

# Informatyka 1

Politechnika Białostocka - Wydział Elektryczny  
Elektrotechnika, semestr II, studia niestacjonarne I stopnia  
Rok akademicki 2017/2018

## Wykład nr 8 (11.05.2018)

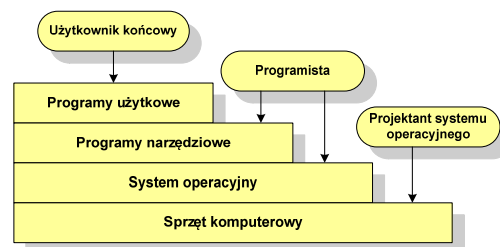
dr inż. Jarosław Forenc

## Plan wykładu nr 8

- System operacyjny
  - definicje systemu operacyjnego
- Zarządzanie procesami
  - definicja procesu, blok kontrolny procesu
  - dwu- i pięciostanowy model procesu
- Zarządzanie dyskowymi operacjami we-wy
  - metody przydziału pamięci dyskowej
  - systemy plików (FAT12, FAT16, FAT32, exFAT, NTFS, ext2)
- Zarządzanie pamięcią operacyjną
  - partycjonowanie, stronicowanie, segmentacja
  - pamięć wirtualna
- Sieci komputerowe
  - definicja, podział, topologie i media transmisyjne

## System operacyjny - definicja

- **System operacyjny** - jest to program sterujący wykonywaniem aplikacji i działający jako interfejs pomiędzy aplikacjami (użytkownikiem) a sprzętem komputerowym
- **użytkownik końcowy** nie jest zainteresowany sprzętem, interesują go tylko **aplikacje** (programy użytkowe)
- aplikacje są tworzone przez **programistów** za pomocą języków programowania



## System operacyjny - definicja

- **System operacyjny - administrator zasobów** - zarządza i przydziela zasoby systemu komputerowego oraz steruje wykonaniem programu
- **zasób systemu** - każdy element systemu, który może być przydzielony innej części systemu lub oprogramowaniu aplikacyjnemu
- do zasobów systemu zalicza się:
  - czas procesora
  - pamięć operacyjną
  - urządzenia zewnętrzne

## Zarządzanie procesami

- głównym zadaniem systemu operacyjnego jest **zarządzanie procesami**
- Definicja procesu:
  - **proces** - program w trakcie wykonania
  - **proces** - ciąg wykonań instrukcji wyznaczanych kolejnymi wartościami licznika rozkazów wynikających z wykonywanej procedury (programu)
  - **proces** - jednostka, którą można przypisać procesorowi i wykonać
- Proces składa się z kilku elementów:
  - **kod programu**
  - **dane potrzebne programowi** (zmienne, przestrzeń robocza, bufor)
  - **kontekst wykonywanego programu** (stan procesu) - dane wewnętrzne, dzięki którym system operacyjny może nadzorować proces i nim sterować

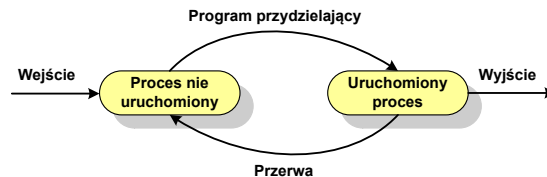
## Blok kontrolny procesu

- struktura danych tworzona i zarządzana przez system operacyjny, a opisująca właściwości procesu
- **identyfikator** - unikatowy numer skojarzony z procesem, dzięki któremu można odróżnić go od innych procesów
- **stan procesu**: nowy, gotowy, uruchomiony, zablokowany, anulowany
- **priorytet** - niski, normalny, wysoki, czasu rzeczywistego
- **licznik programu** - adres kolejnego rozkazu w programie, który ma zostać wykonany
- **wskaźniki pamięci** - wskaźniki do kodu programu, danych skojarzonych z procesem, dodatkowych bloków pamięci
- **dane kontekstowe** - dane znajdujące się w rejestrach procesora, gdy proces jest wykonywany
- **informacje na temat stanu żądań we-wy** - informacje na temat urządzeń we-wy przypisanych do tego procesu

Identyfikator
Stan
Priorytet
Licznik programu
Wskaźniki pamięci
Dane kontekstowe
Informacje na temat stanu żądań we/wy
Informacje ewidencyjne
...

## Dwustanowy model procesu

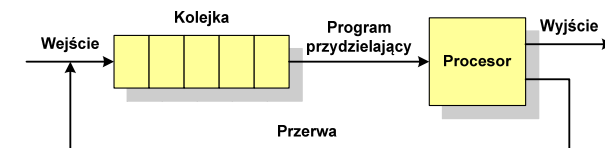
- najprostszy model polega na tym, że w dowolnej chwili proces jest wykonywany przez procesor (**uruchomiony**) lub nie (**nie uruchomiony**)



- system operacyjny tworząc nowy proces, tworzy blok kontrolny procesu po czym wprowadza proces do systemu jako nie uruchomiony
- w pewnym momencie aktualnie wykonywany proces zostaje przerwany i program przydzielający wybiera inny proces do wykonania
- stan poprzednio uruchomionego procesu jest zmieniany z uruchomionego na nie uruchomiony

## Dwustanowy model procesu

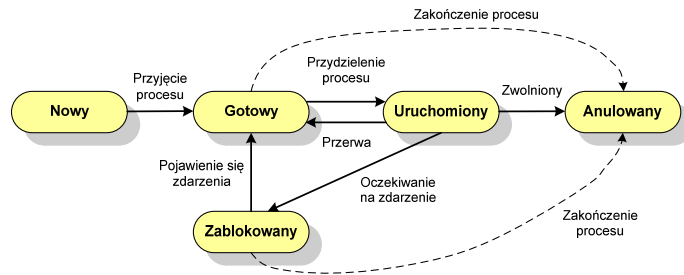
- procesy, które nie są uruchomione czekają w kolejce na wykonanie



- jeśli wykonywanie procesu zostało anulowane lub zakończone, to opuszcza on system, a program przydzielający wybiera kolejny proces z kolejki, który zostanie wykonany

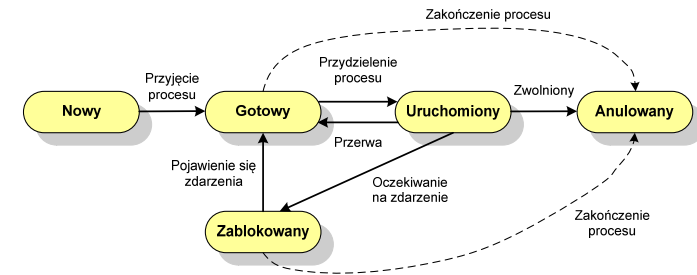
## Pięciostanowy model procesu

- w dwustanowym modelu procesu kolejka działa na zasadzie FIFO, a procesor wykonuje procesy cyklicznie z kolejki
- problem pojawia się w przypadku, gdy kolejny proces pobierany do wykonania z kolejki jest zablokowany, gdyż oczekuje na zakończenie operacji we-wy
- rozwiązaniem powyższego problemu jest podział procesów nieruchomych na **gotowe do wykonania** i **zablokowane**



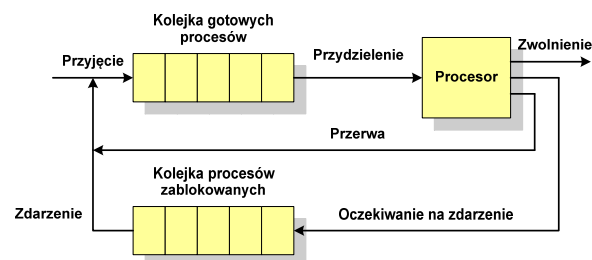
## Pięciostanowy model procesu

- **uruchomiony** - proces aktualnie wykonywany
- **gotowy** - proces gotowy do wykonania przy najbliższej możliwej okazji
- **zablokowany** - proces oczekujący na zakończenie operacji we-wy
- **nowy** - proces, który właśnie został utworzony (ma utworzony blok kontrolny procesu, nie został jeszcze załadowany do pamięci), ale nie został jeszcze przyjęty do grupy procesów oczekujących na wykonanie
- **anulowany** - proces, który został wstrzymany lub anulowany z jakiegoś powodu



## Pięciostanowy model procesu

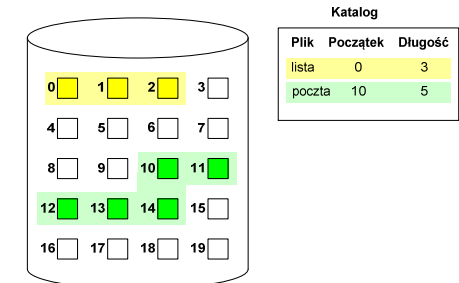
- podział procesów nieruchomych na **gotowe do wykonania** i **zablokowane** wymaga zastosowania minimum dwóch kolejek



- gdy pojawia się zdarzenie system operacyjny musi przejrzeć kolejkę szukając procesów, który związane są z danym zdarzeniem
- w celu zapewnienia większej wydajności lepiej jest gdy dla każdego zdarzenia istnieje oddzielna kolejka

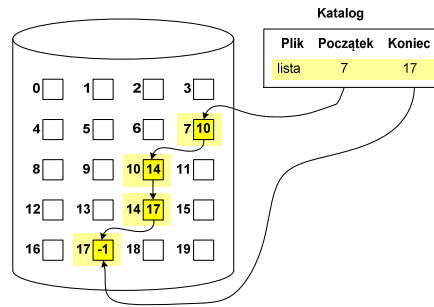
## Przydział pamięci dyskowej - alokacja ciągła

- każdy plik zajmuje ciąg kolejnych bloków na dysku
- plik zdefiniowany jest przez adres pierwszego bloku i ilość kolejnych zajmowanych bloków
- zalety: małe opóźnienia w transmisji danych, łatwy dostęp do dysku
- wady: trudność w znalezieniu miejsca na nowy plik



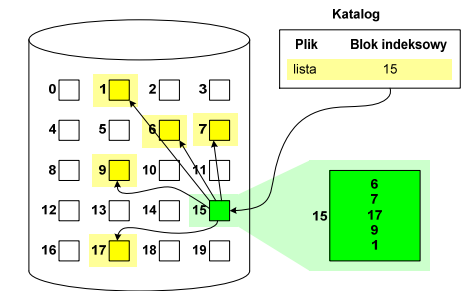
## Przydział pamięci dyskowej - alokacja listowa

- każdy plik jest listą powiązanych ze sobą bloków dyskowych, które mogą znajdować się w dowolnym miejscu na dysku
- w katalogu dla każdego pliku zapisany jest wskaźnik do pierwszego i ostatniego bloku pliku
- każdy blok zawiera wskaźnik do następnego bloku



## Przydział pamięci dyskowej - alokacja indeksowa

- każdy plik ma własny blok indeksowy, będący tablicą adresów bloków dyskowych
- w katalogu zapisany jest dla każdego pliku adres bloku indeksowego



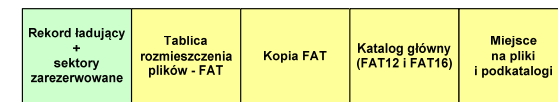
## System plików FAT (File Allocation Table)

- 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
- obsługuje  $2^{12} = 4096$  jednostek alokacji, max. rozmiar partycji to 16 MB
- 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



## FAT12

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

Zawartość pola:

Bajty	Rozmiar	Zawartość
00H-07H	8	Nazwa pliku w kodach ASCII
08H-0AH	3	Rozszerzenie nazwy pliku
0BH	1	Atrybuty pliku
0CH-15H	10	Zarezerwowane
16H-17H	2	Czas utworzenia lub aktualizacji pliku
18H-19H	2	Data utworzenia lub aktualizacji pliku
1AH-1BH	2	Numer pierwszej JAP
1CH-1DH	2	Mniej znaczące słowo rozmiaru pliku
1EH-1FH	2	Bardziej znaczące słowo rozmiaru pliku

Atrybuty pliku:

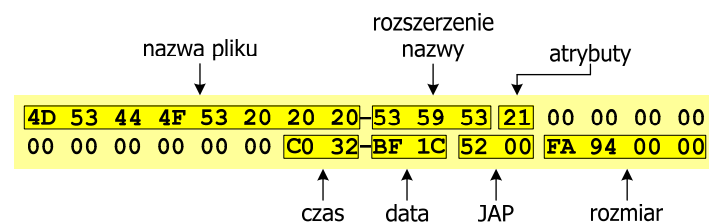
Bit	Znaczenie
0	Plik tylko do odczytu (read only)
1	Plik ukryty (hidden)
2	Plik systemowy (system)
3	Etykieta dysku (volume label)
4	Podkatalog
5	Plik archiwalny (archive)
6,7	Nie wykorzystywane

## FAT12

- przykładowa zawartość katalogu głównego:

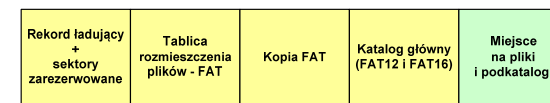
```

0000 49 4F 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

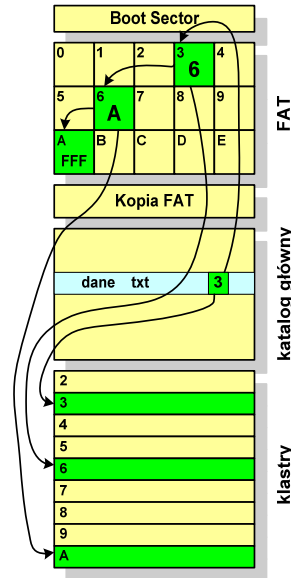
- pozostałą część dysku zajmuje miejsce na pliki i podkatalogi



- podkatalogi nie są ograniczone co do wielkości, zapisywane są na dysku w sposób identyczny jak pliki użytkowe i także zawierają 32-bajtowe pola

## 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



## FAT12 - struktura dyskietki (1,44 MB)



- całkowita liczba sektorów na dyskietce: 2880
- liczba sektorów systemowych: 33

Rekord ładujący + sektory zarezerwowane	Tablica rozmieszczenia plików - FAT	Kopia FAT	Katalog główny (FAT12 i FAT16)	Miejsce na pliki i podkatalogi
1 sektor	9 sektorów	9 sektorów	14 sektorów	2847 sektorów

- maksymalna liczba plików w katalogu głównym: 224
- liczba sektorów na pliki i podkatalogi:  $2880 - 33 = 2847$
- dostępne miejsce na pliki i podkatalogi:  $2847 \times 512 = 1\,457\,664$  bajty

## FAT16

- po raz pierwszy pojawił się w systemie MS-DOS 3.3
- ogólna struktura dyskietki / dysku logicznego w systemie FAT16 jest taka sama jak w przypadku FAT12

Rekord ładujący + sektory zarezerwowane	Tablica rozmieszczenia plików - FAT	Kopia FAT	Katalog główny (FAT12 i FAT16)	Miejsce na pliki i podkatalogi

- maksymalna liczba jednostek alokacji ograniczona jest do  $2^{16}$  czyli 65536
- w systemach **DOS** i **Windows 95** maksymalny rozmiar JAP to  $2^{15}$  bajtów czyli 32 kB, stąd maksymalny rozmiar dysku logicznego w tych systemach to  $65536 \times 32$  kB czyli **ok. 2 GB**
- w systemie **Windows 2000** górna granica rozmiaru JAP wynosi  $2^{16}$  bajtów (64 kB), czyli rozmiar dysku logicznego zwiększa się do **4 GB**.

## 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

Rekord ładujący + sektory zarezerwowane	Tablica rozmieszczenia plików - FAT	Kopia FAT	Miejsce na 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
- dla JAP od 4 kB do 32 kB, teoretycznie dysk może mieć rozmiar **8 TB**, ale praktycznie ograniczenie (MBR) to liczba  $2^{32}$  sektorów, czyli **2 TB**
- w systemie FAT32 można formatować tylko dyski, nie można natomiast zainstalować go na dyskietkach

## FAT32

- w systemie FAT32 katalog główny może znajdować się w dowolnym miejscu na dysku i może zawierać maksymalnie 65 532 pliki i katalogi

Bajty	Rozmiar	Zawartość
00H-07H	8	Nazwa pliku w kodach ASCII
08H-0AH	3	Rozszerzenie nazwy pliku
0BH	1	Atrybuty pliku
0CH	1	Wielkość liter nazwy i rozszerzenia pliku
0DH	1	Czas utworzenia w milisekundach
0EH-0FH	2	Czas utworzenia
10H-11H	2	Data utworzenia
12H-13H	2	Czas ostatniego dostępu
14H-15H	2	Numer pierwszej JAP (16 starszych bitów)
16H-17H	2	Czas utworzenia lub aktualizacji pliku
18H-19H	2	Data utworzenia lub aktualizacji pliku
1AH-1BH	2	Numer pierwszej JAP (16 młodszych bitów)
1CH-1DH	2	Mniej znaczące słowo rozmiaru pliku
1EH-1FH	2	Bardziej znaczące słowo rozmiaru pliku

## FAT32 - długie nazwy plików

- wprowadzone w systemie Windows 95
- informacje o nazwie pliku zapamiętywane są jako:
  - długa nazwa
  - skrócona nazwa (tzw. alias długiej nazwy)
- metoda tworzenia skróconej nazwy pliku:
  - rozszerzenie długiej nazwy staje się rozszerzeniem skróconej nazwy
  - pierwsze sześć znaków długiej nazwy staje się pierwszymi sześcioma znakami skróconej nazwy (nieodzwolone znaki zamieniane są na znak podkreślenia, małe litery zamieniane są na wielkie litery)
  - pozostałe dwa znaki nazwy skróconej to ~1 lub jeśli plik o takiej nazwie istnieje ~2, itd.

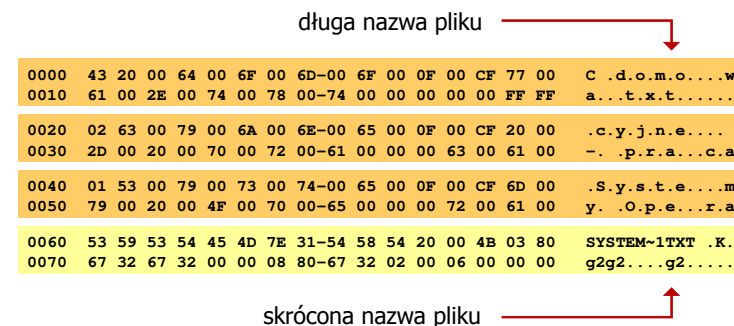
## FAT32 - długie nazwy plików

- 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)

Bajty	Rozmiar	Zawartość
00H	1	Pierwsze 6 bitów określa numer fragmentu nazwy, bit 7 - czy jest to ostatni fragment nazwy, a bit 8 - czy plik został usunięty lub jego nazwa skrócona
01H-0AH	10	Pierwsze 5 znaków nazwy
0BH	1	Atrybut (zawsze F)
0CH	1	Zarezerwowany (zawsze 0)
0DH	1	Suma kontrolna wersji krótkiej 8+3
0EH-19H	12	Kolejne 6 znaków nazwy
1AH-1BH	2	Numer początkowego kłastera (zawsze 0)
1CH-1FH	4	Dwie ostatnie litery nazwy

## FAT32 - długie nazwy plików

- Nazwa pliku: **Systemy Operacyjne - praca domowa.txt**



## FAT - wady systemu plików FAT

- **fragmentacja wewnętrzna** - nawet najmniejszy plik zajmuje całą JAP - gdy rozmiar klastra jest duży, a na dysku znajduje się dużo małych plików - pewna część miejsca jest tracona
- **fragmentacja zewnętrzna** - silna fragmentacja plików pomiędzy wiele klastrów o bardzo różnym fizycznym położeniu na dysku (konieczność okresowej defragmentacji przy użyciu specjalnych narzędzi programowych)
- duże prawdopodobieństwo powstawania błędów zapisu, polegających na przypisaniu jednego klastra dwóm plikom (tzw. **crosslinks**), co kończy się utratą danych z jednego lub obu „skrzyżowanych” plików
- typowym błędem, pojawiającym się w systemie FAT, jest również pozostawianie tzw. **zagubionych klastrów (lost chains)**, tj. jednostek alokacji nie zawierających informacji, ale opisanych jako zajęte
- brak mechanizmów ochrony - praw dostępu

## exFAT (FAT64)

- stworzony przez Microsoft na potrzeby pamięci Flash
- po raz pierwszy pojawił się w listopadzie 2006 roku w Windows Embedded CE 6.0 i Windows Vista SP1
- obsługiwany także przez Windows 7, Windows 8, Windows 10, Windows XP SP2/SP3, Linux
- może być używany wszędzie tam, gdzie NTFS nie jest najlepszym rozwiązaniem ze względu na dużą nadmiarowość struktury danych
- 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 1.1** (jesień 1994 r.) - Windows NT 3.5
- **wersja 1.2** (NTFS 4) - Windows NT 3.51 (1995 r.)
- **wersja 3.0** (NTFS 5) - Windows 2000
- **wersja 3.1** (NTFS 5.1) - Windows XP/Server 2003/Vista/7/8/10
- teoretyczny **rozmiar partycji** NTFS wynosi  $2^{64}-1$  klastrów, ale Windows potrafi obsłużyć tylko  $2^{32}-1$  klastrów (dla klastra 64 kB - **ok. 256 TB**)
- tabela partycji w MBR dysku twardego ogranicza rozmiar partycji do **2 TB**
- teoretyczna **wielkość pliku** wynosi  $2^{64}$  bajtów minus 1 kB, ale Windows ogranicza ten rozmiar do  $2^{44}$  bajtów minus 64 kB (**ok. 16 TB**)

## NTFS

- struktura wolumenu (dysku) NTFS:



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



## NTFS

- struktura wolumenu (dysku) NTFS:



- pozostała część pliku MFT przeznaczona jest na rekordy wszystkich plików i katalogów umieszczonych na dysku
- jeśli pierwszy rekord MFT jest uszkodzony to system automatycznie odczytuje drugi rekord, w którym zapisana jest kopia pierwszego
- położenie obu metaplików **\$Mft** i **\$MftMirr** zapisane jest w sektorze startowym partycji

## NTFS

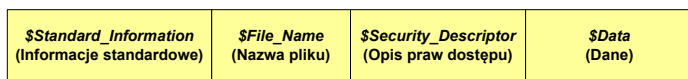
- struktura wolumenu (dysku) NTFS:



- 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

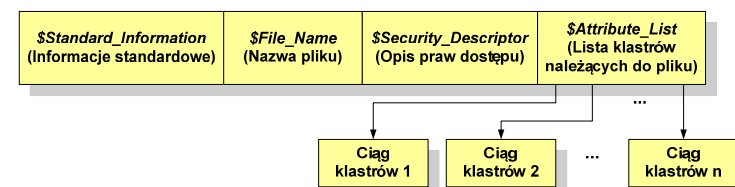
- pliki w systemie NTFS są reprezentowane w MFT przez rekord zawierający atrybuty:
  - **\$Standard\_Information**
  - **\$File\_Name**
  - **\$Security\_Descriptor**
  - **\$Data**



- 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**

Boot Sector	Bloki grupy 1	Bloki grupy 2	...	Bloki grupy N
-------------	---------------	---------------	-----	---------------

## ext2

- **Boot Sector** (blok startowy) przechowuje informacje wykorzystywane przez system operacyjny podczas jego uruchamiania

Boot Sector	Bloki grupy 1	Bloki grupy 2	...	Bloki grupy N
-------------	---------------	---------------	-----	---------------

- na poziomie logicznym **grupę bloków** tworzą:
  - deskryptor grupy (32 bajty)
  - blok z mapą zajętości bloków danych (1 blok dyskowy)
  - blok z mapą zajętości i-węzłów (1 blok dyskowy)
  - bloki z tablicą i-węzłów
  - bloki danych

Deskryptor grupy	Blok z mapą bitową zajętości bloków danej grupy	Blok z mapą bitową zajętości i-węzłów danej grupy	Tablica i-węzłów	Bloki danych
------------------	---	---	------------------	--------------

## ext2

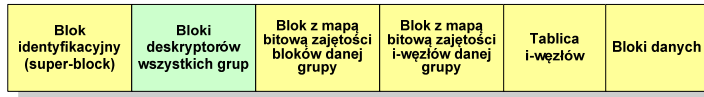
- każda **grupa fizyczna bloków** zawiera informacje o jednej grupie logicznej, a ponadto pewne informacje o całym systemie plików

Blok identyfikacyjny (super-blok)	Bloki deskryptorów wszystkich grup	Blok z mapą bitową zajętości bloków danej grupy	Blok z mapą bitową zajętości i-węzłów danej grupy	Tablica i-węzłów	Bloki danych
-----------------------------------	------------------------------------	---	---	------------------	--------------

- w każdej grupie fizycznej bloków znajduje się kopia tego samego bloku identyfikacyjnego oraz kopia bloków z deskryptorami wszystkich grup
- **blok identyfikacyjny** zawiera informacje na temat systemu plików:
  - numer urządzenia, na którym jest super-blok
  - rodzaj systemu plików
  - rozmiar bloku
  - struktury do synchronizacji dostępu
  - czas dokonanej ostatnio zmiany
  - informacje specyficzne dla konkretnej implementacji

## ext2

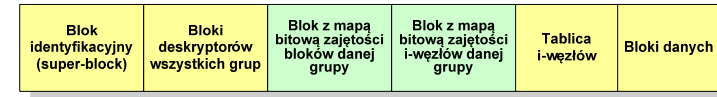
- każda **grupa fizyczna bloków** zawiera informacje o jednej grupie logicznej, a ponadto pewne informacje o całym systemie plików



- 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 wolnych i-węzłów w grupie
  - liczba katalogów w grupie

## ext2

- każda **grupa fizyczna bloków** zawiera informacje o jednej grupie logicznej, a ponadto pewne informacje o całym systemie plików



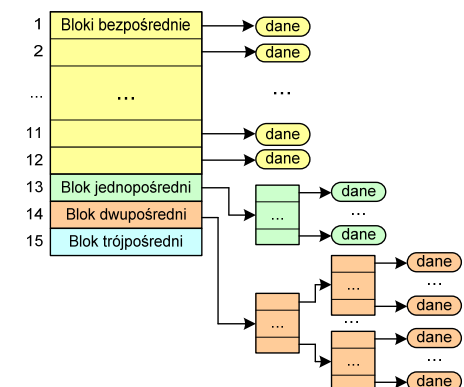
- blok z mapą bitową zajętości bloków danej grupy** jest tablicą bitów o rozmiarze jednego bloku
  - jeśli blok ma rozmiar 1 kB to pojedynczą mapą można opisać fizyczna grupę 8096 bloków czyli 8 MB danych
  - jeśli natomiast blok ma rozmiar 4 kB, to fizyczna grupa bloków zajmuje 128 MB danych
- przed tablicą i-węzłów znajduje się **blok z mapą 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 jednopięsredni** (ang. single indirect block)
  - 1 adres bloku zawierającego adresy bloków jednopięsrednich - **blok dwupięsredni** (ang. double indirect block)
  - 1 adres bloku zawierającego adresy bloków dwupięsrednich - **blok trójpięsredni** (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)

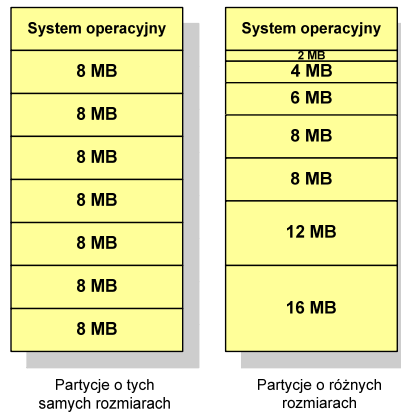
```
struct ext2_dir_entry
{
    _u32  inode           /* numer i-wezla      */
    _u16  rec_len        /* dlugosc pozycji katalogowej */
    _u16  name_len      /* dlugosc nazwy     */
    char  name[EXT2_NAME_LEN] /* nazwa             */
}
```

## 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:
  - partycjonowanie statyczne, partycjonowanie dynamiczne
  - proste stronicowanie, prosta segmentacja
  - stronicowanie pamięci wirtualnej, segmentacja pamięci wirtualnej
  - stronicowanie i segmentacja pamięci wirtualnej

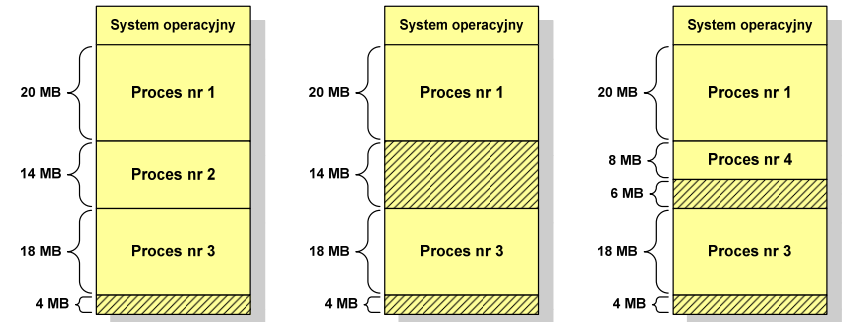
## Partycjonowanie statyczne

- podział pamięci operacyjnej na obszary o takim samym lub różnym rozmiarze, ustalonym podczas generowania systemu



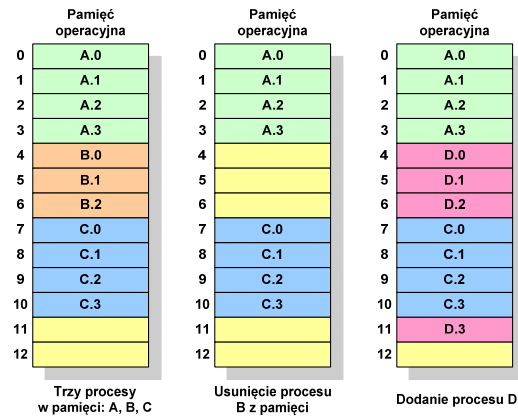
## Partycjonowanie dynamiczne

- partycje są tworzone dynamicznie w ten sposób, że każdy proces jest ładowany do partycji o rozmiarze równym rozmiarowi procesu
- partycje mają różną długość, może zmieniać się także ich liczba
- przykład - w systemie działa 5 procesów: 20 MB, 14 MB, 18 MB, 8 MB, 8 MB



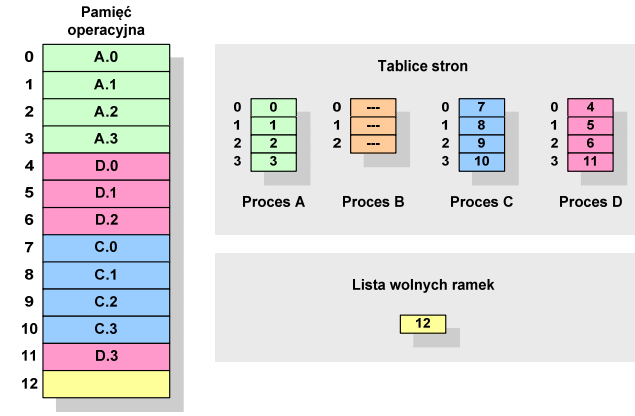
## Proste stronicowanie

- 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

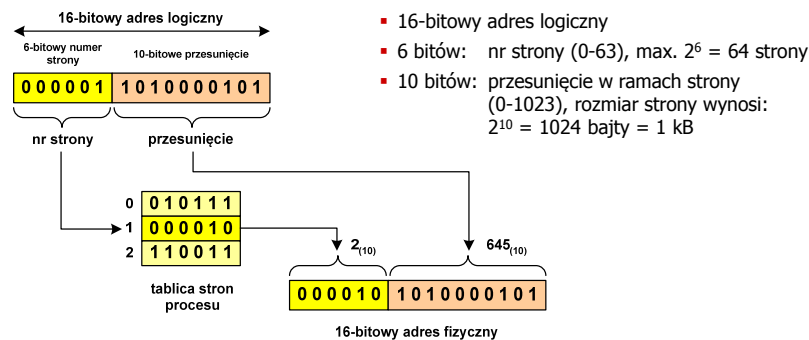
- dla każdego procesu przechowywana jest **tablica strony** (page table) zawierająca lokalizację ramki dla każdej strony procesu



## Proste stronicowanie

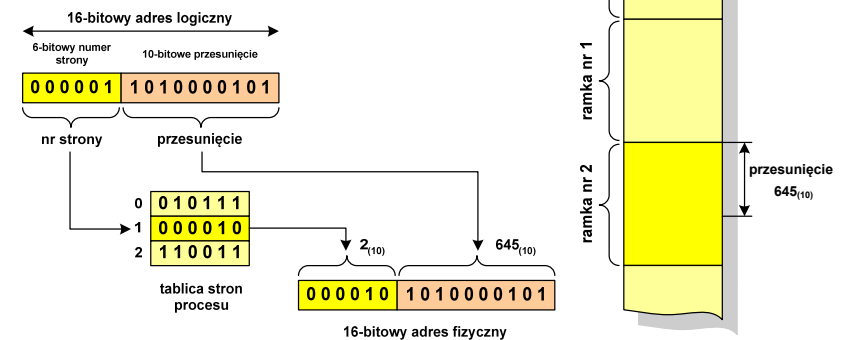
- aby mechanizm stronicowania był wygodny ustala się, że rozmiar strony jest liczbą podniesioną do potęgi drugiej - dzięki temu adres względny oraz adres logiczny (numer strony + jej przesunięcie) są takie same

Przykład:



## Proste stronicowanie

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



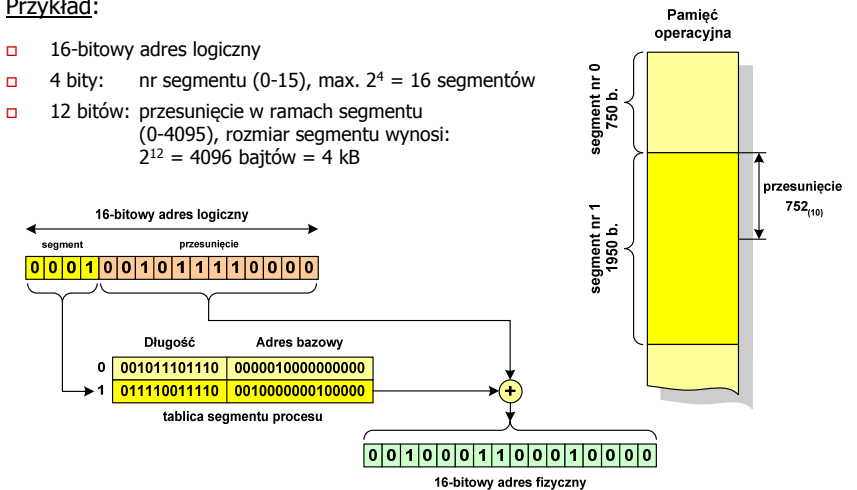
## Prosta segmentacja

- 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?

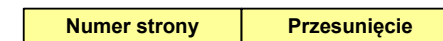
- generowanie przerwania sygnalizującego błąd w dostępie do pamięci
- zmiana stan procesu na zablokowany
- wstawienie do pamięci operacyjnej fragment procesu zawierający adres logiczny, który był przyczyną błędu
- zmiana stanu procesu na uruchomiony

Dzięki zastosowaniu pamięci wirtualnej:

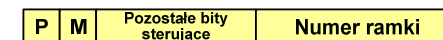
- w pamięci operacyjnej może być przechowywanych więcej procesów
- proces może być większy od całej pamięci operacyjnej

## Stronicowanie pamięci wirtualnej

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



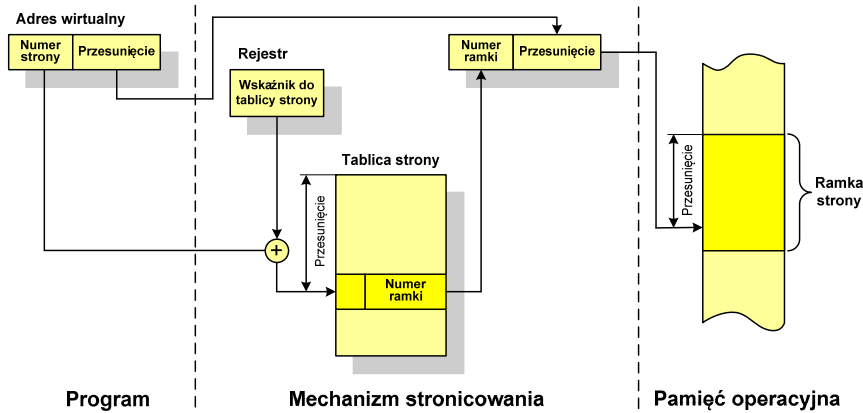
- mechanizm pamięci wirtualnej bazującej na stronicowaniu wymaga również tablicy stron



- **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

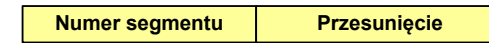
## Stronicowanie pamięci wirtualnej

- odczytanie strony wymaga translacji adresu wirtualnego na fizyczny

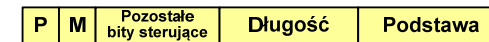


## Segmentacja pamięci wirtualnej

- w przypadku segmentacji, **adres wirtualny** ma postać:



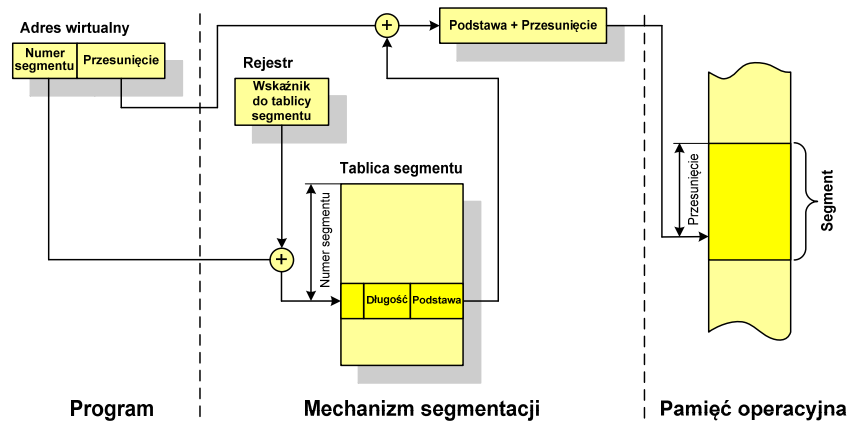
- mechanizm pamięci wirtualnej wykorzystujący segmentację wymaga **tablicy segmentu** zawierającej więcej pól



- **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

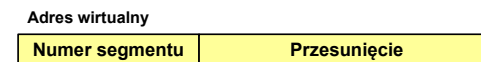
## Segmentacja pamięci wirtualnej

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

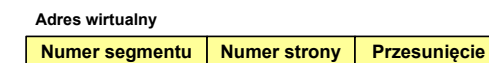


## Stronicowanie i segmentacja pamięci wirtualnej

- 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

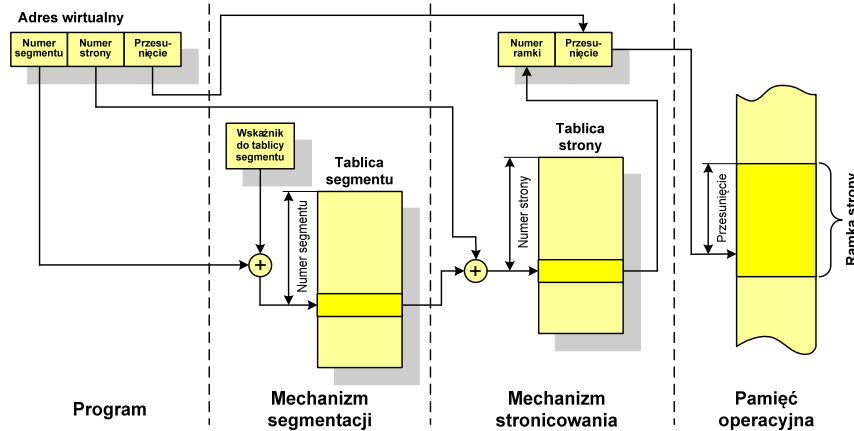


- z punktu widzenia systemu, przesunięcie segmentu jest postrzegane jako numer strony oraz przesunięcie strony dla strony wewnątrz określonego segmentu



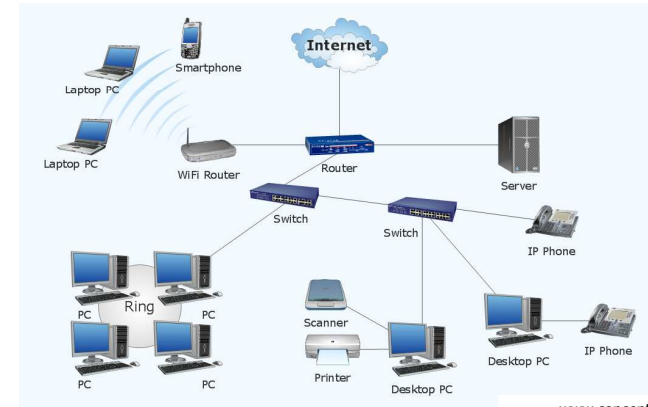
## Stronicowanie i segmentacja pamięci wirtualnej

- tłumaczenie adresu wirtualnego na adres fizyczny:



## Sieć komputerowa

- Sieć komputerowa - zbiór komputerów i innych urządzeń umożliwiających wzajemne przekazywanie informacji oraz udostępnianie zasobów



## Podział sieci w zależności od ich rozmiaru

- LAN (Local Area Network) - sieć lokalna, łączy komputery znajdujące się na określonym, niewielkim obszarze (kilka budynków, przedsiębiorstwo), wykonana jest w jednej technologii (np. Ethernet)
- MAN (Metropolitan Area Network) - sieć miejska, obejmuje zasięgiem aglomerację lub miasto łącząc oddzielne sieci LAN (np. Biaman)
- WAN (Wide Area Network) - sieć rozległa, łączy ze sobą sieci MAN i LAN na obszarze wykraczającym poza jedno miasto (POL-34, Pionier)
- Internet - ogólnosiwiatowa sieć komputerowa łącząca ze sobą wszystkie rodzaje sieci („sieć sieci”)
- Intranet - sieć podobna do Internetu, ale ograniczająca się do komputerów w firmie lub organizacji

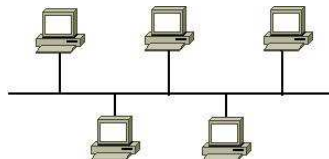
## Topologie sieci komputerowych

- Topologia sieci - określa strukturę sieci
  - zbiór zasad fizycznego łączenia elementów sieci (topologia fizyczna)
  - zbiór reguł komunikacji poprzez medium transmisyjne (topologia logiczna)
- Topologia fizyczna - opisuje sposoby fizycznego łączenia ze sobą komputerów (układ przewodów, media transmisyjne)
- Topologia logiczna - opisuje sposoby komunikowania się hostów za pomocą urządzeń topologii fizycznej; standardy komunikacji definiowane przez IEEE:
  - IEEE 802.3 - 10 Mb Ethernet
  - IEEE 802.3u - 100 Mb Ethernet
  - IEEE 802.3z - 1 Gb Ethernet
  - IEEE 802.5 - Token Ring
  - IEEE 802.11 - Wireless LAN
  - IEEE 802.14 - Cable Modem

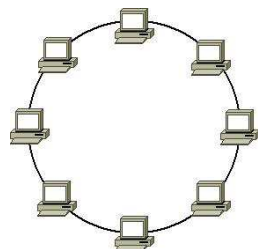


## Topologie sieci komputerowych

- **topologia magistrali (bus)** - wszystkie komputery podłączone są do jednego współdzielonego medium transmisyjnego (najczęściej kabla koncentrycznego)



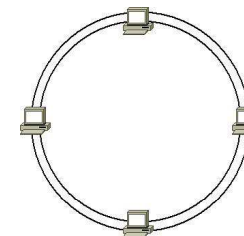
- **topologia pierścienia (ring)** - komputery połączone są pomiędzy sobą odcinkami kabla tworząc zamknięty pierścień (sieci światłowodowe, sieci LAN)



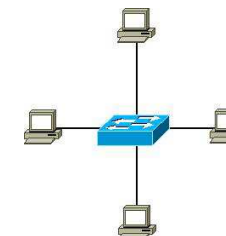
pl.wikipedia.org

## Topologie sieci komputerowych

- **topologia podwójnego pierścienia (dual-ring)** - komputery połączone są pomiędzy sobą odcinkami kabla tworząc dwa zamknięte pierścienie (większa niezawodność, sieci: szkieletowe, MAN, Token Ring, FDDI)



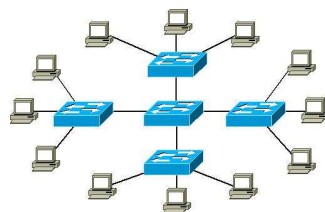
- **topologia gwiazdy (star)** - komputery podłączone są do jednego punktu centralnego (koncentrator, przełącznik), obecnie jest to najczęściej stosowana topologia sieci LAN



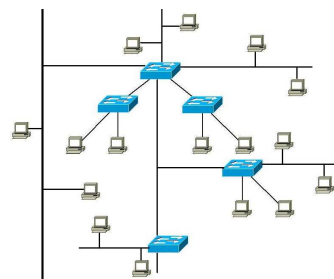
pl.wikipedia.org

## Topologie sieci komputerowych

- **topologia rozszerzonej gwiazdy (extended star)** - posiada punkt centralny i punkty poboczne (stosowana w rozbudowanych sieciach lokalnych)



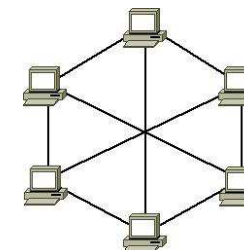
- **topologia hierarchiczna (drzewa)** - jest kombinacją topologii gwiazdy i magistrali



pl.wikipedia.org

## Topologie sieci komputerowych

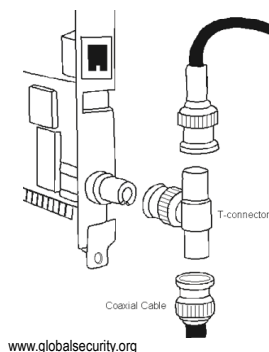
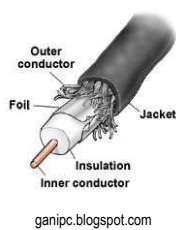
- **topologia siatki (mesh)** - każde urządzenie połączone jest z więcej niż jednym urządzeniem (sieci MAN i WAN, Internet)
  - **pełna siatka (full mesh)** - każdy węzeł sieci jest połączony fizycznie z każdym innym węzłem sieci
  - **częściowa siatka (partial mesh)** - węzły mają różną ilość połączeń sieciowych do innych węzłów



pl.wikipedia.org

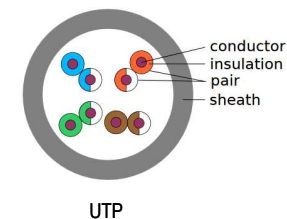
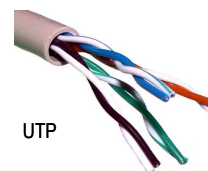
## Media transmisyjne - przewód koncentryczny

- **Ethernet gruby (Thick Ethernet), 10Base-5**
  - kabel RG-8 lub RG-11, impedancja falowa: 50  $\Omega$ , grubość: 1/2"
  - max. odległość między stacjami: 500 m
- **Ethernet cienki (Thin Ethernet), 10Base-2**
  - kabel RG-58, impedancja falowa: 50  $\Omega$ , grubość: 1/4"
  - max. odległość między stacjami: 185 m



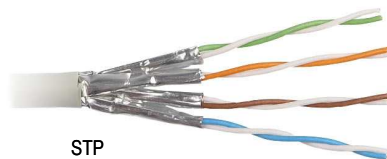
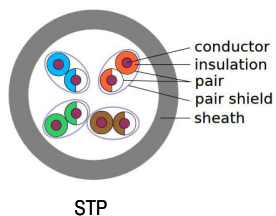
## Media transmisyjne - skrętka UTP

- **UTP (Unshielded Twisted Pair)** - skrętka nieekranowana
  - zbudowana z ośmiu przewodów skręconych po dwa (cztery pary) umieszczonych we wspólnej izolacji
  - wyróżnia się różne kategorie kabli (CAT-1, CAT-2, ..., CAT-7), najczęściej stosowane są kable kategorii 5 i 6
  - maksymalna długość segmentu sieci: 100 m
  - stosowane typy końcówek: RJ-11, RJ-45



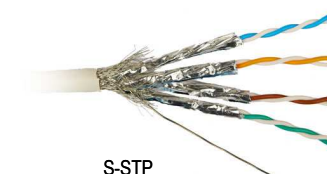
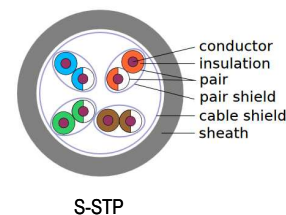
## Media transmisyjne - skrętka STP

- **STP (Shielded Twisted Pair)** - skrętka ekranowana
  - ekranowany kabel skręcony zbudowany z czterech skręconych ze sobą par przewodów miedzianych otoczonych ekranującą siatką lub folią i umieszczonych w izolacyjnej osłonie
  - ekran chroni skrętkę przed wpływem zewnętrznego promieniowania elektromagnetycznego



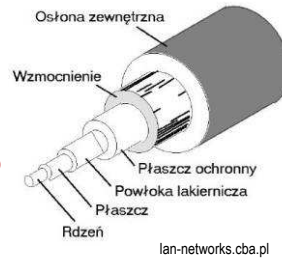
## Media transmisyjne - skrętka

- **F-FTP** - każda para przewodów otoczona jest osobnym ekranem z folii, cały kabel jest również pokryty folią
- **S-FTP** - każda para przewodów otoczona jest osobnym ekranem z folii, cały kabel pokryty jest oplotem
- **S-STP** - każda para przewodów otoczona jest osobnym ekranem (oplotem), cały kabel pokryty jest oplotem



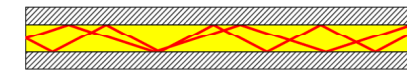
## Media transmisyjne - światłowód

- światłowód (fiber optic cable) przesyła impulsy świetlne między nadajnikiem i odbiornikiem
- nadajnik przekształca sygnały elektryczne na świetlne, a odbiornik przekształca sygnały świetlne na elektryczne
- impulsy świetlne są przenoszone przez **włókno optyczne** składające się z dwóch rodzajów szkła o różnych współczynnikach załamania światła
- budowa światłowodu:
  - rdzeń (core), średnica: 9  $\mu\text{m}$  lub 50  $\mu\text{m}$
  - płaszcz zewnętrzny (cladding), średnica: 125  $\mu\text{m}$
  - pokrycie zewnętrzne
- promień światła wędrując w rdzeniu pada na płaszcz pod pewnym kątem i następuje **zjawisko całkowitego odbicia wewnętrznego światła** - umożliwia to transmisję strumienia światła przez włókno



## Media transmisyjne - światłowody wielomodowe

- w światłowodzie **wielomodowym (multi mode fiber)** promień światła może zostać wprowadzony pod różnymi kątami - modami
- fala świetlna o takiej samej długości może rozchodzić się wieloma drogami



- medium
- włókno szklane
- ▨ powłoka zewnętrzna

- źródło światła: diody LED
- długość fali świetlnej (850 nm i 1300 nm)
- ze względu na dyspersję maksymalna długość kabla to 5 km

## Media transmisyjne - światłowody jednomodowe

- w światłowodzie **jednomodowym (single mode fiber)** propaguje tylko jeden mod

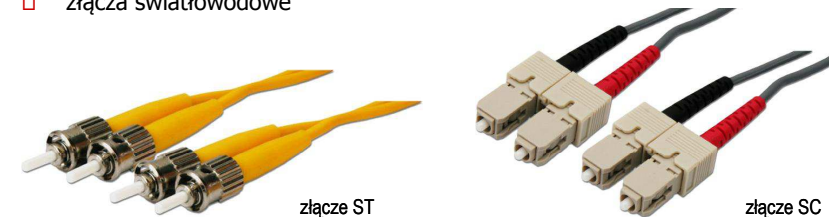


- medium
- włókno szklane
- ▨ powłoka zewnętrzna

- źródło światła: dioda laserowa
- długość fali świetlnej (1300 nm i 1500 nm)
- długość kabla: do 100 km
- wyższy koszt od światłowodów wielomodowych

## Media transmisyjne - światłowody

- złącza światłowodowe



- wybrane standardy transmisji
  - 10Base-FL - 10 Mb/s, rzadko spotykany
  - 100Base-FX - 100 Mb/s, do 2 km
  - 1000Base-LX - 1 Gb/s, do 10 km, jednomodowy
  - 10GBase-ZR - 10 Gb/s, do 80 km, jednomodowy

**Koniec wykładu nr 8**

**Dziękuję za uwagę!**  
**(następny wykład: 08.06.2018)**