

Informatyka 2

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

Wykład nr 5 (20.11.2017)

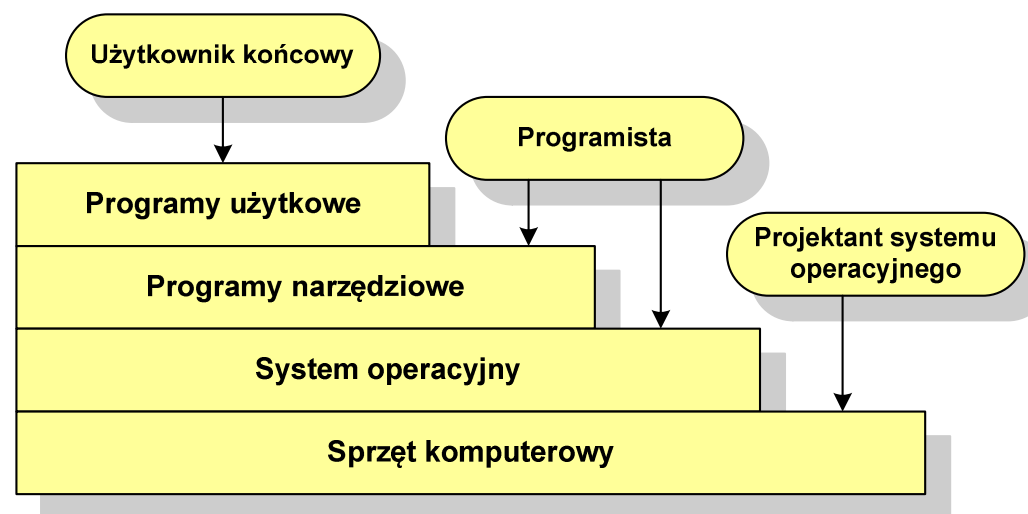
dr inż. Jarosław Forenc

Plan wykładu nr 5

- Definicje systemu operacyjnego
- Zarządzanie procesami
 - definicja procesu, blok kontrolny procesu
 - dwu- i pięciostanowy model procesu
- Zarządzanie dyskowymi operacjami we-wy
 - metody przydziału pamięci dyskowej
(alokacja ciągła, alokacja listowa, alokacja indeksowa)
 - struktura dysku twardego (MBR, GPT)
 - systemy plików (FAT)

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, bufory)
 - **kontekst wykonywanego programu** (stan procesu) - dane wewnętrzne, dzięki którym system operacyjny może nadzorować proces i nim sterować

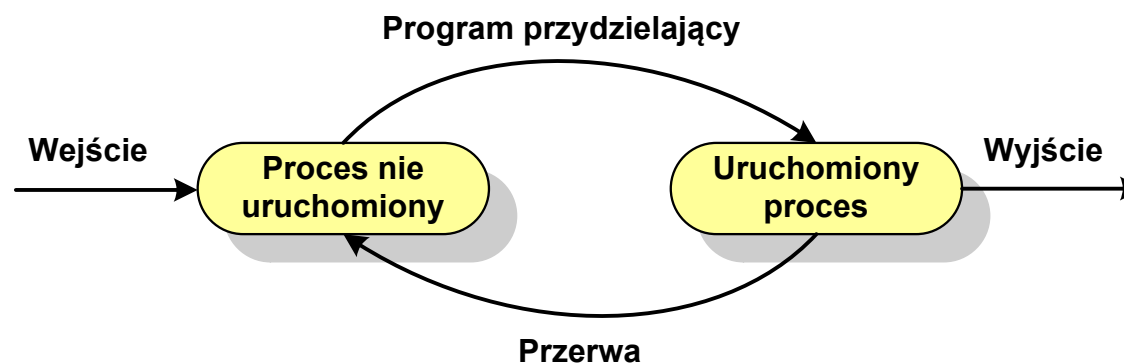
Blok kontrolny procesu (deskryptor procesu)

- 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

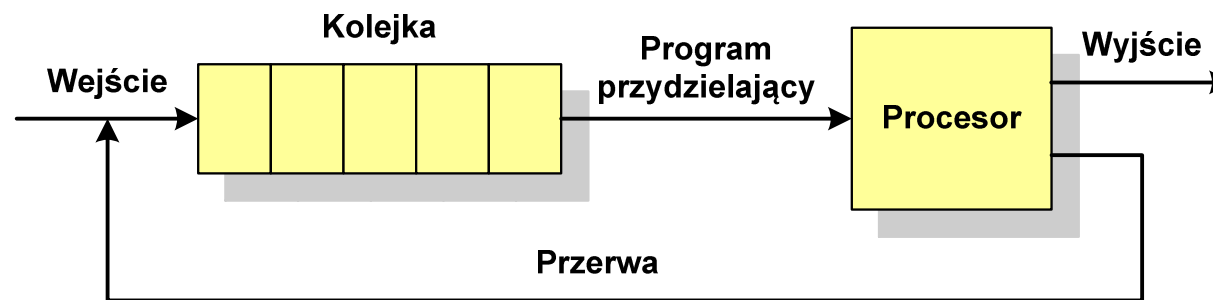
- najprostszy model polega na tym, że w dowolnej chwili proces jest wykonywany przez procesor (**uruchomiony**) lub nie (**nie uruchomiony**)



- 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

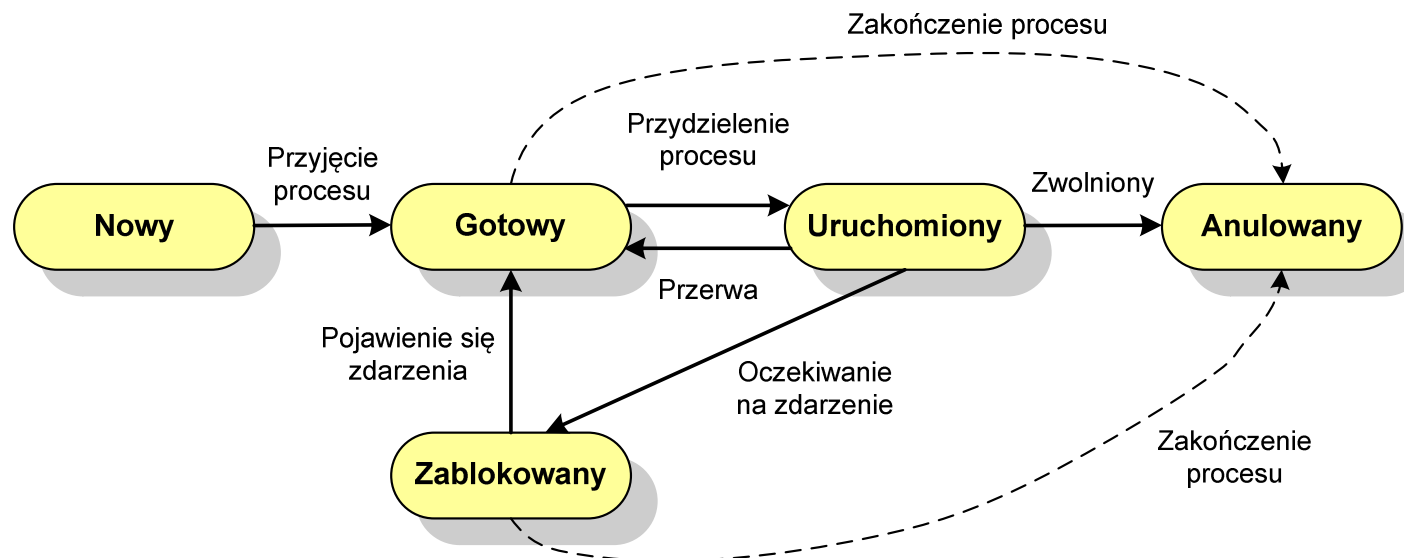
- procesy, które nie są uruchomione czekają w kolejce na wykonanie



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

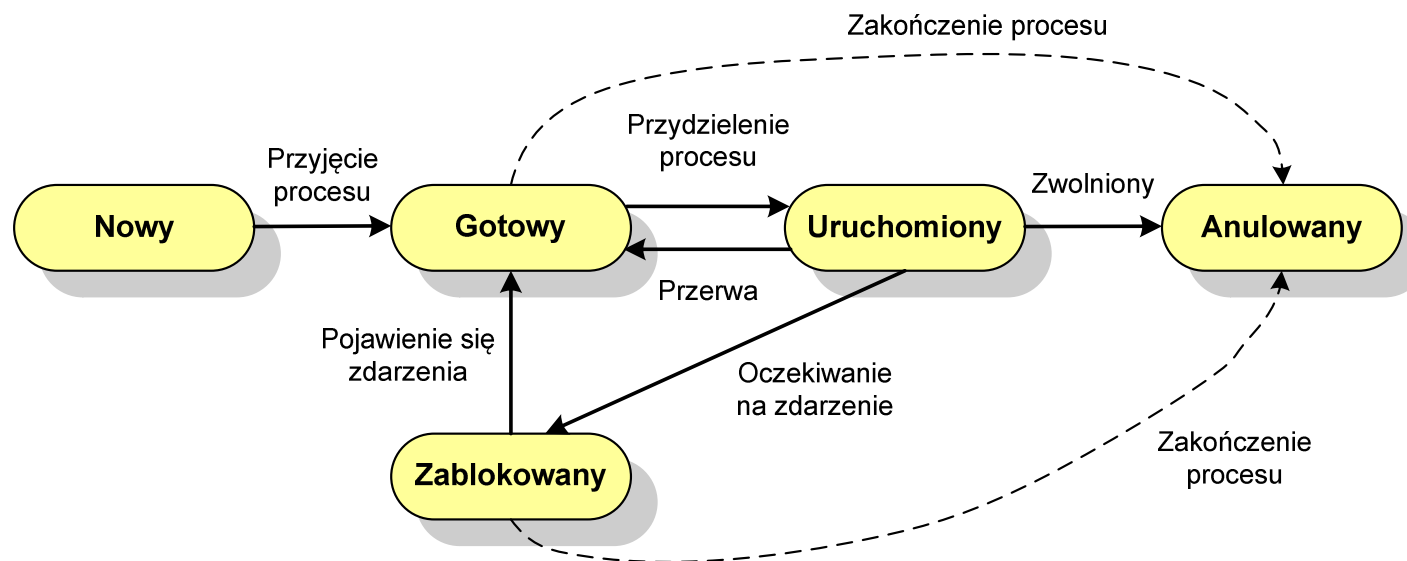
Pięciostanowy model procesu

- 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 nieuruchomionych na **gotowe do wykonania** i **zablokowane**



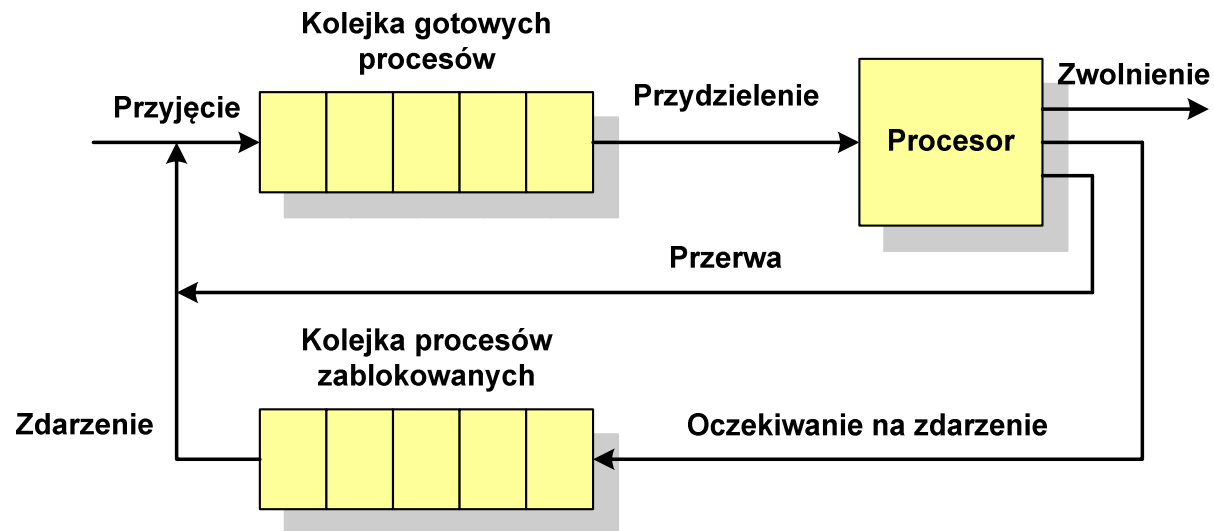
Pięciostanowy model procesu

- **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 nieuruchomionych 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

Zarządzanie dyskowymi operacjami we-wy

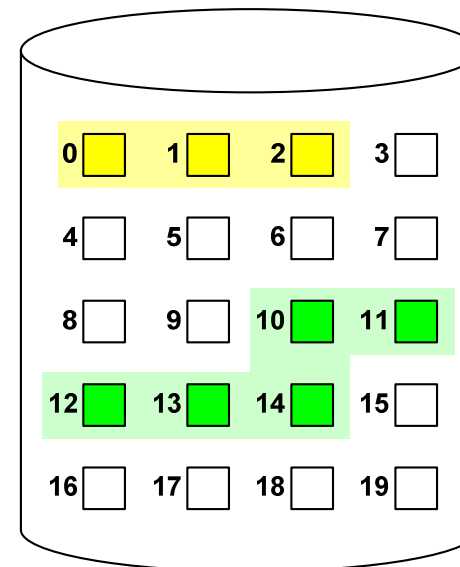
- Metody przydziału pamięci dyskowej (teoria)
 - alokacja ciągła
 - alokacja listowa
 - alokacja indeksowa

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

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

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

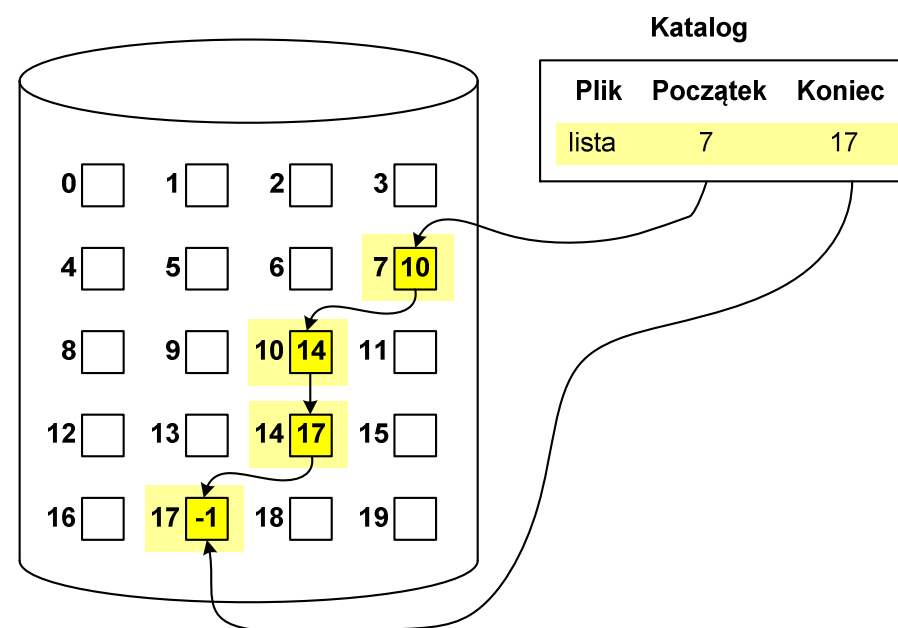


Katalog

| Plik | Początek | Długość |
|--------|----------|---------|
| lista | 0 | 3 |
| poczta | 10 | 5 |

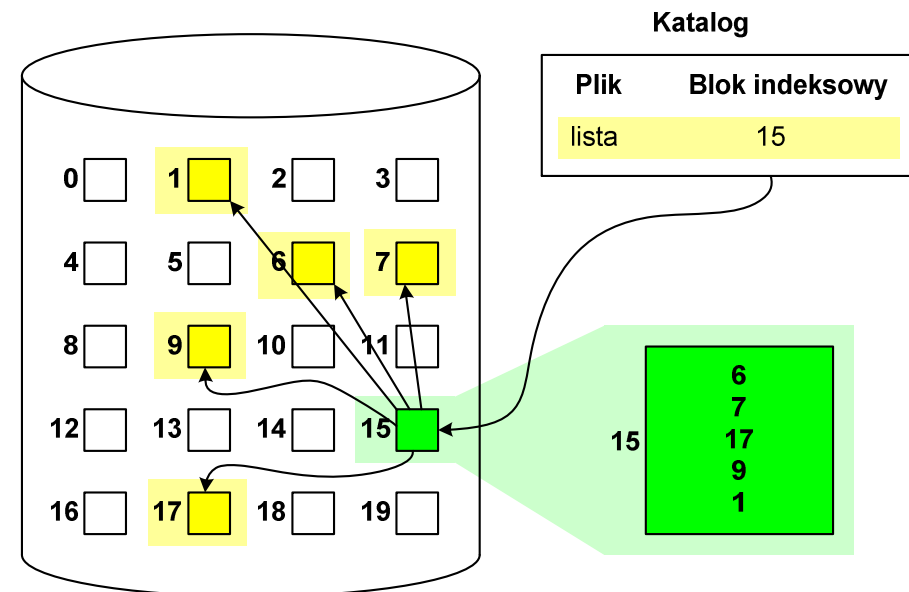
Przydział pamięci dyskowej - alokacja listowa

- 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



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 | $4 \times 16 = 64$ 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**)

Struktura dysku twardego - MBR (tablica partycji)

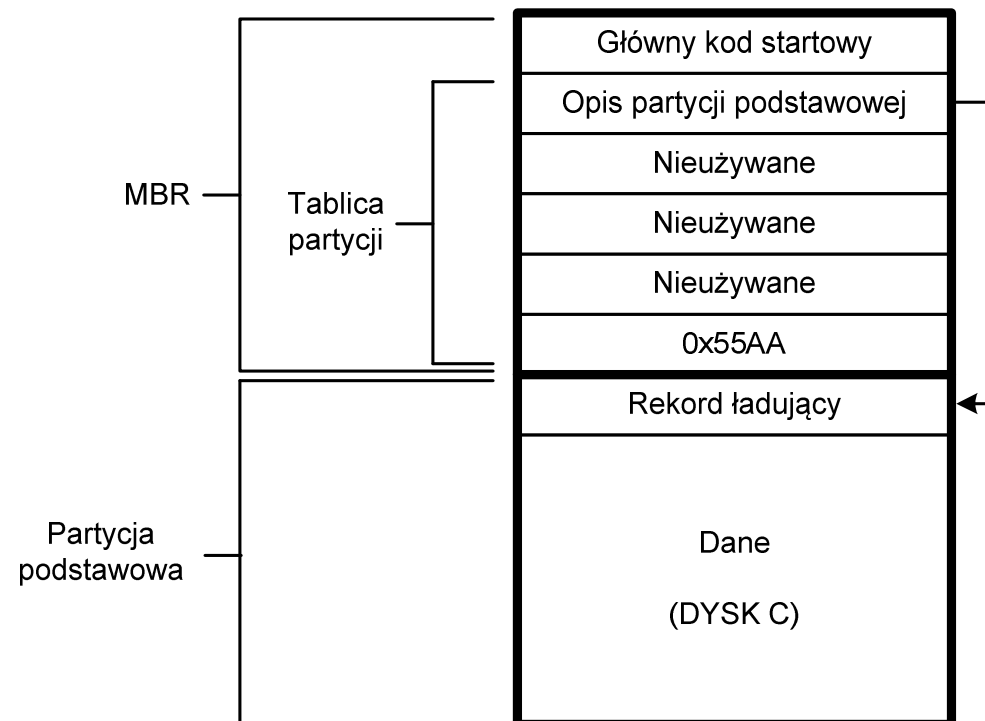
- zawartość rekordu w tablicy partycji

| Bajty | Rozmiar | Zawartość |
|---------|---------|--|
| 00H | 1 | Znacznik aktywności: 00H - nieaktywna, 80H - aktywna |
| 01H | 1 | Początek partycji: numer głowicy |
| 02H-03H | 2 | Początek partycji: numer cylindra i sektora |
| 04H | 1 | Typ partycji (system plików) |
| 05H | 1 | Koniec partycji: numer głowicy |
| 06H-07H | 2 | Koniec partycji: numer cylindra i sektora |
| 08H-0BH | 4 | Liczba sektorów: początek dysku → pierwszy sektor partycji |
| 0CH-0FH | 4 | Rozmiar partycji: liczba sektorów w partycji |

- zawartość i organizacja tablicy jest niezależna od systemu operacyjnego
- niewykorzystywany rekord zawiera same zera
- maksymalny rozmiar partycji to **2 TB** ($2^{32} \times 512$ bajtów)

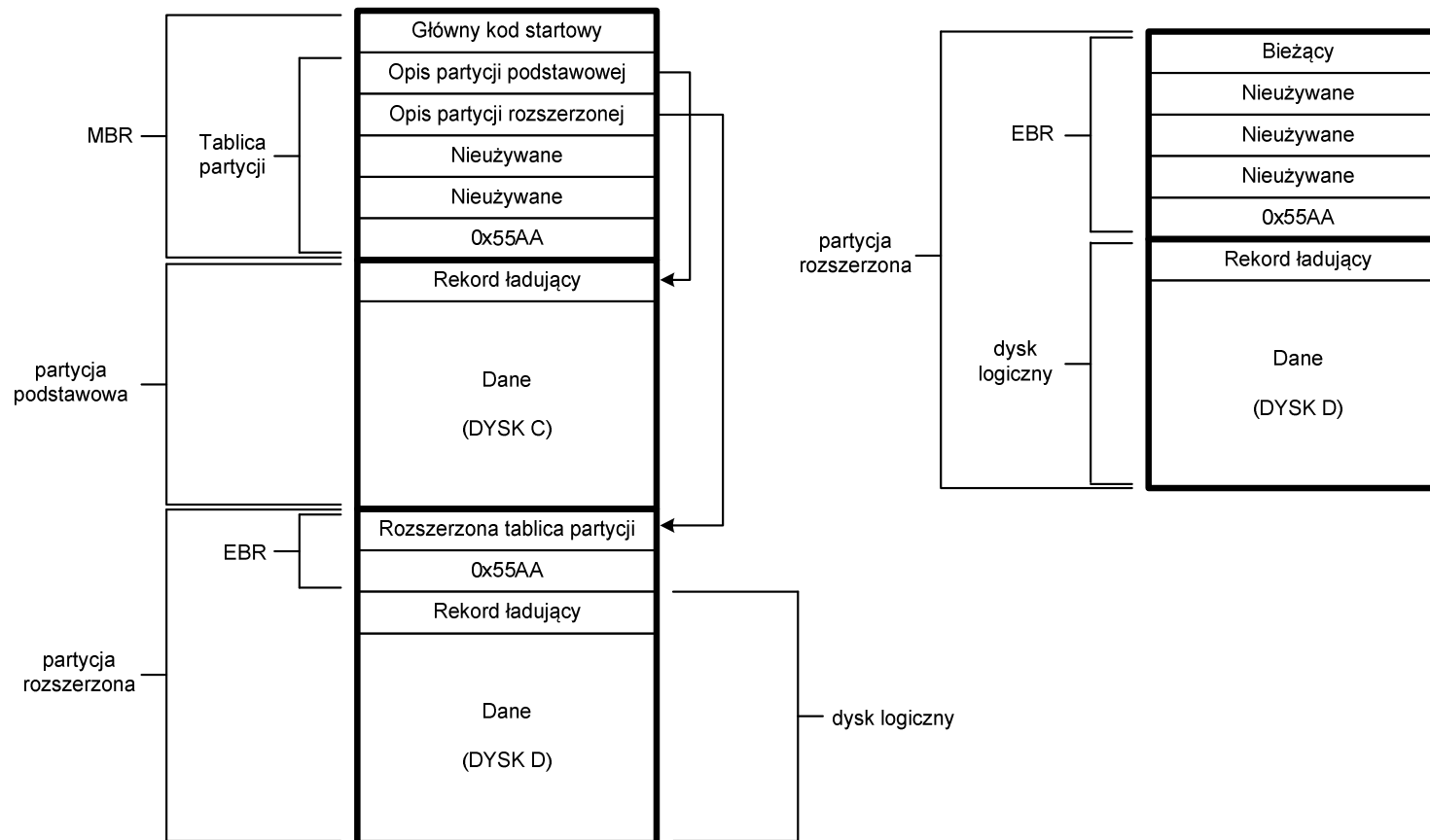
Struktura dysku twardego - MBR (tablica partycji)

- na dysku mogą znajdować się maksymalnie 4 **partycje podstawowe** (**primary partition**)
- każda partycja podstawowa może zawierać jeden **dysk logiczny**



Struktura dysku twardego - MBR (tablica partycji)

- w tablicy partycji można utworzyć jedną **partycję rozszerzoną** (**extended partition**), która może zawierać wiele dysków logicznych

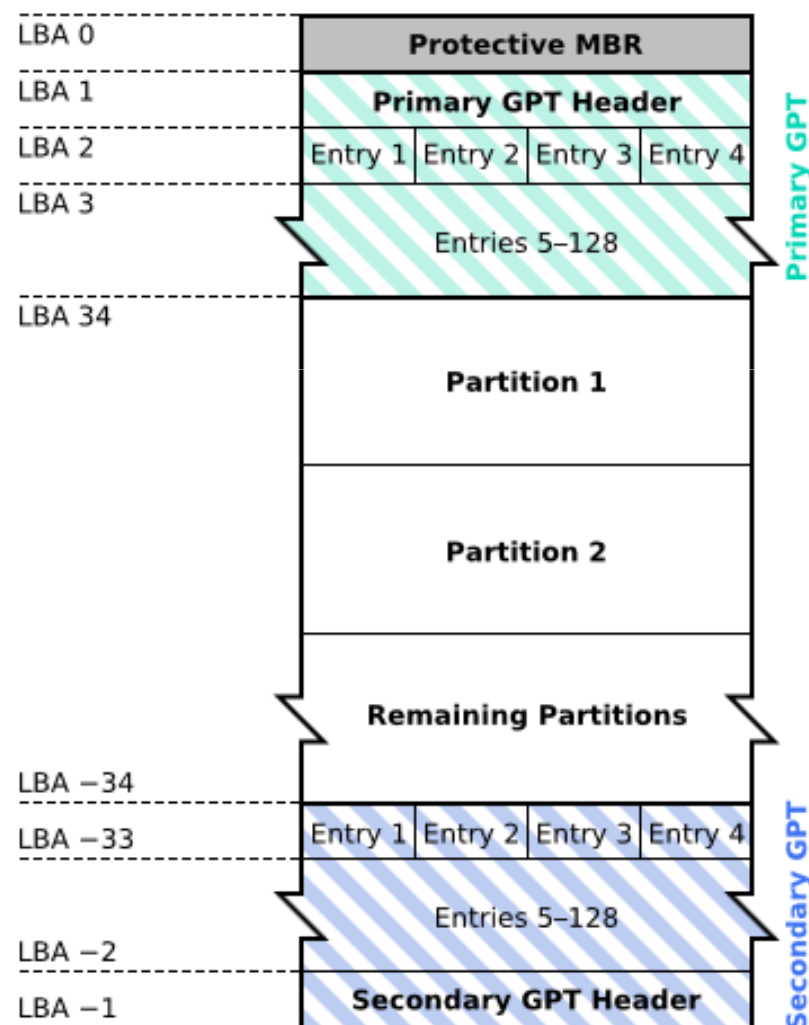


Struktura dysku twardego - GPT

- **GPT (GUID Partition Table)** - standard zapisu informacji o partycjach na dysku twardym
- **GUID (Globally Unique Identifier)** - 128-bitowa liczba stosowana do identyfikowania informacji w systemach komputerowych
- GPT to część standardu **UEFI (Unified Extensible Firmware Interface)**, który zastąpił BIOS w komputerach PC (interfejs graficzny, obsługa myszki)
- opracowanie: IBM/Microsoft, 2010 rok
- maksymalny rozmiar dysku to **9,4 ZB** (2^{64} sektorów \times 512 bajtów)
- możliwość utworzenia do 128 partycji podstawowych

Struktura dysku twardego - GPT (struktura)

- **Protective MBR** - pozostawiony dla bezpieczeństwa
- **GPT Header** (512 bajtów):
 - liczba pozycji w tablicy partycji
 - rozmiar pozycji w tablicy partycji
 - położenie zapasowej kopii GPT
 - unikatowy identyfikator dysku
 - sumy kontrolne
- **Entry x** (128 bajtów):
 - typ partycji
 - unikatowy identyfikator
 - początkowy i końcowy numer LBA
 - atrybuty
 - nazwa



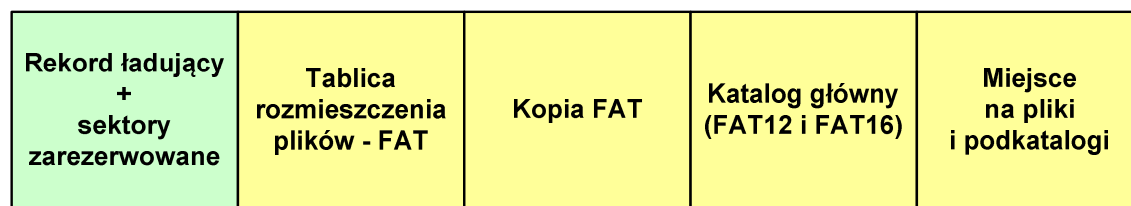
System plików FAT (File Allocation Table)

- 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
 - 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

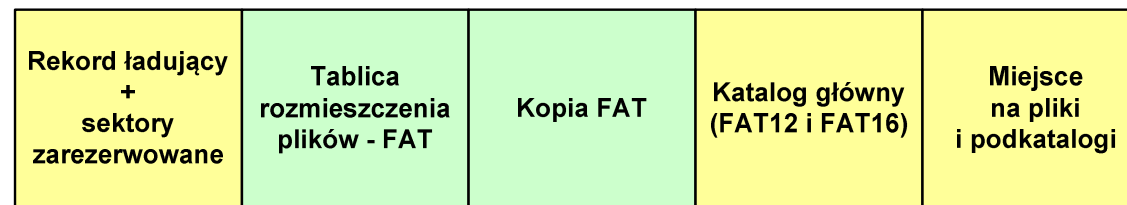
- system plików FAT12 przeznaczony jest dla nośników o małej pojemności
- obsługuje $2^{12} = 4096$ jednostek alokacji, max. rozmiar partycji to 16 MB
- **rekord ładujący** zajmuje pierwszy sektor dyskiety lub dysku logicznego



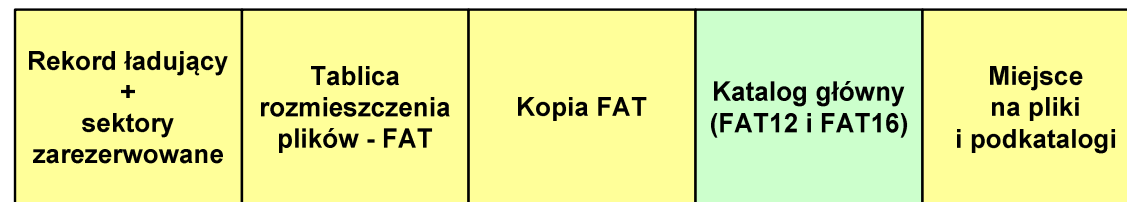
- rekord ładujący zawiera następujące dane:
 - instrukcja skoku do początku programu ładującego (3 bajty)
 - nazwa wersji systemu operacyjnego (8 bajtów)
 - struktura BPB (ang. BIOS Parametr Block) - blok parametrów BIOS (25 bajtów)
 - rozszerzony BPB (ang. Extended BPB, 26 bajtów)
 - wykonywalny kod 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

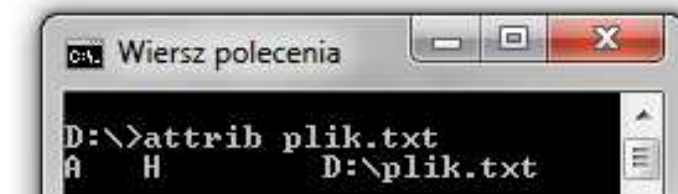
- katalog główny zawiera 32-bajtowe pola mogące opisywać pliki, podkatalogi lub etykietę dysku

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 |



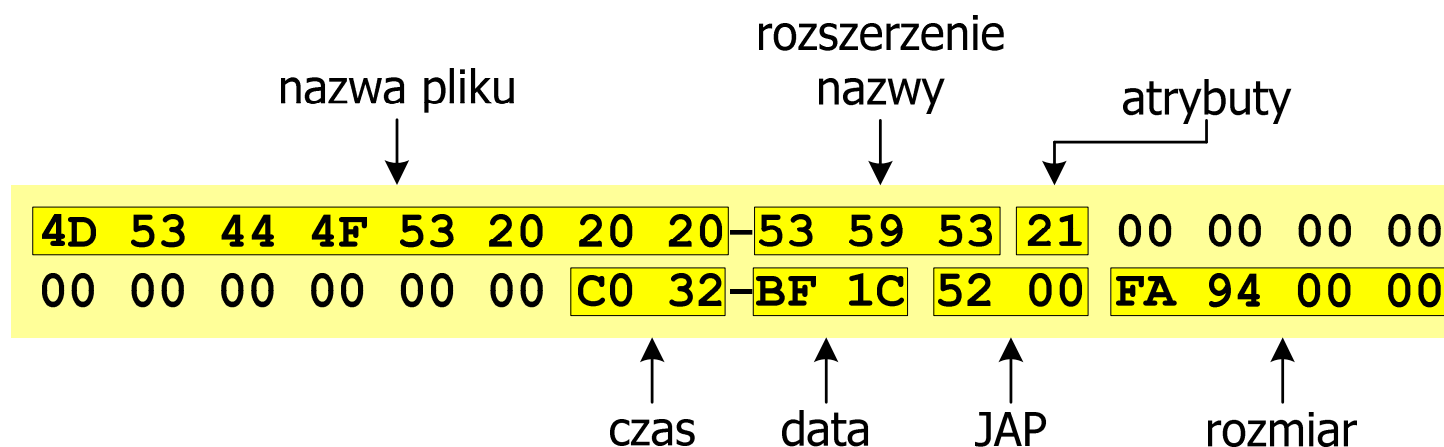
```
Wiersz polecenia
D:\>attrib plik.txt
A H D:\plik.txt
```

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 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 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 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 00 C0 32-BF 1C 08 01 C8 2B 00 00  .....2.....+..
    
```



FAT12

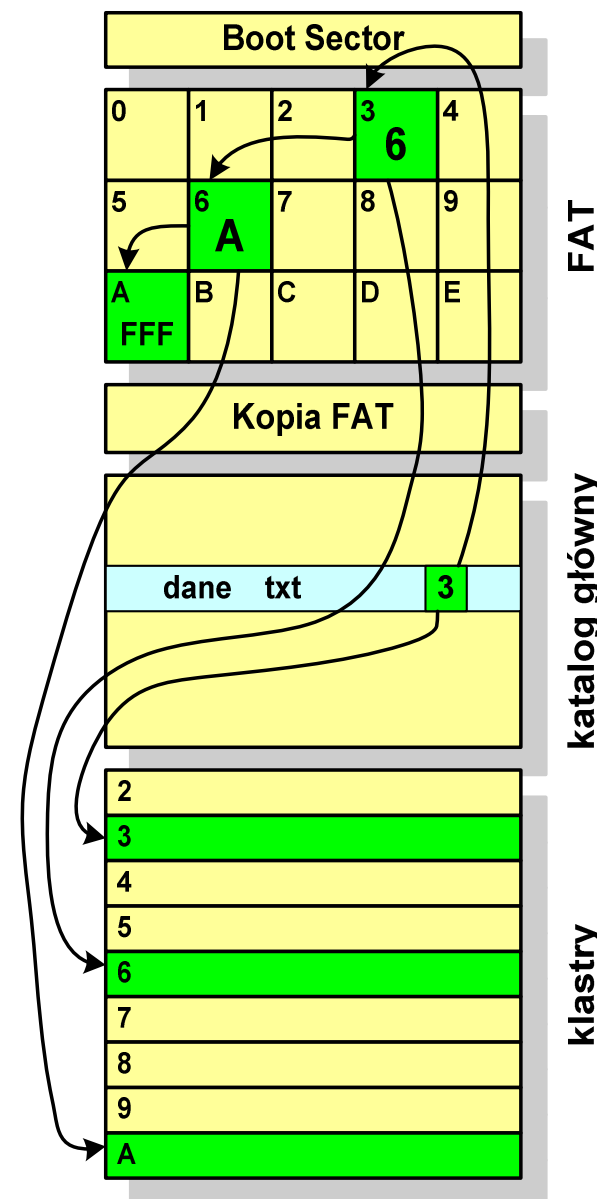
- pozostałą część dysku zajmuje miejsce na pliki i podkatalogi

| | | | | |
|--|---|-----------|-----------------------------------|--------------------------------------|
| Rekord ładujący + sektory zarezerwowane | Tablica rozmieszczenia plików - FAT | Kopia FAT | Katalog główny (FAT12 i FAT16) | 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 65536×32 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 (niedozwolone 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

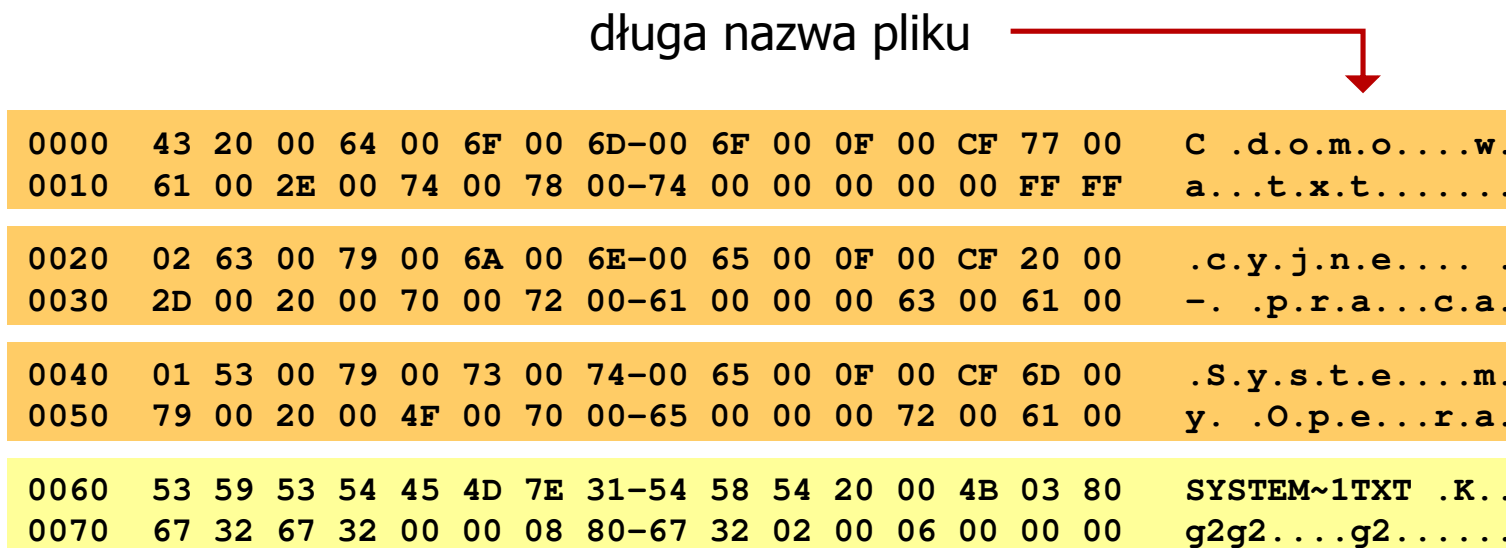
- **skrótowa nazwa pliku** przechowywana jest w identycznej, 32-bajtowej, strukturze jak w przypadku plików w starym formacie 8+3
- **długie nazwy plików** zapisywane są także w 32-bajtowych strukturach, przy czym jedna nazwa zajmuje kilka struktur (w jednej strukturze umieszczonych jest 13 kolejnych 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 klastra (zawsze 0) |
| 1CH-1FH | 4 | Dwie ostatnie litery nazwy |

FAT32 - długie nazwy plików

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

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 | . | . | . | . | . | . | . | |
| 0040 | 01 | 53 | 00 | 79 | 00 | 73 | 00 | 74 | 00 | 65 | 00 | 0F | 00 | CF | 6D | 00 | . | S | . | y | . | s | . | t | . | e | . | . | . | . | . | |
| 0050 | 79 | 00 | 20 | 00 | 4F | 00 | 70 | 00 | 65 | 00 | 00 | 00 | 72 | 00 | 61 | 00 | y | . | . | O | . | p | . | e | . | . | . | . | . | . | . | |
| 0060 | 53 | 59 | 53 | 54 | 45 | 4D | 7E | 31 | 54 | 58 | 54 | 20 | 00 | 4B | 03 | 80 | S | Y | S | T | E | M | ~ | 1 | T | X | T | . | K | . | . | |
| 0070 | 67 | 32 | 67 | 32 | 00 | 00 | 08 | 80 | 67 | 32 | 02 | 00 | 06 | 00 | 00 | 00 | g | 2 | g | 2 | . | . | . | . | . | . | . | . | . | . | . | . |

skrótowa nazwa pliku

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 7/8/10, Windows Server 2003/2008, 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

Koniec wykładu nr 5

Dziękuję za uwagę!

(Następny wykład: 04.12.2017)