

Informatyka 1 (ES1F1002)

Politechnika Białostocka - Wydział Elektryczny
Elektrotechnika, semestr I, studia stacjonarne I stopnia
Rok akademicki 2024/2025

Wykład nr 13 (17.01.2025)

dr inż. Jarosław Forenc

Plan wykładu nr 13

- System operacyjny
 - definicje systemu operacyjnego
- Zarządzanie procesami
 - definicja procesu, dwu- i pięciostanowy model procesu
- Zarządzanie dyskowymi operacjami we-wy
 - struktura dysku twardego (MBR, GPT)
 - systemy plików (FAT, NTFS, ext2)

System operacyjny - definicja

- **System operacyjny** - jest to program sterujący wykonywaniem aplikacji i działający jako interfejs pomiędzy aplikacjami (użytkownikiem) a sprzętem komputerowym
- 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, bufory)
 - **kontekst wykonywanego programu** (stan procesu) - dane wewnętrzne, dzięki którym system operacyjny może nadzorować proces i nim sterować

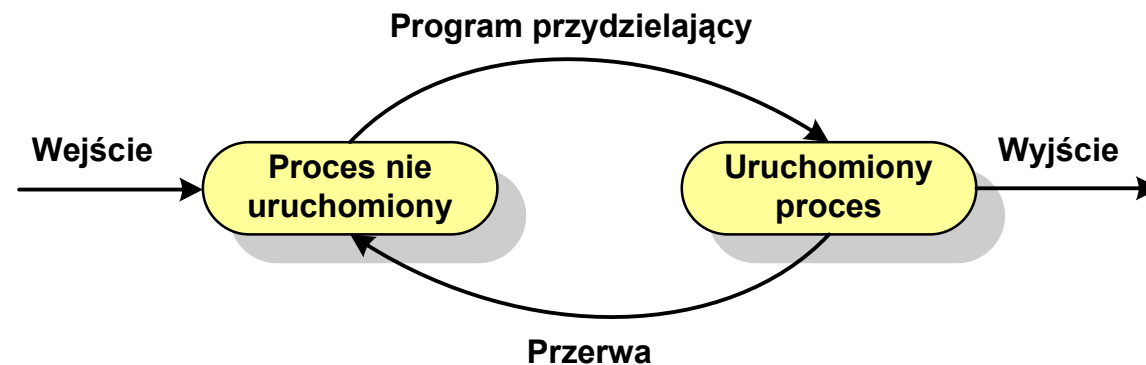
Blok kontrolny procesu (deskryptor procesu)

- 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

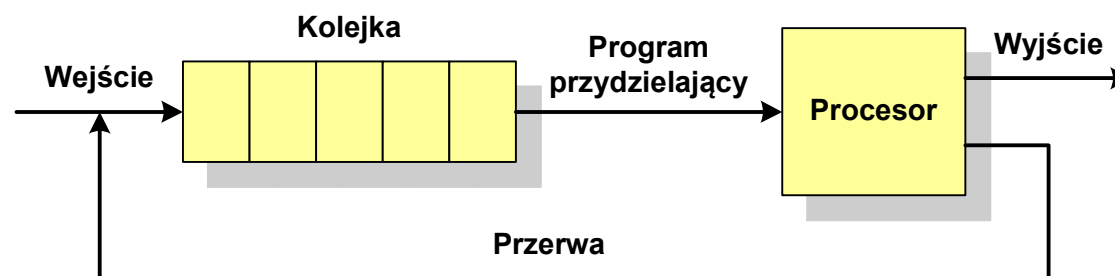
Identyfikator
Stan
Priorytet
Licznik programu
Wskaźniki pamięci
Dane kontekstowe
Informacje na temat stanu żądań we/wy
Informacje ewidencyjne
...

Dwustanowy model procesu

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



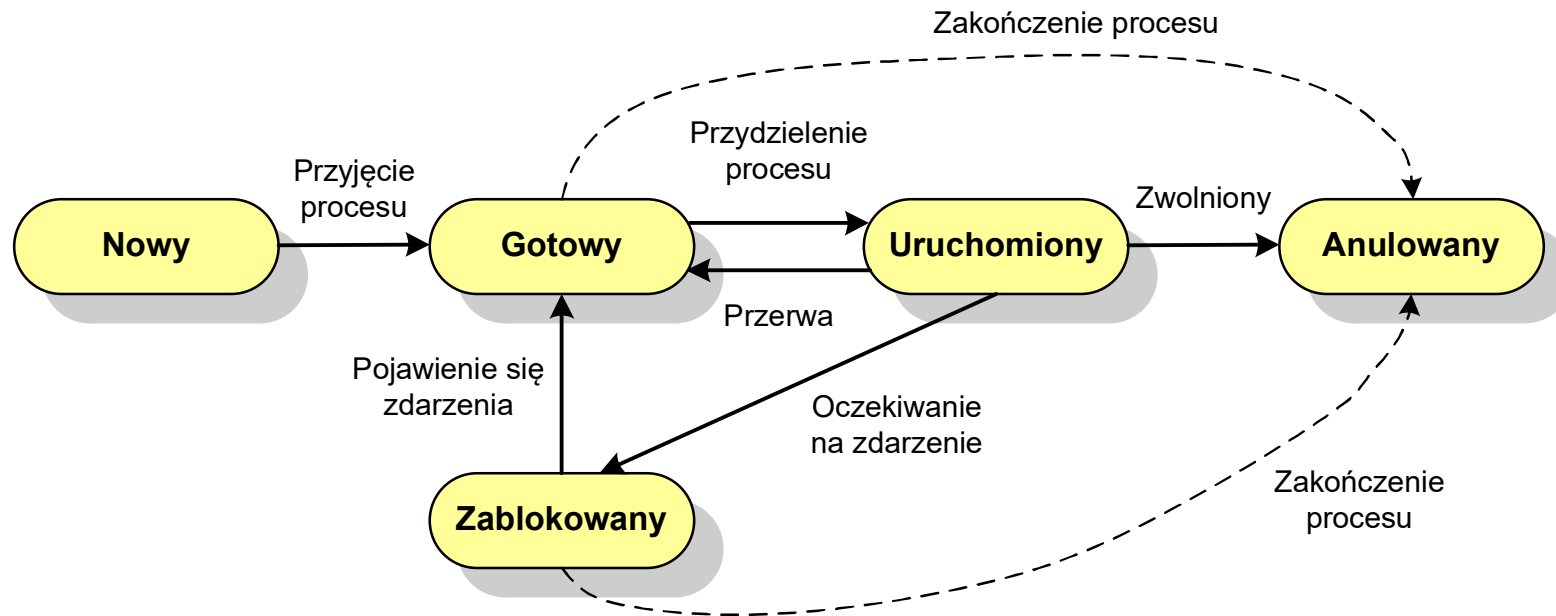
- procesy, które nie są uruchomione czekają w kolejce na wykonanie



- wadą tego modelu jest sytuacja, gdy kolejny proces pobierany do wykonania z kolejki jest **zablokowany**, gdyż oczekuje na **zakończenie operacji we-wy**

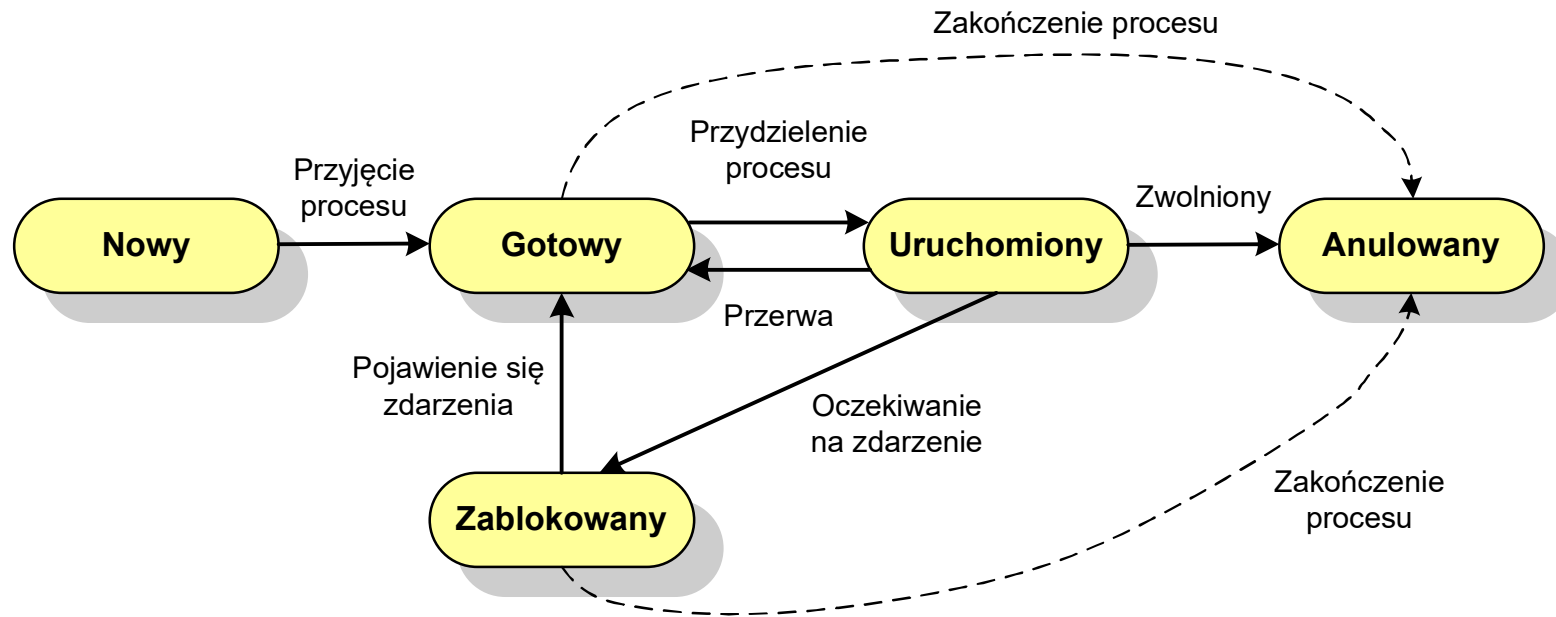
Pięciostanowy model procesu

- rozwiązaniem powyższego problemu jest podział procesów nieuruchomionych na **gotowe do wykonania** i **zablokowane**



- pięciostanowy model procesu wymaga zastosowania minimum dwóch kolejek: dla procesów **gotowych do wykonania** i **zablokowanych**

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

Zarządzanie dyskowymi operacjami we-wy

- Struktura dysku twardego
 - MBR (BIOS)
 - GPT (UEFI)

- Systemy plików
 - FAT (FAT12, FAT16, FAT32, exFAT)
 - NTFS
 - ext2

Struktura dysku twardego - MBR

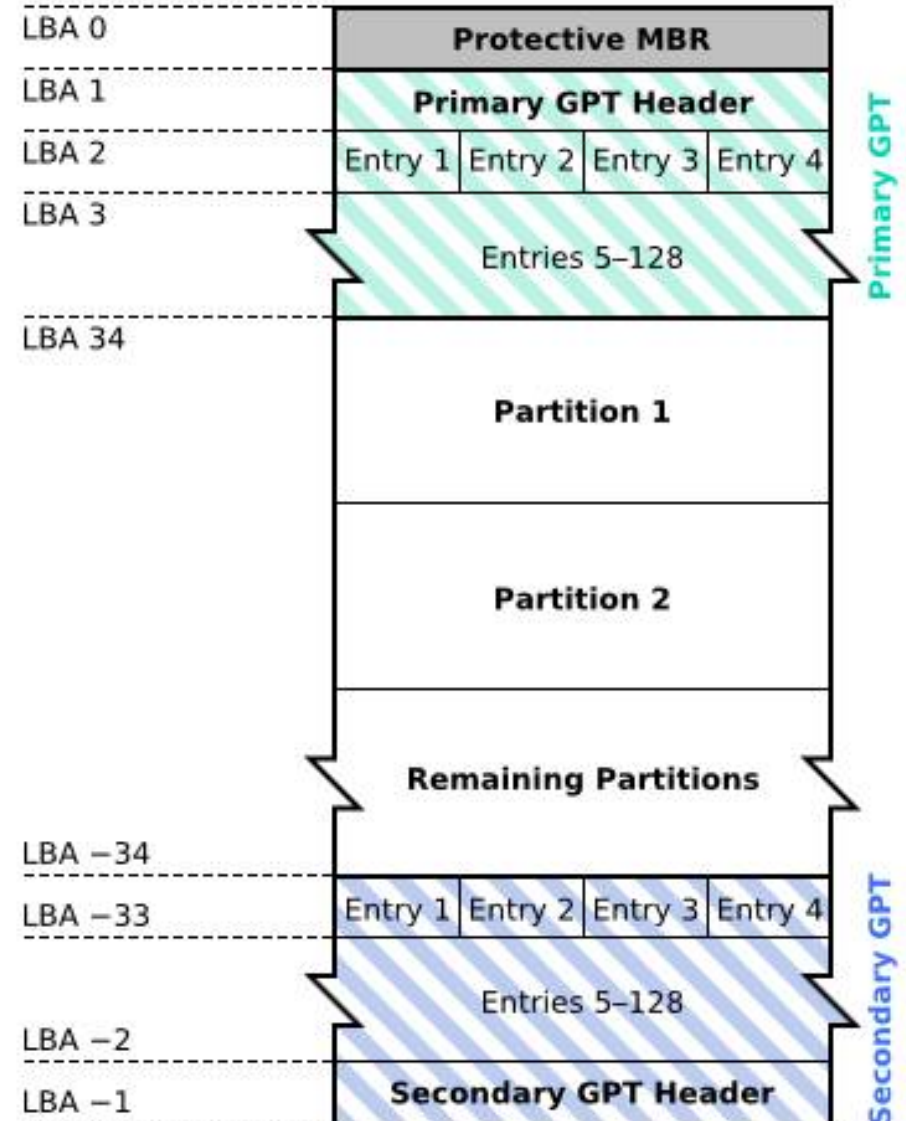
- ❑ **MBR (Master Boot Record)** - główny rekord ładujący (1983, PC DOS 2.0)
- ❑ struktura danych opisująca podział dysku na partycje
- ❑ pierwszy sektor logiczny dysku (CHS → 0,0,1), zajmuje 512 bajtów

446 bajtów	4 × 16 = 64 bajty				2 bajty
Główny kod startowy	Tablica partycji				Sygnatura rozruchu
	Partycja 1	Partycja 2	Partycja 3	Partycja 4	

- ❑ **główny kod startowy (Master Boot Code, bootloader)** - program odszukujący i ładujący do pamięci zawartość pierwszego sektora aktywnej partycji
- ❑ **tablica partycji** - cztery 16-bajtowe rekordy opisujące partycje na dysku
- ❑ **sygnatura rozruchu (boot signature)** - znacznik końca MBR (**0x55AA**)
- ❑ maksymalny rozmiar partycji to **2 TB** ($2^{32} \times 512$ bajtów)

Struktura dysku twardego - GPT

- **GPT (GUID Partition Table)**, 2010 r.
- IBM/Microsoft, 2010 rok
- Maks. rozmiar dysku: **9,4 ZB**
- Maks. liczba partycji: **128**
- **Protective MBR** - pozostawiony dla bezpieczeństwa
- **GPT Header** (512 bajtów):
 - liczba pozycji i rozmiar pozycji w tablicy
 - położenie zapasowej kopii GPT
 - unikatowy identyfikator dysku
- **Entry x** (128 bajtów):
 - typ partycji, unikatowy identyfikator
 - początkowy i końcowy numer LBA
 - atrybuty, nazwa



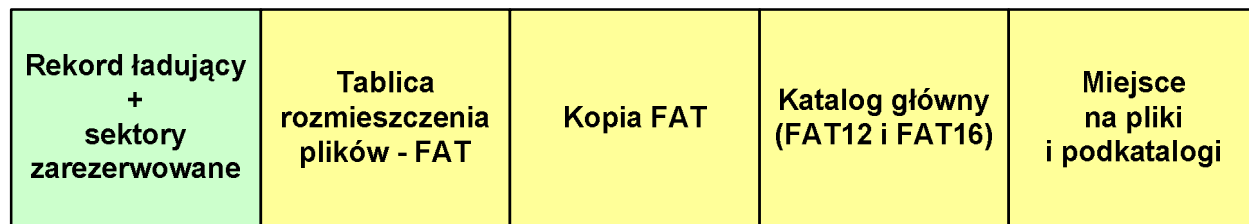
System plików FAT (File Allocation Table)

- 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 (praktycznie 28)
 - 64 bity w systemie exFAT (FAT64)
- ogólna struktura dysku logicznego / dyskietki w systemie FAT:

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

FAT12

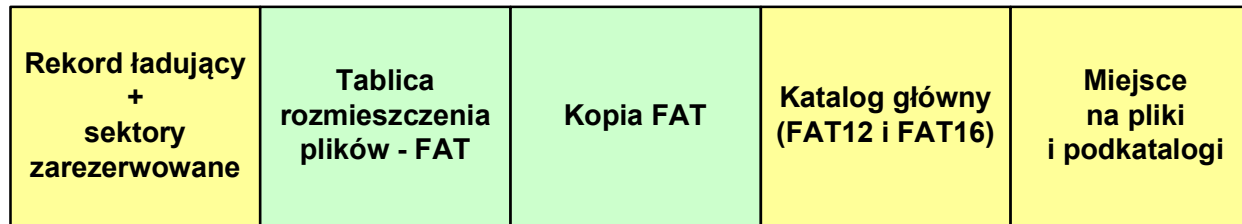
- system plików FAT12 przeznaczony jest dla nośników o małej pojemności
- **rekord ładujący** zajmuje pierwszy sektor dyskietki lub dysku logicznego



- 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

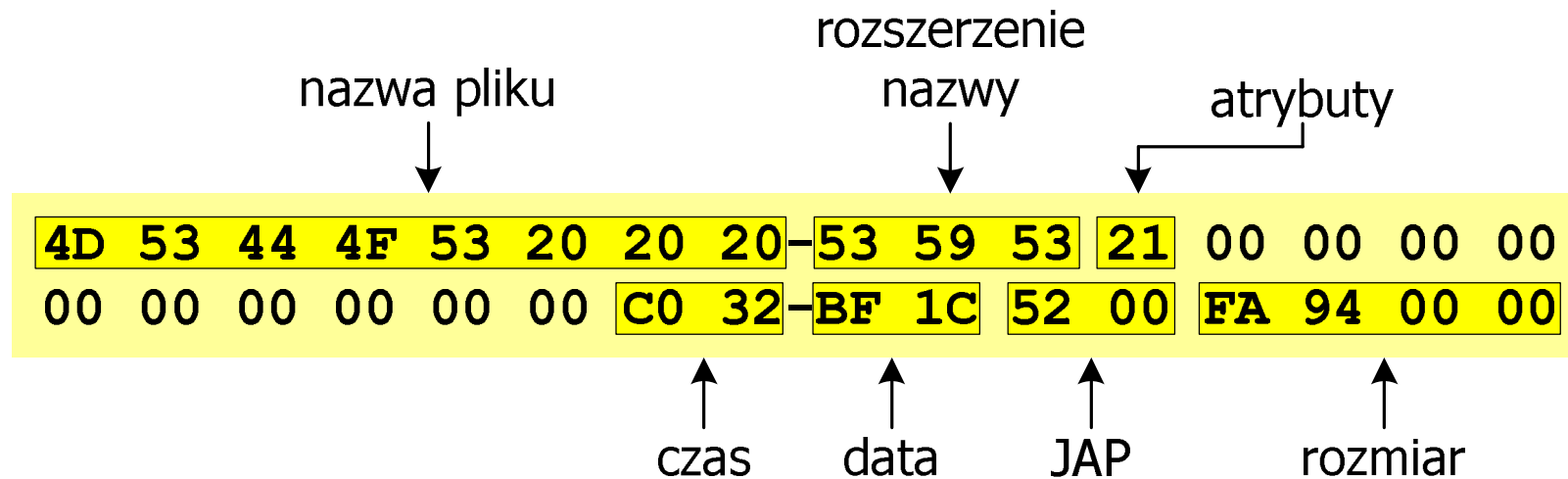


- katalog główny zawiera 32-bajtowe pola mogące opisywać pliki, podkatalogi lub etykietę dysku

FAT12

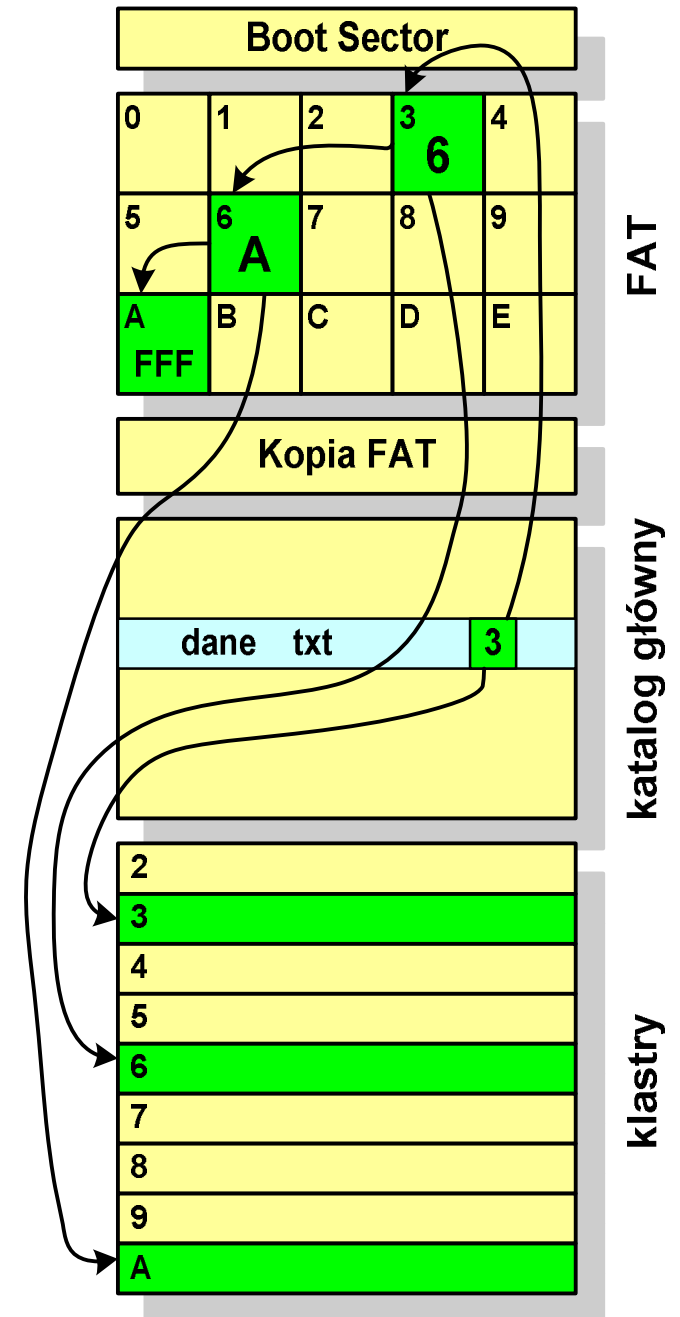
- przykładowa zawartość katalogu głównego:

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	00	C0	32	BF	1C	02	00	46	9F	002....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	00	C0	32	BF	1C	52	00	FA	94	002..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	00	C0	32	BF	1C	9D	00	75	D5	002....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	00	C0	32	BF	1C	08	01	C8	2B	002.....+..	



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



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 JAP ograniczona jest do 2^{16} czyli 65536
- maksymalny rozmiar dysku logicznego:
 - **DOS, Windows 95** - ok. 2 GB (gdyż maksymalny rozmiar JAP to 2^{15} bajtów)
 - **Windows 2000** - ok. 4 GB (gdyż maksymalny rozmiar JAP to 2^{16} bajtów)

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

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
- 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 (13 znaków w każdej 32-bajtowej strukturze, Unicode)
 - skrócona nazwa (32-bajtowa struktura, stary format 8+3)

długa nazwa pliku



0000	43	20	00	64	00	6F	00	6D-00	6F	00	0F	00	CF	77	00	C .d.o.m.o....w.
0010	61	00	2E	00	74	00	78	00-74	00	00	00	00	00	FF	FF	a...t.x.t.....
0020	02	63	00	79	00	6A	00	6E-00	65	00	0F	00	CF	20	00	.c.y.j.n.e....
0030	2D	00	20	00	70	00	72	00-61	00	00	00	63	00	61	00	-. .p.r.a...c.a.
0040	01	53	00	79	00	73	00	74-00	65	00	0F	00	CF	6D	00	.S.y.s.t.e...m.
0050	79	00	20	00	4F	00	70	00-65	00	00	00	72	00	61	00	y. .O.p.e...r.a.
0060	53	59	53	54	45	4D	7E	31-54	58	54	20	00	4B	03	80	SYSTEM~1TXT .K..
0070	67	32	67	32	00	00	08	80-67	32	02	00	06	00	00	00	g2g2....g2.....

skrócona nazwa pliku



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

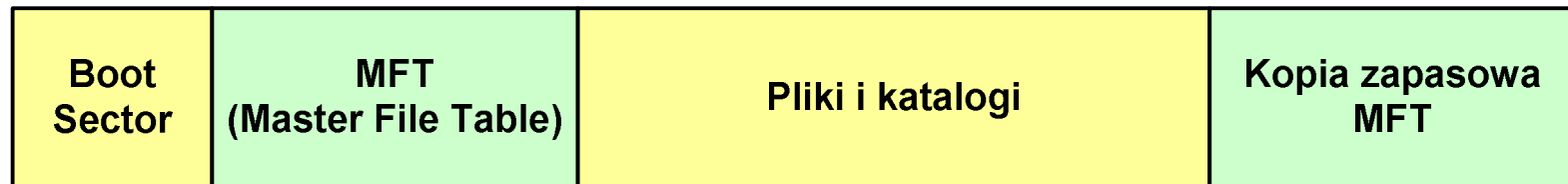
NTFS (New Technology File System)

- **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

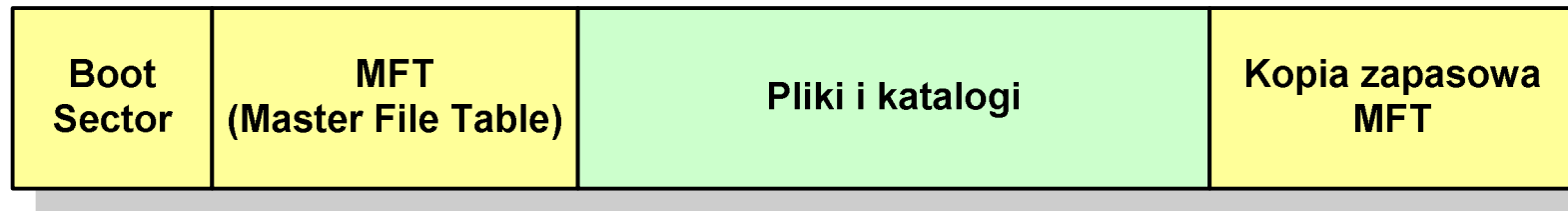
NTFS



- **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

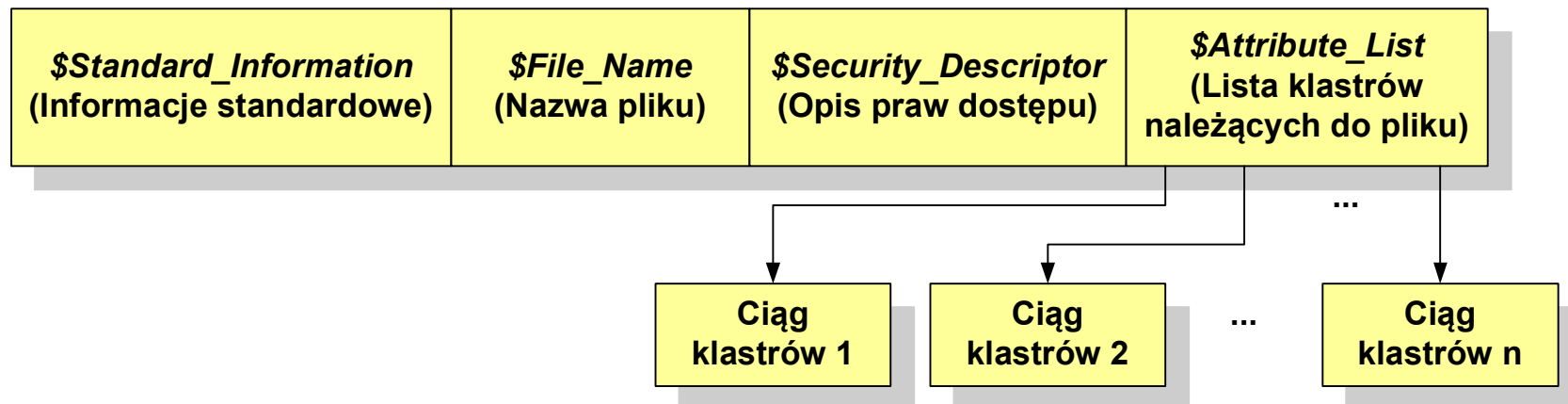
- pliki w systemie NTFS są reprezentowane w MFT przez rekord zawierający atrybuty:
 - **\$Standard_Information**
 - **\$File_Name**
 - **\$Security_Descriptor**
 - **\$Data**

<i>\$Standard_Information</i> (Informacje standardowe)	<i>\$File_Name</i> (Nazwa pliku)	<i>\$Security_Descriptor</i> (Opis praw dostępu)	<i>\$Data</i> (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**

<i>\$Standard_Information</i> (Informacje standardowe)	<i>\$File_Name</i> (Nazwa pliku)	<i>\$Security_Descriptor</i> (Opis praw dostępu)	<i>\$Index_Root</i>	<i>\$Index_Allocation</i>	<i>\$Bitmap</i>
--	--	--	----------------------------	----------------------------------	------------------------

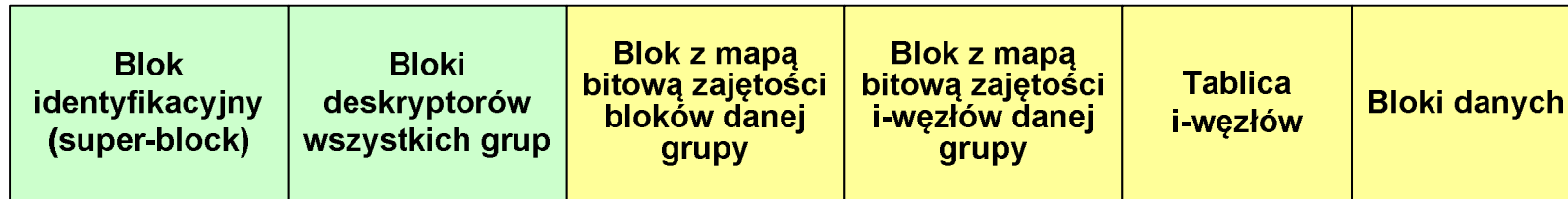
- zamiast atrybutu **\$Data** umieszczone są trzy atrybuty przeznaczone do tworzenia list, sortowania oraz lokalizowania plików i podkatalogów
 - **\$Index_Root**
 - **\$Index_Allocation**
 - **\$Bitmap**

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



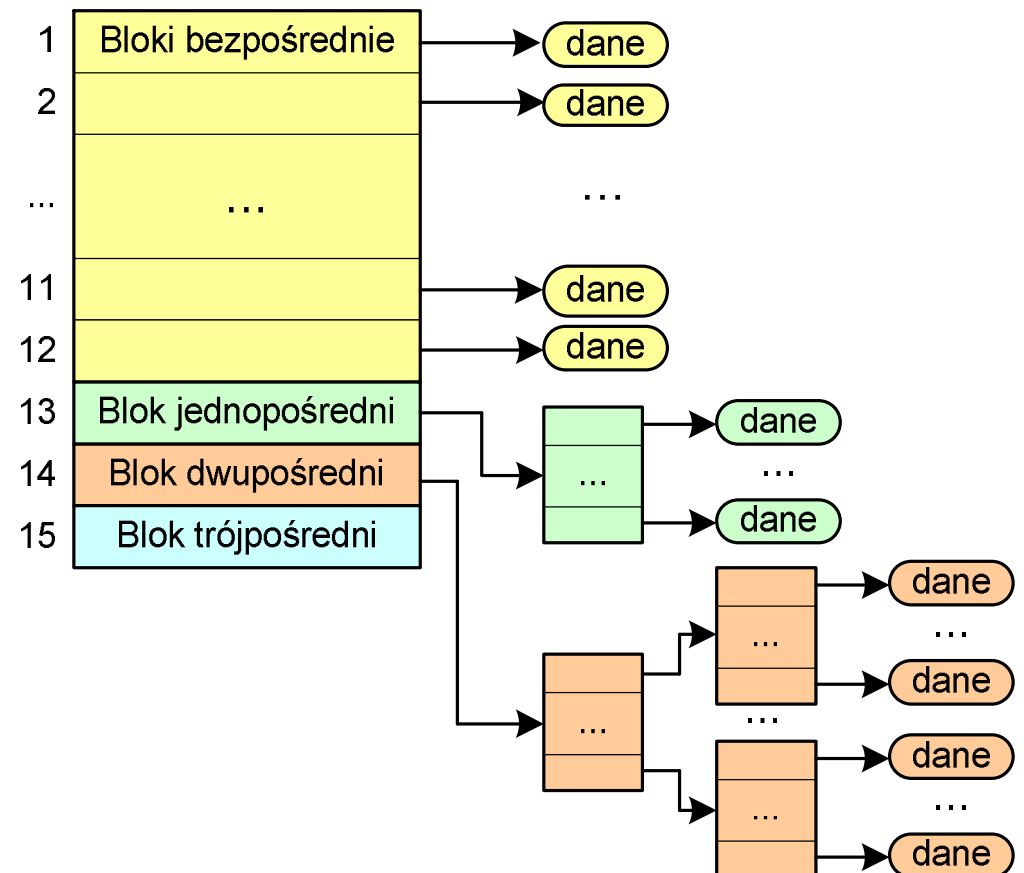
- każda grupa bloków zawiera ten sam blok identyfikacyjny oraz kopie bloków z deskryptorami wszystkich grup
- **blok identyfikacyjny** zawiera informacje na temat systemu plików (np. rodzaj systemu plików, rozmiar bloku)
- **deskryptor grupy** opisuje grupę bloków (np. położenie bloków z mapami bitowymi, liczba wolnych bloków, liczba katalogów w grupie)
- **blok z mapą bitową zajętości bloków danej grupy** - tablica bitów, zajmuje jeden blok (np. dla bloku o rozmiarze 1 kB opisuje 8096 bloków danych)
- **blok z mapą bitową zajętości i-węzłów danej grupy** - tablica bitów, każdy bit 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 jednopięśredni** (ang. single indirect block)
 - 1 adres bloku zawierającego adresy bloków jednopięśrednich - **blok dwupięśredni** (ang. double indirect block)
 - 1 adres bloku zawierającego adresy bloków dwupięśrednich - **blok trójpięśredni** (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)

Koniec wykładu nr 13

Dziękuję za uwagę!