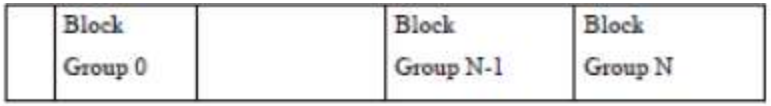
Implementacje systemów plików

Ostatnia modyfikacja: 27.05.2021

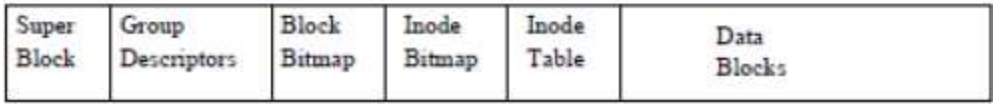
Spis treści

- Linux/Unix
  - ext2, ext3, ext4, proc, swap
- MS Windows – FATxx, exFAT, NTFS
- ZFS/OpenZFS
- Zarządzanie dyskami

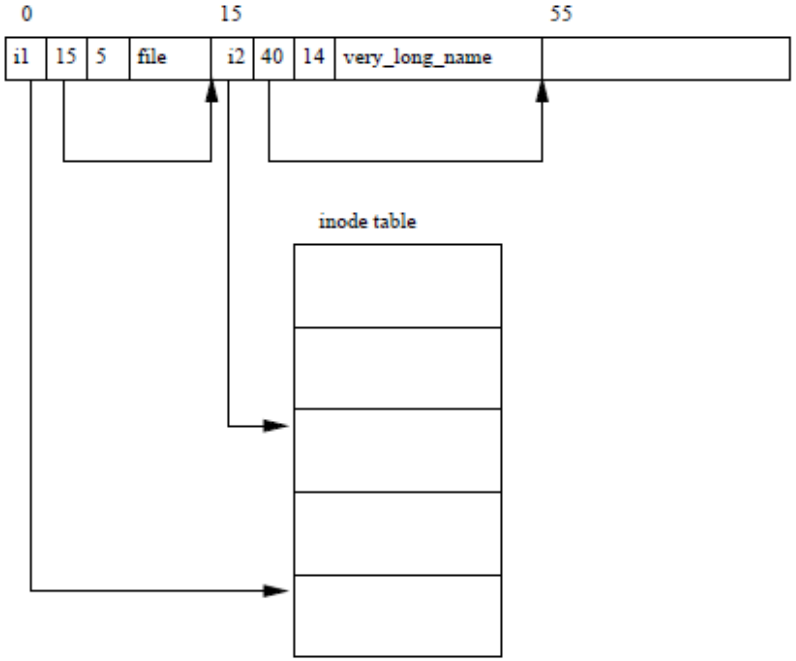
Linux: system plików ext2



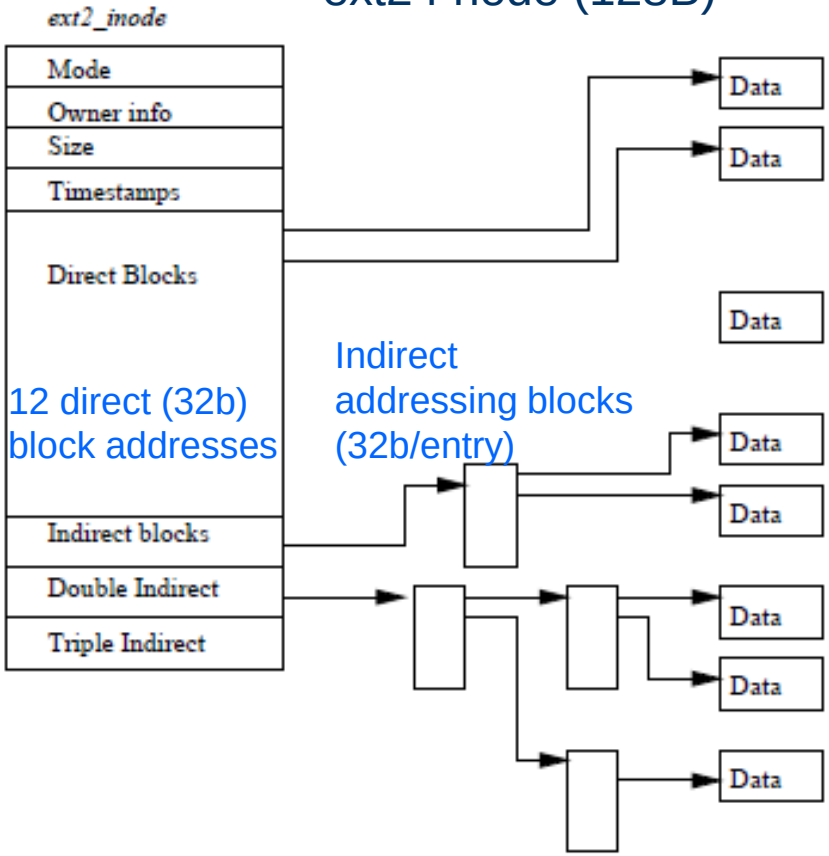
– Physical layout of ext2 file system



ext2 directory



ext2 i-node (128B)



Systemy plików ext3 i ext4

- **ext3** jest bodaj najczęściej stosowanym systemem plików systemu Linux
  - Zastąpił starszy system plików **ext2** ; utrzymany max. rozmiar pliku: 16GiB-2TiB i systemu plików: 2-32TiB
  - Do znajdowania bloków pliku używany jest podobny mechanizm jak w BSD Fast File System (FFS) . Główne różnice dotyczą sposobu przydziału bloków
    - FFS przydziela bloki dyskowe o rozmiarze 8kB, następnie dzielone na fragmenty o rozmiarze 1kB do przechowywania małych plików, bądź końcówek większych plików. ext3 nie używa fragmentów bloków, ale przydziela mniejsze bloki dyskowe: 1, 2, 4, 8kB (zależnie od rozmiaru systemu plików)
    - ext3 przydziela klastry bloków, aby sąsiednie bloki pliku były przechowywane na dysku w pobliżu; można wtedy używać operacji wej/wyj na **grupie bloków**.
    - Mapa wolnych bloków przechowywana jest w grupie bloków (a nie w grupie cylindrów, jak w FFS). Szukane jest miejsce wolne dla 8 wolnych bloków na raz (***pre-allocation***).
  - Katalogi mogą wykorzystywać „hashed B-trees”.
- **ext4** jest rozwinięciem ext3 (można montować systemy ext2 i ext3 jako ext4). :
  - Maks. Wielkość systemu plików:1 exbibyte (Eib), i.e. 2^{60} B, ok... 1.15 10^{18}B,
  - Pliki do 16 tebibytes (TiB) 2^{40} B, ok. 17.6 TB (dla 4 kB bloków)
  - Nieograniczona liczba podkatalogów (ext3: 32000)
  - Transparentne szyfrowanie
  - System wykorzystuje fragmenty nadmiarowe plików (***extents***)

Systemy plików z dziennikiem

- W systemach **ext3, ext4** zrealizowano modyfikacje systemu plików z dziennikiem; modyfikacje są realizowane transakcyjnie, dziennik przechowuje plan zmian
  - Zmiany zapisane w dzienniku traktuje się jako zatwierdzone (*committed*) – operacja plikowa: pomyślna
  - Stopniowo zapisy w logu są odtwarzane, a zmiany zapisywane na dysku
- W razie awarii systemu niektóre zapisy mogą być ciągle w dzienniku
  - Muszą być zrealizowane po powrocie systemu do sprawności
  - Nie ma dodatkowego sprawdzania spójności systemu plików po awarii (powrót jest szybszy niż dla innych implementacji)
- Szybkość zapisu jest poprawiana przez zastępowanie operacji zapisu bezpośredniego (*random* I/O) przez zapis sekwencyjny (do dziennika).
- Poziomy obsługi w trybie transakcyjnym (patrz man ext4(5))
  - **Journal** (lowest risk). Both metadata and file contents are written to the journal before being committed to the main file system.
  - **Ordered** (medium risk, default). Only metadata is journaled; file contents are not, but it's guaranteed that file contents are written to disk before associated metadata is marked as committed in the journal.
  - **Writeback** (highest risk).. Only metadata is journaled; file contents are not.
- Dziennik można umieścić na oddzielnym urządzeniu.

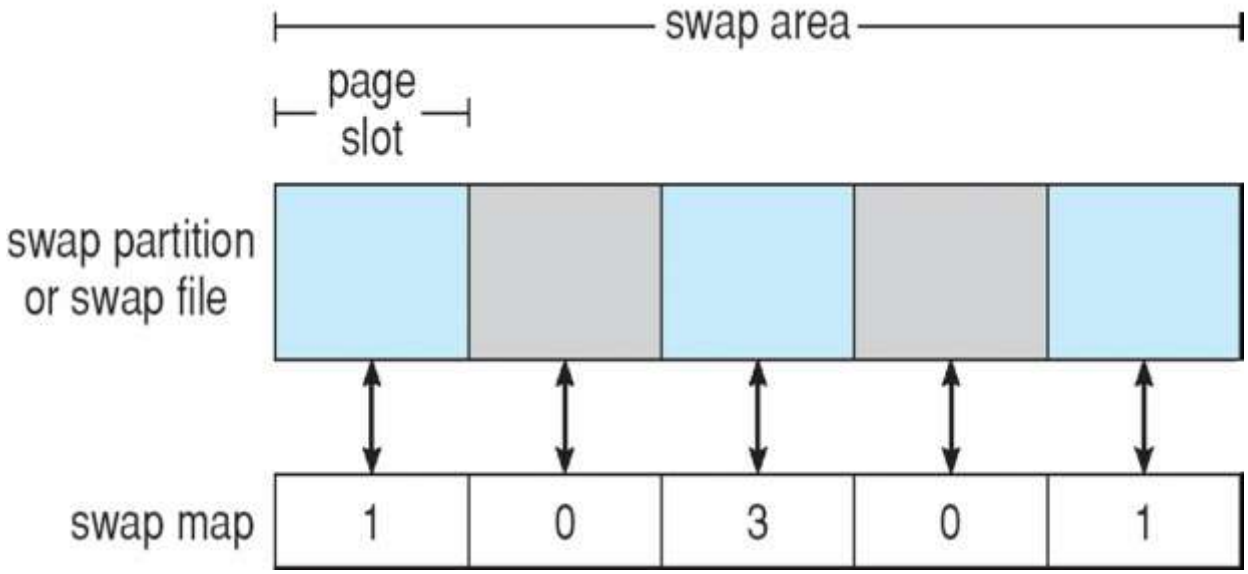
System plików *proc* w SO Linux

- System plików **proc** (patrz: man proc(5)) nie przechowuje danych; jego zawartość jest wyznaczana na żądanie procesów (wywołanie odpowiednich funkcji wejścia/wyjścia)
- **proc** tworzy drzewiastą strukturę katalogową; każdy plik i katalog posiada więc własny **i-węzeł**
  - i-węzeł określa jakie operacje trzeba wykonać, jeśli proces wywoła funkcję systemową związaną z tym i-węzłem. Informacje związane z procesem o danym PID związane są z i-węzłami o numerach np. postaci PID*2^16+id, gdzie 16-bitowy id określa rodzaj informacji. Informacje globalne mają numery i-węzłów jak proces o PID=0.
  - Zmienne jądra są dostępne w katalogu **/proc/sys**
  - W przypadku operacji odczytu kod obsługujący system plików **proc** gromadzi wymagane informacje, formatuje w postaci tekstu i umieszcza w buforze funkcji czytającej

Przestrzeń wymiany

- **Wymiana** (*swapping*) – ściśle biorąc polega na przesyłanie obrazów procesów pomiędzy pamięcią główną, a obszarem wymiany. W niektórych systemach operacyjnych terminy: wymiana oraz stronicowanie nie są rozróżniane, a przestrzeń wymiany jest używana do obydwóch celów. Przestrzeń wymiany (*swap-space*) — zestaw obiektów/obszarów pamięci fizycznej wykorzystywanych przez mechanizm pamięci wirtualnej jako rozszerzenie pamięci głównej.
- **Przestrzeń wymiany** może być utworzona w zwykłym systemie plików (plik wymiany), lub może być oddzielną partycją dysku.
- Zarządzanie przestrzenią wymiany w systemach Unixowych
  - 4.3BSD przydziela przestrzeń wymiany przy starcie procesu; zawiera segmenty kodu i danych
  - Jądro używa map wymiany (*swap maps*), aby śledzić użytkowanie przestrzeni wymiany
  - Solaris 2 przydziela przestrzeń wymiany dla procesu dopiero kiedy strona jest usuwana z pamięci fizycznej, a nie gdy strona pamięci wirtualnej została utworzona
  - Linux – wykorzystuje przestrzeń wymiany zarówno do wymiany jak i do stronicowania (man mkswap(8), swapon(8))

Przestrzeń wymiany w systemie Linux



- Przestrzeń wymiany jest używana jedynie na anonimowe obszary pamięci oraz na pamięć dzieloną pomiędzy procesy.
- W systemie może być więcej niż jeden obszar wymiany: plik wymiany czy partycja wymiany (man mkswap(8), swapon(8)). Lista w: ***/proc/swaps***
- Obszary wymiany składają się z bloków o rozmiarze 4kB (*page slots*).
- Mapa wymiany (*swap map*): tablica liczników odwzorowań danego bloku

FAT

	FAT12	FAT16	FAT16B	FAT32
Cluster addr/ FAT entry sz	12b/ 12b	16b/ 16b	16b / 32b	28b / 32b
Max.vol./ cluster sz	32MB/ 8KB	32MB	2-16GB/ 32-256KB	2TB / 512B 16TB/64KB (4KB sectors)
Max. file	Vol.sz	Vol.sz	2/4GB-1	2/4GB-1
Max.files/cluster sz	4068/ 8KB	65536 / 32KB	65540/ 32KB	268173300/32KB

Struktura logiczna partycji FAT

Reserved sectors
FAT #1
FAT #2
Root directory
Data area

- Directory entry (32B long):
  - File pointer: 8.3 format name, 1st data cluster nr, file length, attributes (RHSA), timestamp
  - Subdirectory pointer (atrib.:dir)
  - Volume label (attrib: label)
  - VFAT Long File Name (1 of max. 5 parts of a long file name: up to 13*2B chars)

MS exFAT

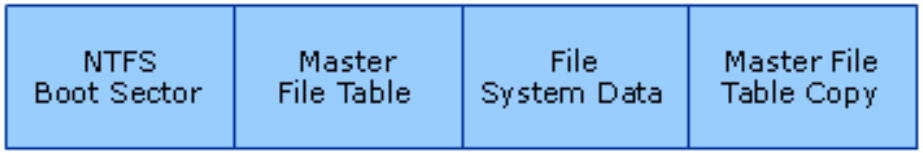
- System plików dla kart pamięci (def. dla SP kart SDXC >=32GB); dokumentacja publicznie dostępna od 2019 (MS); 2013 – drajwer Samsunga dla Linuxa (GPL)
- Cechy szczególne:
  - Maks. rozmiar pliku: 16EiB (2^64-1); partycji 128PiB (2^57-1), zal.: 512TB
  - Do 2796200 plików/katalog
  - Maks. liczba plików w systemie plików: 2^32-1
  - Rozmiar klastra <= 32MiB
  - Bitowa mapa wolnych klastrów
  - Alokacja wstępna plików i klastrów
    - 1 bit określa status rezerwacji każdego klastra
    - 1 bit w pozycji katalogowej pliku wskazuje, że plik ma ciągłą alokację (nie potrzeba używać FAT, by znaleźć kolejne klastry pliku)
    - Rekord katalogu zawiera wpisy: valid space oraz allocated space (*pre-allocation*)
  - LFN (brak 8.3); hash-based lookup (opatentowany)
  - Pojedyncza tablica FAT, ale integralność metadanych chroniona przez sumy kontrolne

System plików NTFS

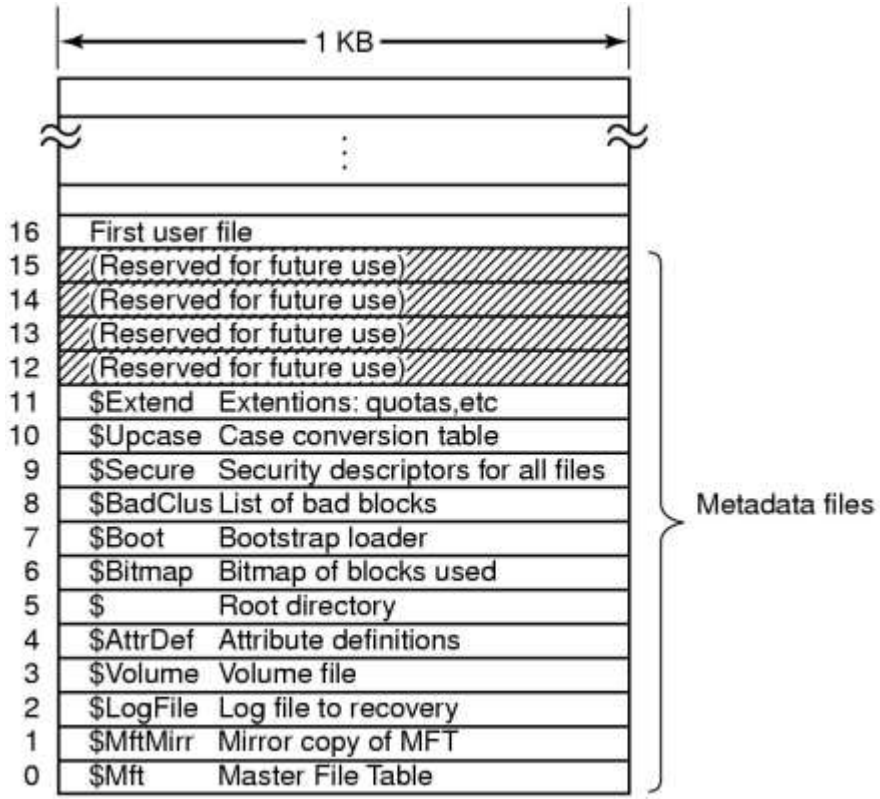
- Podstawową jednostką przechowywania danych masowych w systemach MS Windows wspierających system plików NTFS jest **wolumin (tom)**
  - Tworzony jest przez narzędzie *Zarządzanie dyskami*
  - Wykorzystuje logiczny podział dysku
  - Może wykorzystywać część dysku, cały dysk albo kilka dysków.
- Wszystkie *metadane*, jak informacje o woluminie, są przechowywane w zwykłym pliku
- Jednostkami przydziału pamięci dysku są **klastry** (grona)
  - Klaster tworzy 2^n sektorów dysku
  - Rozmiar klastrów NTFS jest mniejszy niż dla systemu FAT 16 – mniejsze są straty pamięci wywołane fragmentacją wewnętrzną.
- Maks. rozmiar woluminu: teoret. 2^{64}-1 klastrów, Win10./1709: 8PB-2MB: <=Win10/1703: 256TiB-64KiB, 1TiB=1024^3; maks. rozmiar pliku: : Win10./1709: 8PB-2MB' Win7: 16TiB-64KiB. Maks. Liczba plików: 2^{32}-1.

System plików NTFS

- Fizyczna organizacja woluminu NTFS



- NTFS adresuje nośnik przy pomocy **logicznych numerów klastrów** (logical cluster numbers, LCNs).
- Wszystkie metadane tomu są przechowywane w plikach:
  - MFT
  - kopia 16 pozycji MFT
  - plik dziennika (*$LogFile*), dziennika zmian (*$UsnJrnl*),
  - inne pliki: tomu, tablica definicji atrybutów, katalog główny, plik z mapą bitów, plik rozruchowy, plik popsutych klastrów.



Tanenbaum, Modern Operating Systems, Prentice Hall

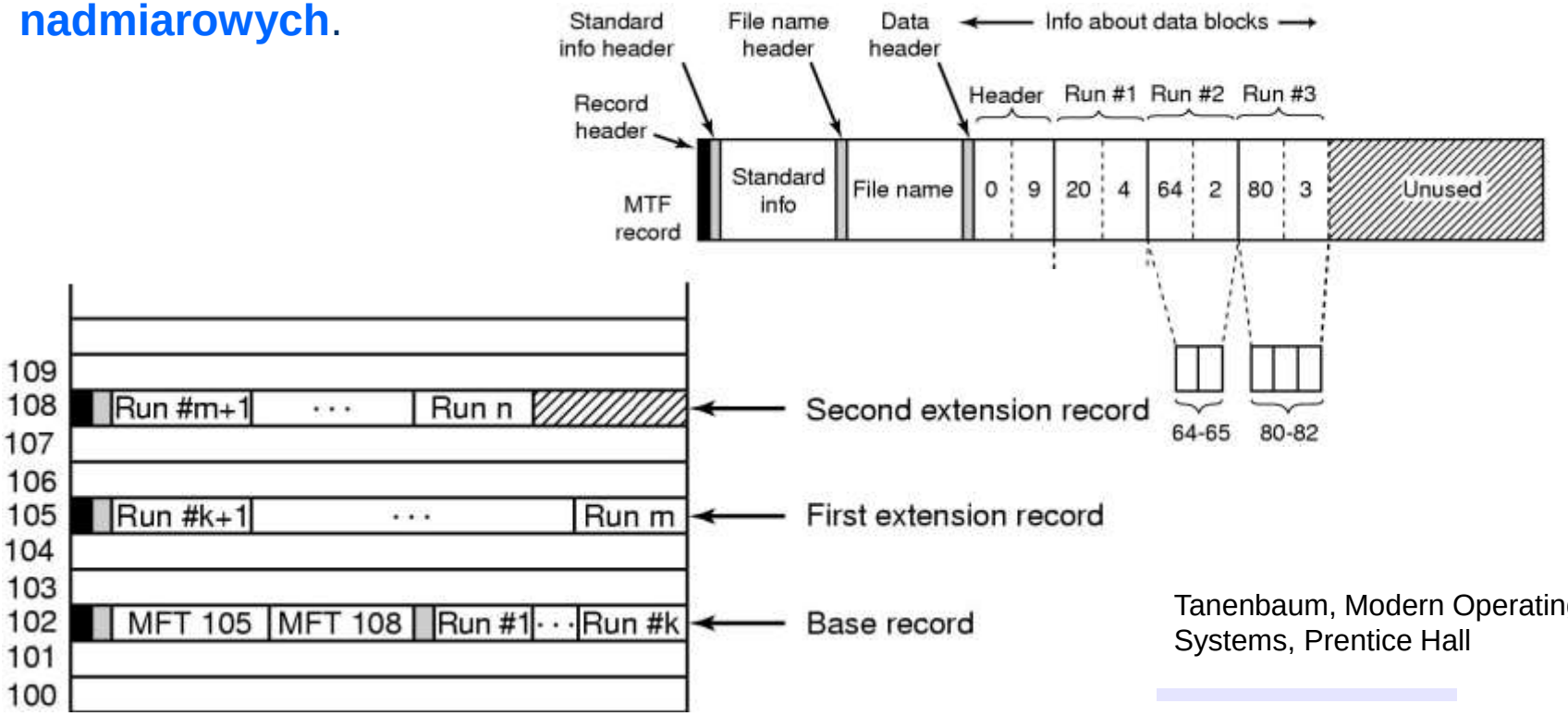
System plików NTFS

- Plik NTFS nie jest prostym strumieniem bajtów (jak dla systemów MS-DOS czy UNIX), jest to obiekt strukturalny złożony z atrybutów. Strumień bez nazwy – odpowiada tradycyjnej treści pliku. Mogą być strumienie alternatywne (ADS).

- Każdy plik woluminu NTFS ma unikalny ID nazywany **odsyłaczem do pliku** (*file reference)*. 64-bitowy identyfikator składa się z 48-bitowego numeru pliku (związany z numerem rekordu w MFT) oraz 16-bitowego **numeru kolejnego** (*sequence number*) – zwiększanego przy każdym powtórnym wykorzystaniu wpisu MFT. Identyfikator może być używany do wewnętrznych testów spójności.

System plików NTFS

- Każdy plik NTFS jest opisany przez jeden lub więcej wpisów (**base** and **extension records**) w **głównej tablicy plików** (*Master File Table*, MFT).
- Małe atrybuty (**rezydentne**) przechowuje się w rekordzie MFT, wielkie (jak duży atrybut danych) są przechowywane w jednym lub większej liczbie ciągłych **rozszerzeń** na dysku (wskaźnik rozszerzenia w MFT). Jeśli wskaźniki nie mieszczą się w podstawowym rekordzie pliku – pozostałe są zapamiętane w **rekordach nadmiarowych**.



Tanenbaum, Modern Operating Systems, Prentice Hall

System plików NTFS - katalogi

- NTFS organizuje katalogi hierarchicznie; korzeń indeksu katalogu zawiera główny poziom B+ drzewa (możliwe rozszerzenia dla dużego katalogu).

- Wpis katalogu:
  - nazwa,
  - odsyłacz,
  - rozmiar i czasy uaktualnień (kopie z rekordów MFT dla przyspieszenia operacji typu *dir*).

System plików NTFS – linki itp..

- **Punkt ponownego rozbioru** (analizy) systemu plików (*reparse point*), to specjalny typ pliku, który generuje błąd przy dostępie. Dane związane z tym punktem (*reparse tag and data*) określają akcję podejmowaną przez zarządcę wejścia/wyjścia (I/O manager). Zastosowania:
  - realizacja operacji montażu (link woluminu), znanej z systemu UNIX.
  - organizacja dostępu do plików składowanych poza systemem (*offline storage*).
  - realizacja operacji **directory junction**, montażu katalogu (link symboliczny katalogu).
  - **Linki symboliczne** (*symbolic links*) – pliki zawierające ścieżkę dostępu do innego pliku
- **Linki twarde** (*hard links*) – pozycje katalogowe współdzielące rekord MFT (i odsyłacz pliku).

System plików NTFS – funkcje zaawansowane

- NTFS dzieli dane na **bloki kompresji** (*compression units*) składające się z 16 przyległych klastrów.
- Dla **rzadkich plików** NTFS
  - Klastry zawierające same zera nie są przechowywane na dysku, a MFT zawiera dziury (*gaps*) w numeracji wirtualnych klastrów dla pliku.
  - Przy odczycie pliku dziura w numeracji wirtualnych klastrów zamieniana jest na ciąg zer w buforze funkcji czytającej.
- Systemy Windows umożliwiają doprowadzenie tomu NTFS do spójnego stanu oraz utworzenie **kopii-cienia** (volume *shadow-copy*), której można używać jako kopii zapasowej spójnego obrazu tomu. Wykonywanie takiej kopii jest odmianą **kopiowania przy zapisie** (*copy-on-write*); oryginalna treść bloków zmienionych po utworzenia kopii-cienia jest nadal dostępna w kopii.
- NTFS przechowuje **dziennik zmian** (Update Sequence Number Journal) wykonywanych w systemie plików. Oprogramowanie użytkowe może otrzymywać powiadomienia o zmianach i samodzielnie badać zmiany – bez potrzeby skanowania systemu plików.
- **Alternate data stream (ADS)** - plik może mieć dodatkowe forki (strumienie danych), nazywane w konwencji"filename:streamname"

System plików NTFS - rekonstrukcja

- Wszystkie modyfikacje dotyczące struktury systemu plików (metadane) są wykonane jako transakcje (z udziałem dziennika).
  - Przed modyfikacją struktury na dysku w dzienniku zapisywany jest rekord pozwalający na ponowienie i cofnięcie operacji zmiany.
  - Po dokonaniu zmiany na dysku w dzienniku jest zapisywany rekord zatwierdzenia transakcji.
  - Po awarii struktury systemu plików są przywracane do spójnego stanu przez przetwarzanie wpisów (rekordów) dziennika.
- Powyższe postępowanie nie gwarantuje możliwości rekonstrukcji plików użytkownika, a jedynie to że struktura systemu plików zostanie przywrócona do pewnego spójnego stanu sprzed awarii.
- Obsługą dziennika zajmuje się usługa systemowa: ***log file service***.

System plików NTFS - bezpieczeństwo

- Model bezpieczeństwa NTFS jest pochodną obiektowego modelu MS Windows (NT).
- Po zalogowaniu użytkownika system tworzy żeton dostępu (*access token*), zawierający m.in. SID *(security identifier)* tego użytkownika i SIDy wszystkich grup, do których należy; także listy przywilejów (użytkownika i grup). Procesy działające na rzecz użytkownika mają kopię żetonu; pozwala to na kontrolę dostępu do obiektów w systemie, w tym do obiektów systemu plików.

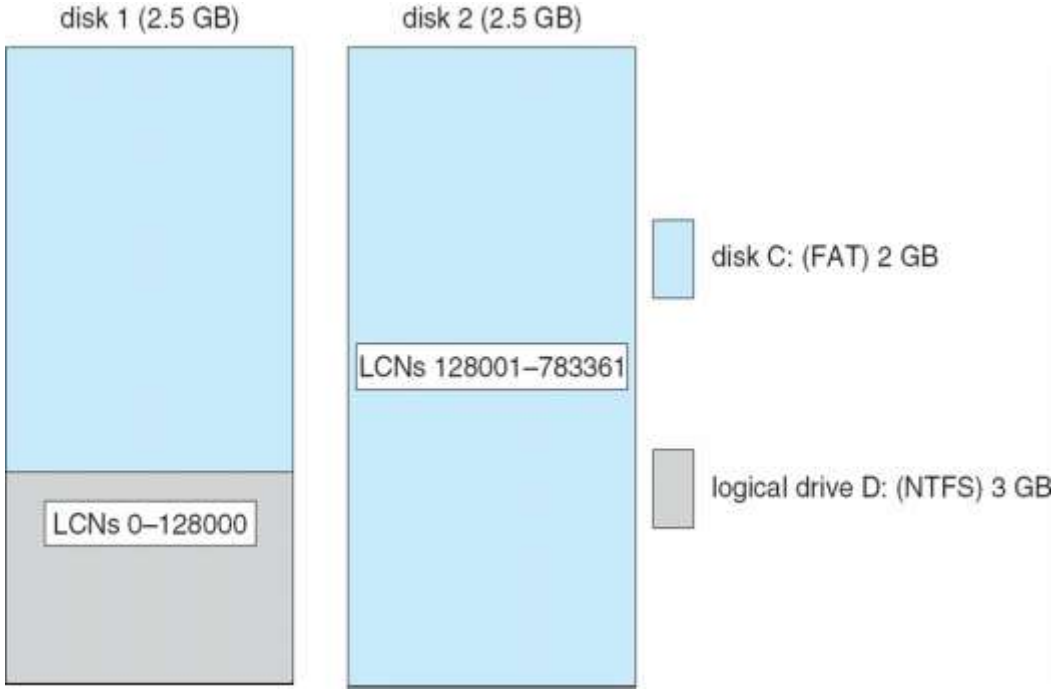
- Z każdym obiektem plikowym jest związany **deskryptor bezpieczeństwa** przechowywany w MFT.
- Deskryptor bezpieczeństwa zawiera identyfikatory zabezpieczeń (SID) właściciela obiektu i głównej grupy oraz dwie listy kontroli dostępu (*access control list, ACL*)
  - **uznaniowa** (*discretionary, DACL*), określa jakie interakcje (reading, writing, executing or deleting) są dozwolone bądź zabronione użytkownikom lub grupie.
  - **systemowa** (*SACL*), określa które interakcje z plikiem czy katalogiem mają być zapisane w dzienniku zdarzeń i kiedy (w przypadku pomyślnego czy niepomyślnego wyniku próby interakcji).

- Dostęp jest przyznany, gdy nie ma **wpisu zabraniającego**, a jest (przynajmniej jeden) **wpis zezwalający**.
- Istnieje możliwość dziedziczenia uprawnień (od katalogów nadrzędnych)

Zarządzenie woluminami

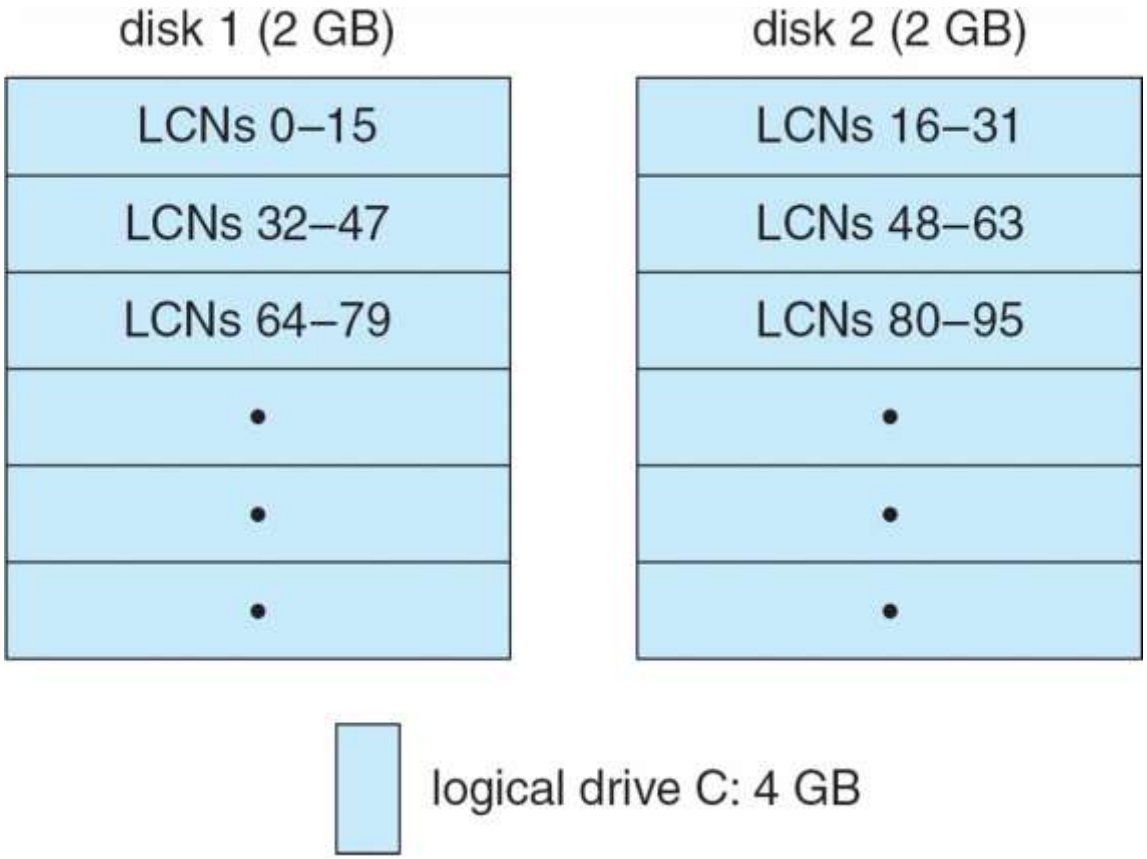
- **`FtDisk`** to odporny na błędy (fault tolerant) drajwer dysków, wykorzystujący nadmiarowe sektory (*sector sparing*). Dodatkowo NTFS pozwala na zmianę odwzorowania klastrów (*cluster remapping)*
- NTFS udostępnia kilka sposobów wykorzystania wielu dysków do realizacji jednego dysku logicznego.
- Kilka dysków (partycji fizycznych) może tworzyć **zestaw** (*volume set*)

Przykład wykorzystania dwóch dysków do konstrukcji dwóch woluminów



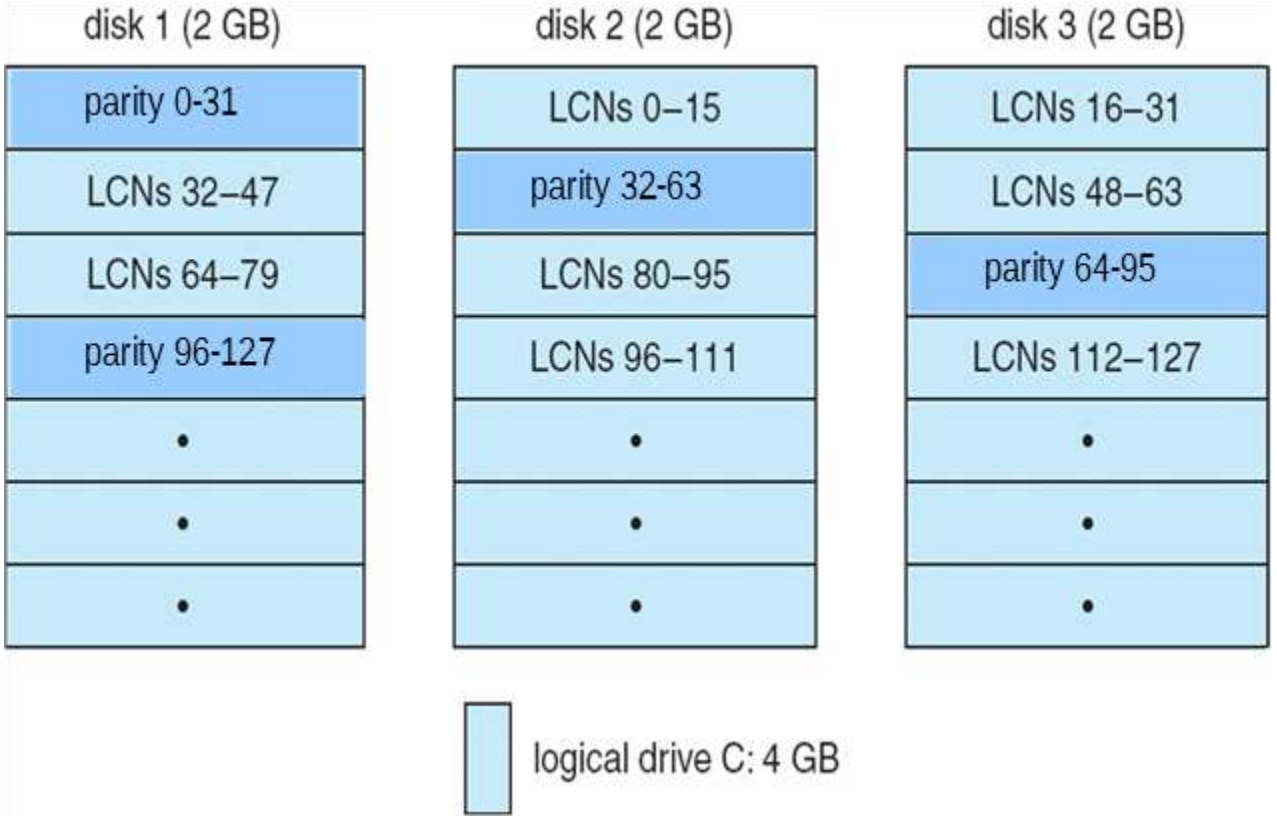
Dwa dyski pracujące z przeplotem (RAID 0)

- Przykład wykorzystania dwóch dysków do zapisu z przeplotem (RAID level 0, *disk striping*). Zaleta – możliwe przyspieszenie dostępu do danych (zwłaszcza dostępu sekwencyjnego)



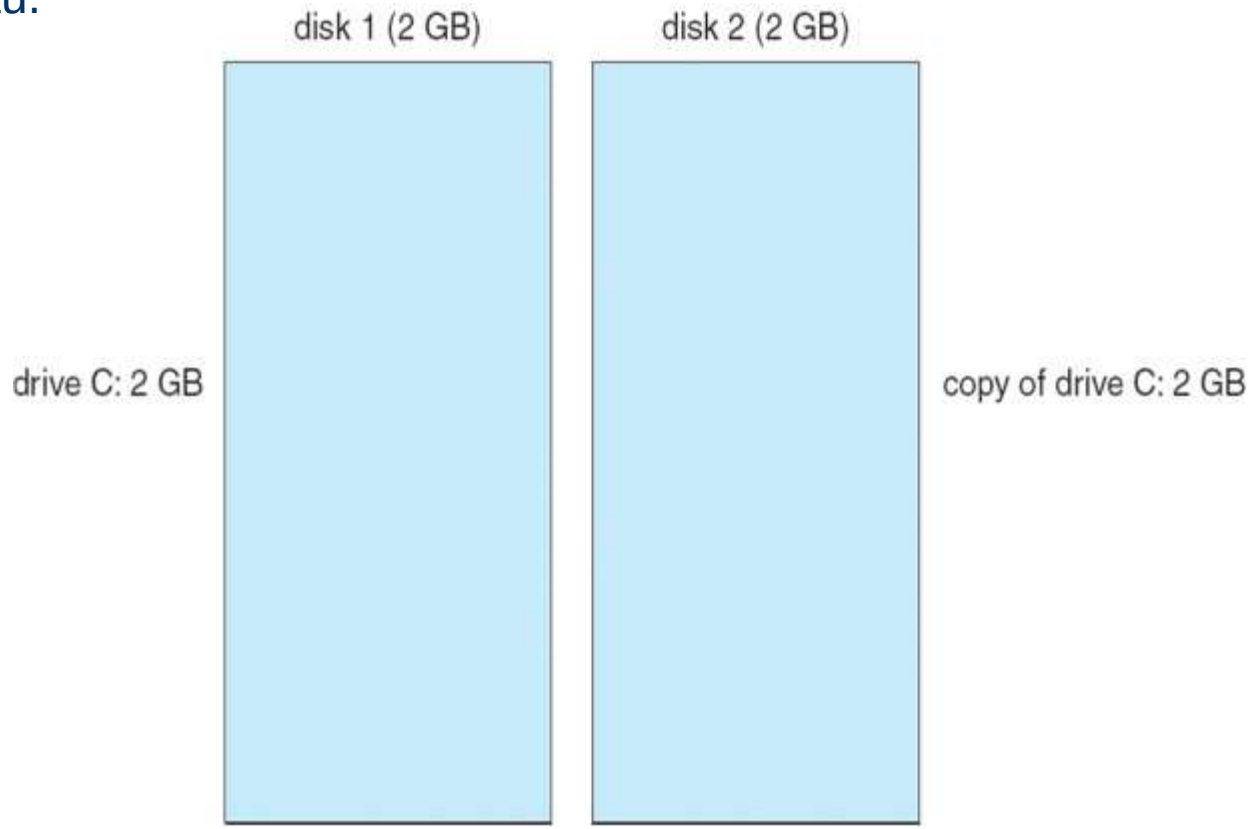
Zapis na trzech dyskach z kontrolą parzystości (RAID 5)

- Rozszerzenie zapisu z przeplotem (RAID level 0, *disk striping*) o kontrolę parzystości (dodatkowy dysk dla utrzymania pojemności. Zaleta – odporność na awarię jednego dysku. Wada: marna przepustowość dla strumienia wielu małych plików (cykl: odczyt-modyfikacja-zapis).



Dwa dyski lustrzane (RAID 1)

Dyski lustrzane (*disk mirroring*, RAID level 1) wykorzystują dwie identyczne partycje do przechowywania identycznej zawartości, tworzonej równolegle. Zalety: odporność na awarię jednego dysku, możliwa poprawa przepustowości odczytu.



ZFS

- **ZFS** jest połączeniem systemu plików i zarządcy woluminów logicznych zaprojektowanym przez Sun Microsystems. Obecnie rozwijany jako OpenZFS.
- **ZFS** jest dostępny w SO Linux (man zfs-fuse(8)). Cechy szczególne
  - Wsparcie bardzo dużych woluminów; do 256 zebibajtów (2^{78}B),ok. 1,50 10^{23}B
  - Zabezpieczenia przed uszkodzeniami danych. Ciągłe sprawdzanie spójności danych (sumy kontrolne), automatyczna naprawa przy wykryciu błędu – nie tylko wskutek awarii całego dysku, ale łagodnej degradacji danych, błędu firmware czy drajwera dysku
  - Integracja funkcji systemu plików i zarządcy woluminów
  - W czasie zapisu blok danych może być kompresowany, szyfrowany, opatrzony skrótem, a później poddany **deduplikacji**
  - Wsparcie migawek oraz kopii w trybie COW (*copy-on-write clones*)
  - Redundancja: RAID-Z1, RAID-Z2, RAID-Z3 (1,2 lub 3 dyski mogą się popsuć, rozwiązania podobne do RAID-5, RAID-6, RAID-7, ale szybsze – dzięki dynamicznym rozmiarom pasków (*stripes*).

Zarządzanie dyskiem

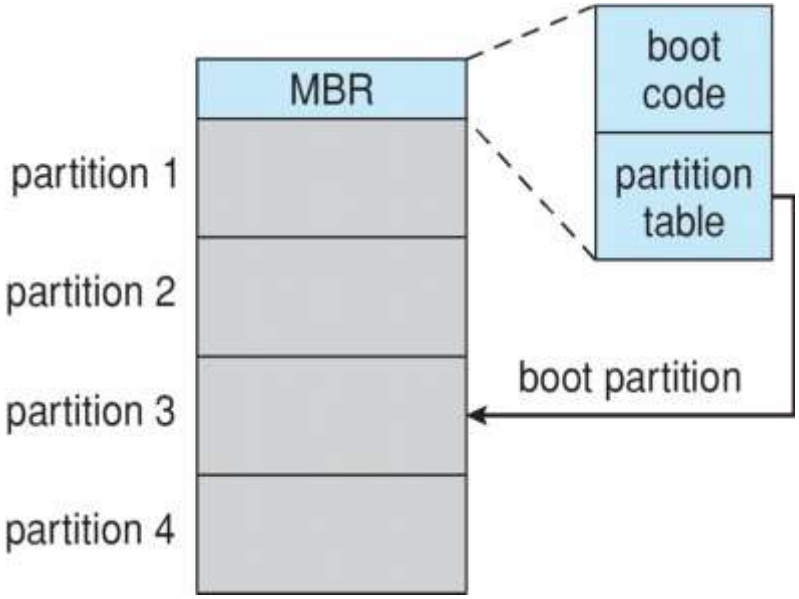
- **Formatowanie niskopoziomowe (fizyczne)** — podział dysku na sektory, które kontroler może odczytywać/zapisywać w jednym kroku.
  - Sektor zawiera nagłówek, obszar danych (na ogół o długości 512B) i zakończenie. Nagłówek i zakończenie zawierają typowo numer sektora i kod korygujący (error correction code, **ECC**)
  - Blok danych zazwyczaj zawiera 512 B danych
  - Formatowanie niskopoziomowe jest zazwyczaj częścią procesu wytwarzania dysku.
- Aby dysk mógł być użyty do przechowywania plików, system operacyjny musi na nim zapisać własne struktury danych.
  - **Partycjonowanie** dysku, to podział dyski ma części logiczne (np. odpowiadające grupom cylindrów).
  - **Formatowanie logiczne** – czyli nanoszenie początkowych danych systemu pliku związanego z partycją, w tym (pusty) główny katalog.
- Komputer, który rozpoczyna pracę wykonuje **program rozruchowy (bootstrap)**, który inicjuje sprzęt po czym uruchamia system operacyjny. Program rozruchowy bywa podzielony na części:
  - Ładowacz programu rozruchowego (*bootstrap loader*), zawarty w ROM
  - Właściwy program rozruchowy (zawarty np. w blokach rozruchowych dysku)

Rozruch PC – MBR/BIOS

Po włączeniu PC napięcia zasilania ustalają wartości. Wówczas:

- Oprogramowanie BIOS zawarte w pamięci ROM rozpoczyna test sprzętu (Power On Self Test ,POST). BIOS znajduje urządzenie rozruchowe (*boot device*)

- BIOS wczytuje pierwszy sektor urządzenia rozruchowego (tzw. MBR) do pamięci pod adresy 7C00 do 7DFF.
- BIOS powoduje, że procesor skacze do adresu 7C00, co rozpoczyna wykonanie kodu MBR.
- Kod MBR (niezależny od systemu operacyjnego) znajduje w tablicy MBR partycję rozruchową, po czym powoduje wykonanie programu ładującego zawartego w blokach rozruchowych (*boot blocks*) tej partycji.
- Program ładujący powoduje wczytanie systemu operacyjnego (zwykle jądra) do pamięci i rozpoczęcie jego wywoływania.



Uwaga:

- BIOS jest programem 16-bitowym z przestrzenią adresową o rozmiarze 1 MB.

GPT (GUID Partition Table)

Ograniczenia rozruchu przy pomocy **MBR**:

- Ograniczenie pojemności dysku systemowego 2 TB.
- Wsparcie do czterech głównych partycji; jedna z nich może być jednak użyta jako „kontener partycji logicznych”

Własności rozruchu przy pomocy **GPT**

- Każda partycja ma unikalny identyfikator (*globally unique identifier*, GUID)
- Tworzone są kopie GPT – do wykorzystania przy wydobywaniu się z awarii
- BIOS jest zastąpiony przez oprogramowanie **UEFI** (Unified Extensible Firmware Interface) o nieporównywalnie większej funkcjonalności. Kod UEFI może się wykonywać w trybie 32-bit owym bądź 64-bitowym. Zapewnia usługi rozruchowe oraz w czasie pracy systemu operacyjnego (*runtime services: date, time, NVRAM access,*…).
- UEFI wspiera dyski GPT, ale i MBR.
- W czasie rozruchu UEFI wczytuje z **partycji ESP** (EFI system partition) pliki, które ładują wybrany system. Partycja EFI zawiera system plików typu FAT (jego specyfikacja jest częścią specyfikacji UEFI).
- Możliwy jest bezpieczny rozruch (Secure Boot), zabezpieczający przed użyciem programów ładujących i sterowników, które nie mają odpowiedniej certyfikacji (*digital signature*).