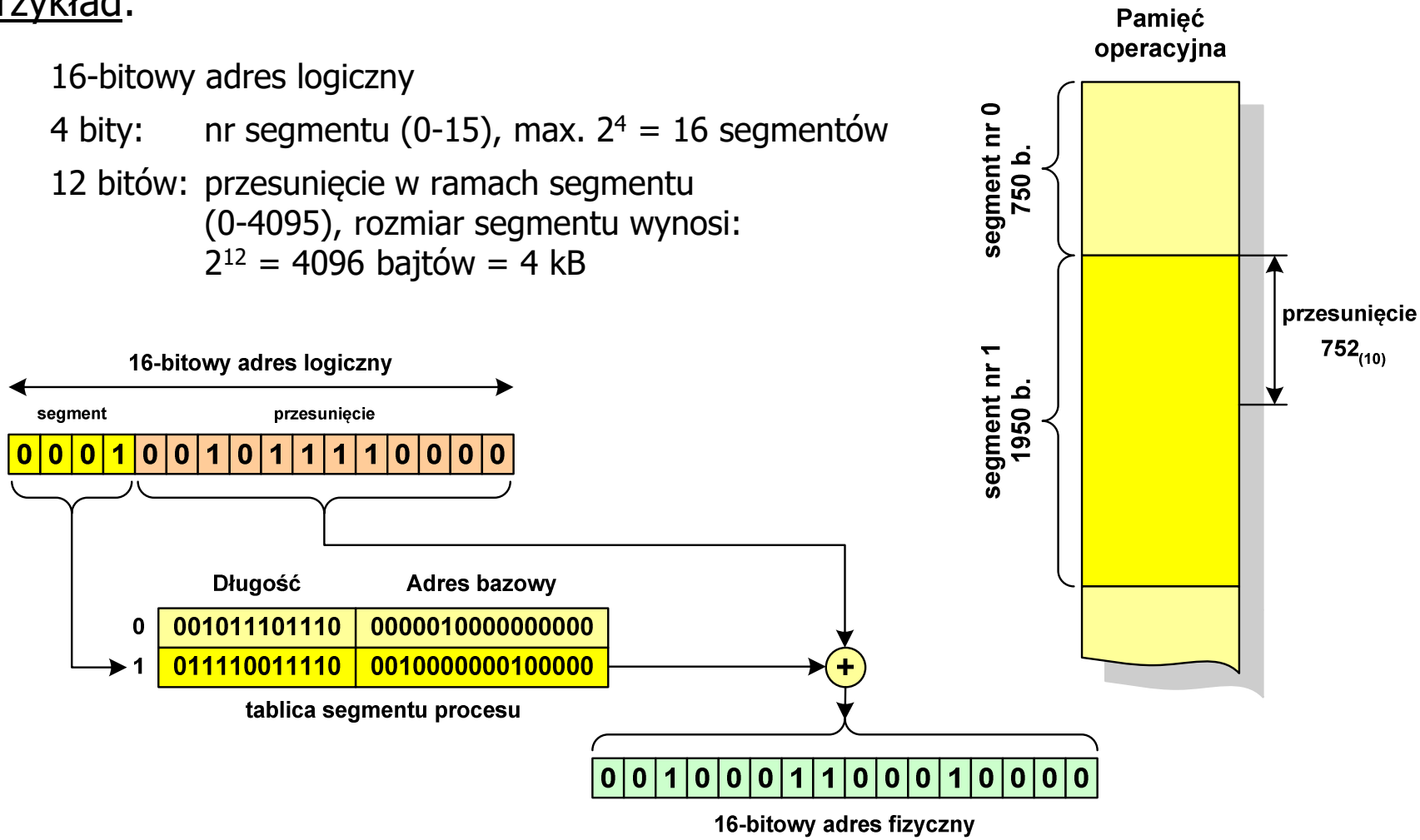
Prosta segmentacja

- polega na podzieleniu programu i skojarzonych z nim danych na odpowiednią liczbę segmentów o różnej długości
- ładowanie procesu do pamięci polega na wczytaniu wszystkich jego segmentów do partycji dynamicznych (nie muszą być ciągłe)
- segmentacja jest widoczna dla programisty i ma na celu wygodniejszą organizację programów i danych
- adres logiczny wykorzystujący segmentację składa się z dwóch części:
  - numeru segmentu
  - przesunięcia
- dla każdego procesu określana jest tablica segmentu procesu zawierająca:
  - długość danego segmentu
  - adres początkowy danego segmentu w pamięci operacyjnej

Prosta segmentacja

Przykład:

- 16-bitowy adres logiczny
- 4 bity: nr segmentu (0-15), max. 2^4 = 16 segmentów
- 12 bitów: przesunięcie w ramach segmentu (0-4095), rozmiar segmentu wynosi: 2^{12} = 4096 bajtów = 4 kB



Pamięć wirtualna

- pamięć wirtualna umożliwia przechowywanie stron/segmentów wykonywanego procesu w pamięci dodatkowej (na dysku twardym)

Co się dzieje, gdy procesor chce odczytać stronę z pamięci dodatkowej?

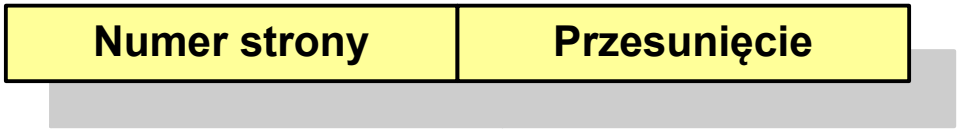
- generowanie przerwania sygnalizującego błąd w dostępie do pamięci
- zmiana stan procesu na zablokowany
- wstawienie do pamięci operacyjnej fragment procesu zawierający adres logiczny, który był przyczyną błędu
- zmiana stanu procesu na uruchomiony

Dzięki zastosowaniu pamięci wirtualnej:

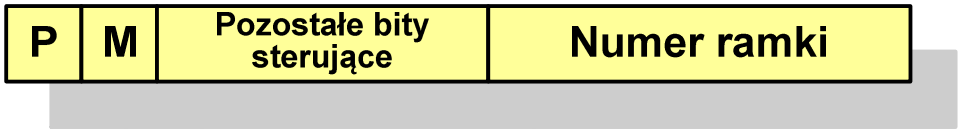
- w pamięci operacyjnej może być przechowywanych więcej procesów
- proces może być większy od całej pamięci operacyjnej

Stronicowanie pamięci wirtualnej

- przy zastosowaniu stronicowania, adres wirtualny (logiczny) ma postać:



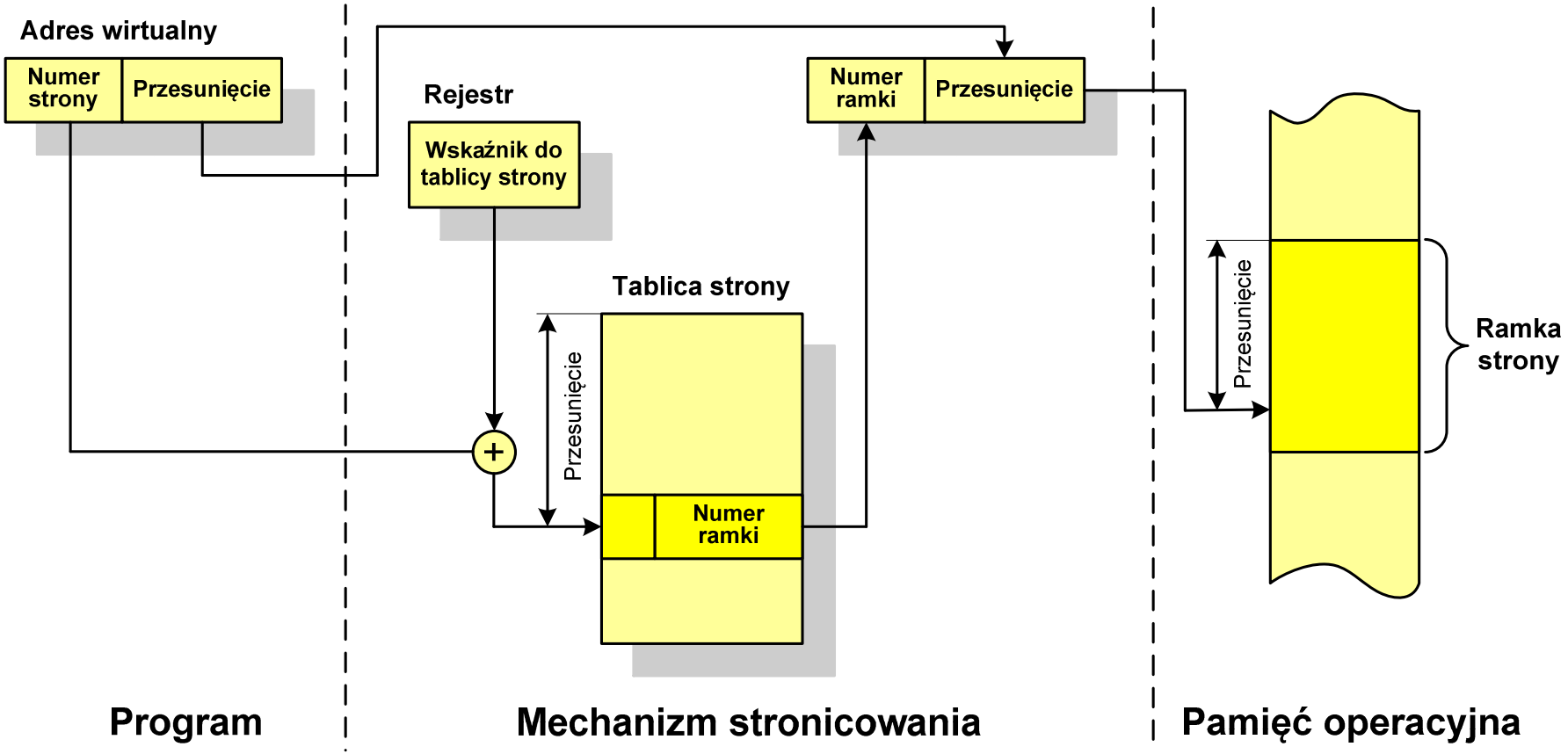
- mechanizm pamięci wirtualnej bazującej na stronicowaniu wymaga również tablicy stron



  - P - bit określający, czy strona znajduje się w pamięci operacyjnej, jeśli tak, to zapis zawiera numer ramki tej strony
  - M - bit określający, czy zawartość strony skojarzonej z tą tablicą została zmodyfikowana od ostatniego załadowania tej strony do pamięci - jeśli nie, to nie trzeba tej strony zapisywać, gdy ma być ona przeniesiona do pamięci pomocniczej

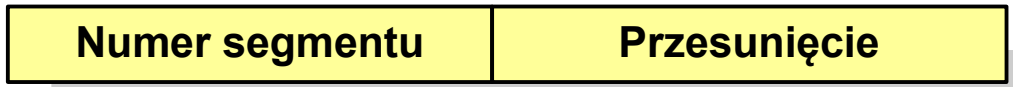
Stronicowanie pamięci wirtualnej

- odczytanie strony wymaga translacji adresu wirtualnego na fizyczny



Segmentacja pamięci wirtualnej

- w przypadku segmentacji, adres wirtualny ma postać:

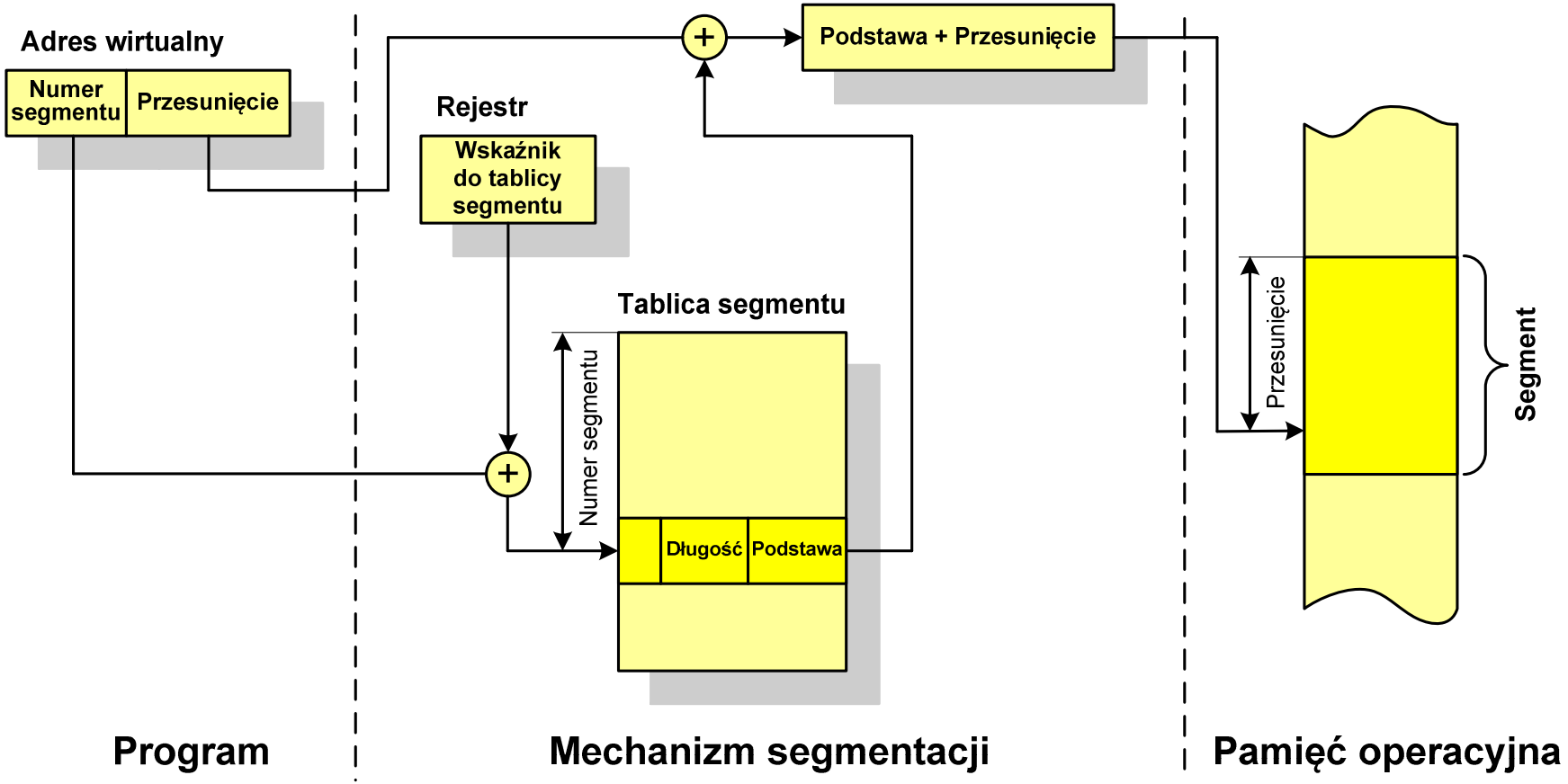


- mechanizm pamięci wirtualnej wykorzystujący segmentację wymaga tablicy segmentu zawierającej więcej pól

  - P - bit określający, czy segment znajduje się w pamięci operacyjnej
  - M - bit określający, czy zawartość segmentu skojarzonego z tablicą została zmodyfikowana od ostatniego załadowania tego segmentu do pamięci

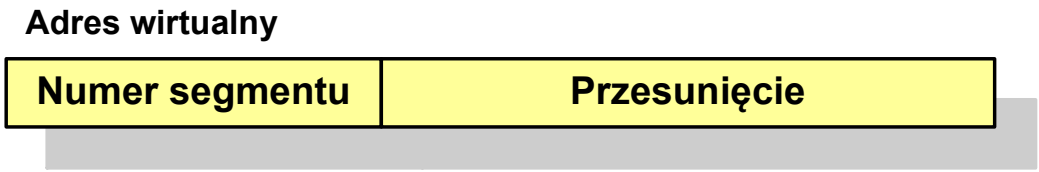
Segmentacja pamięci wirtualnej

- mechanizm odczytania słowa z pamięci obejmuje translację adresu wirtualnego na fizyczny za pomocą tablicy segmentu

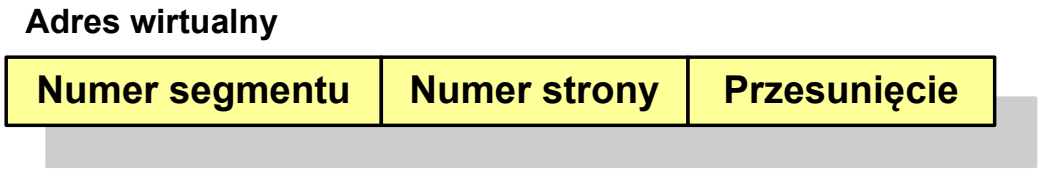


Stronicowanie i segmentacja pamięci wirtualnej

- przestrzeń adresowa użytkownika jest dzielona na dowolną liczbę segmentów według uznania programisty
- każdy segment jest dzielony na dowolną liczbę stron o stałym rozmiarze równym długości ramki pamięci operacyjnej
- z punktu widzenia programisty adres logiczny składa się z numeru segmentu oraz jego przesunięcia

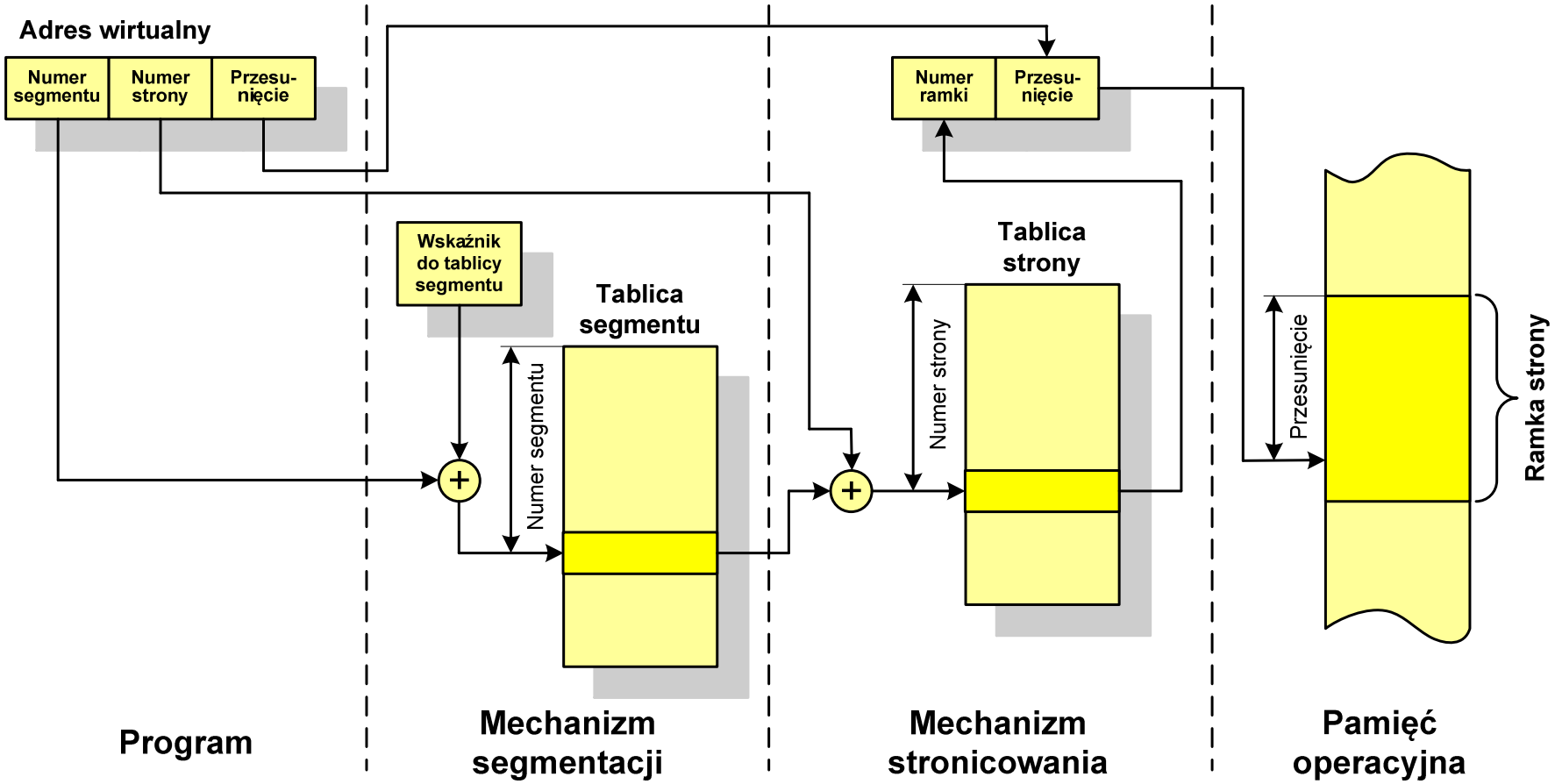


- z punktu widzenia systemu, przesunięcie segmentu jest postrzegane jako numer strony oraz przesunięcie strony dla strony wewnątrz określonego segmentu



Stronicowanie i segmentacja pamięci wirtualnej

- tłumaczenie adresu wirtualnego na adres fizyczny:



Koniec wykładu nr 14

Dziękuję za uwagę!