wydajność

- metody przydziału - różnice w zapotrzebowaniu na pamięć i czas dostępu do bloków danych
- przydział ciągły - pobranie danych wymaga 1 kontaktu z dyskiem (dostęp sekwencyjny i swobodny)
- przydział listowy - (dostęp do *i*-tego bloku *i* operacji czytania z dysku -- dostęp sekwencyjny)
- struktura pliku - zależna od deklarowanego typu dostępu
- konwersja typu pliku - kopiowanie do nowego pliku o wymaganym typie

Zarządzanie wolną przestrzenią

Lista wolnych obszarów *(free-space list)*

- wektor binarny
- lista powiązana
- grupowanie
- zliczanie

Wektor bitowy

Mapa bitowa: 1 blok = 1 bit (0-zajęty 1-wolny)

nr bloku=liczba_bitów_w_słowie x liczba_wyzerowanych_słów + pozycja_pierwszego_ bitu”1”

- mało wydajne
- tylko dla małych dysków

Dysk – 160 GB blok=1024 B- mapa bitowa – 20 MB

4-blokowe grona – 5 MB

1TB-2^{40} B

Lista powiązana

Wskaźnik do 1-go wolnego bloku - w specjalnym m-cu na dysku oraz w pamięci

- metoda niewydajna - aby przejrzeć listę - odczyt każdego bloku (zazwyczaj szukany 1-szy wolny blok)

Grupowanie

w 1-szym wolnym bloku - adresy *n* wolnych bloków;

ostatni z nich zawiera adresy następnych *n* wolnych bloków

- umożliwia szybkie odnajdywanie większej liczby wolnych bloków

Zliczanie

pozycja wykazu wolnych obszarów: adres dyskowy 1-go wolnego bloku + licznik kolejnych wolnych bloków

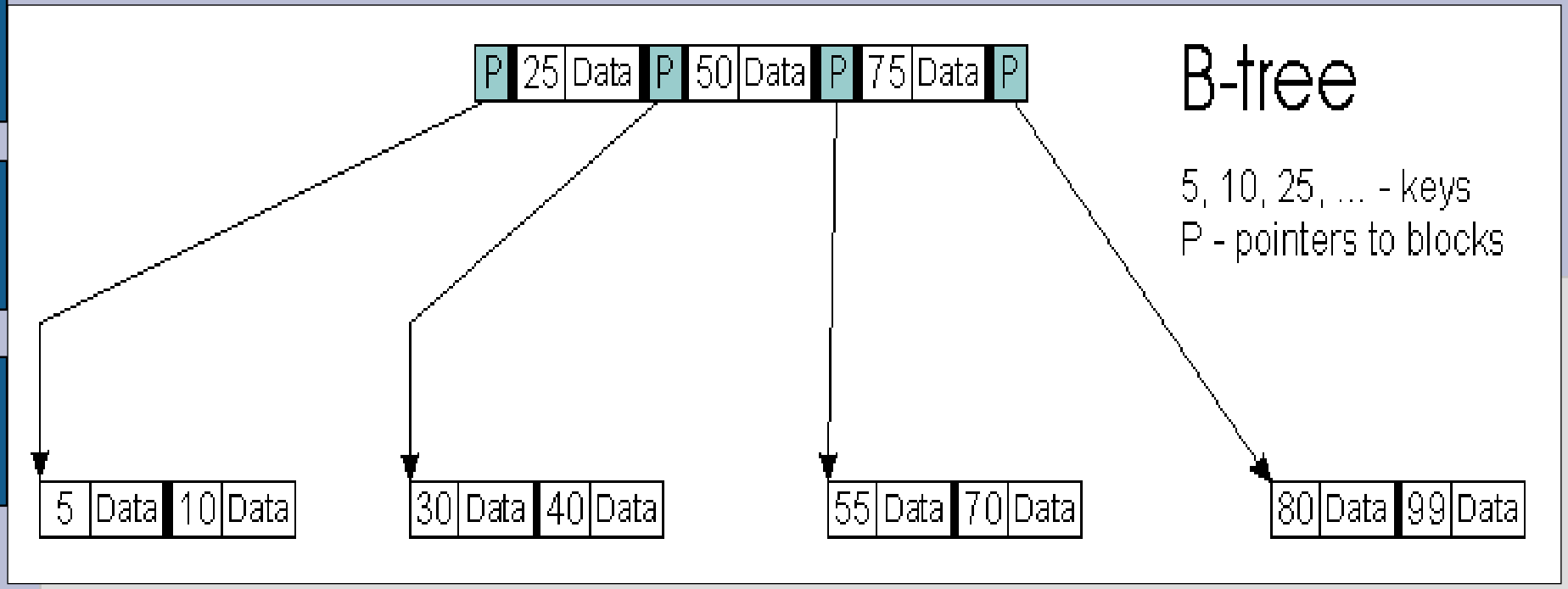
Implementacja katalogu

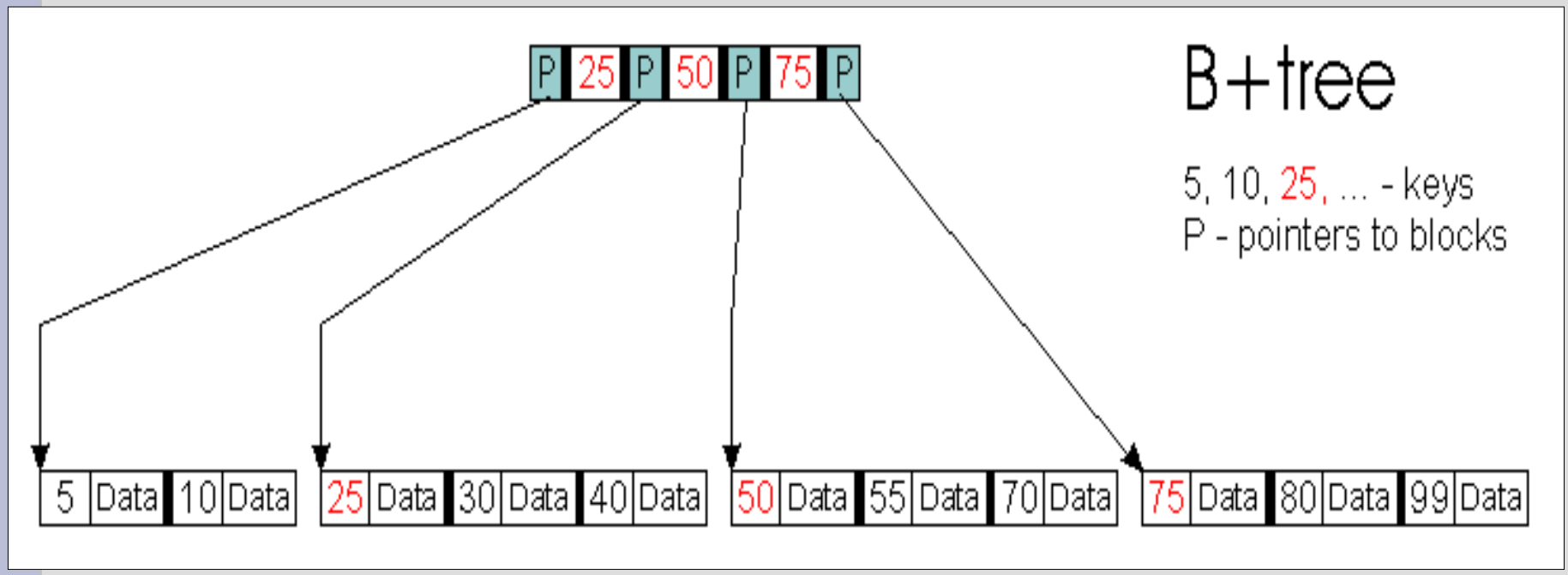
- ***Lista liniowa*** nazw plików ze wskaźnikami do bloków danych

  wada - liniowe przeszukiwanie

  (lista uporządkowana, B-drzewo)

- ***Tablica haszowania*** - funkcja haszowania odwzorowuje nazwę pliku na wskaźnik na liście liniowej


P 25 Data P 50 Data P 75 Data P
5 Data 10 Data
30 Data 40 Data
55 Data 70 Data
80 Data 99 Data
B-tree
5, 10, 25, ... - keys
P - pointers to blocks


P 25 P 50 P 75 P
5 Data 10 Data
25 Data 30 Data 40 Data
50 Data 55 Data 70 Data
75 Data 80 Data 99 Data
B+tree
5, 10, 25, ... - keys
P - pointers to blocks

Efektywność systemu plików

Algorytmy przydziału miejsca i obsługi katalogów

- Wstępny przydział i-węzłów – rozrzucenie ich w strefie dysku (bloki danych blisko i –węzłów)
- Łączenie bloków w grona; zmienne rozmiary gron
- Rodzaje informacji o plikach (daty dostępu; odczyt, zapis…)
- Rozmiar wskaźników w dostępie do danych a rozmiar pliku:

  8b - 256B, 16b - 64kB, 32b - 4GB
- Struktury jądra przydzielane dynamicznie (rozmiar tablicy otwartych plików, tablicy procesów)

INTEGRALNOŚĆ SYSTEMU PLIKÓW

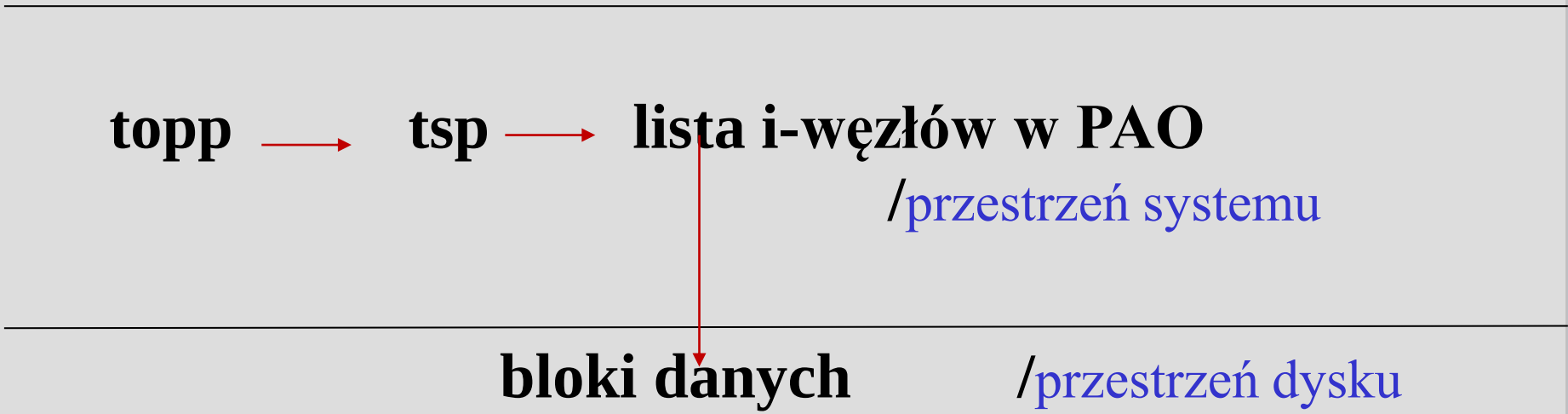
- sprawdzanie spójności (chkdsk, fsck; e2fsck, e4defrag)
- mechanizmy archiwizowania i odtwarzania danych archiwizowanych - awarie sprzętu lub błąd w oprogramowaniu.
- główne metody sporządzania kopii zapasowych plików:
  - okresowe składowanie zawartości pamięci
  - składowanie przyrostowe

UNIX

wywołanie systemowe:

czytaj(4,….) / przestrzeń użytkownika

deskrytptor pliku



topp – tablica otwartych plików procesu

tsp – tablica struktur plików

Lista i-węzłów w PAO <==> lista i-węzłów z dysku

Identyfikacja pliku przez jądro: (nr urządzenia log., nr i-węzła)

Nr urządzenia log. – określa system plików (własny superblok; w PAO; synchronizowany co 30 sec)

UNIX

VFS – *Virtual File System*

Te same funkcje systemowe – dostęp do każdego pliku na dowolnym systemie plikowym

Zaprojektowany na zasadach obiektowych:

- Definicje obiektów
- Oprogramowanie do działań na nich

4.2 BSD – FFS (*Fast File System*) – dwa rozmiary bloków: 8kB, fragment – n*1kB

wersja 7 Unix- katalog – wykaz: 14B – nazwa pliku + 2B – nr i-noda

4.3BSD – wpisy katalogowe zmiennej długości:

- długość
- nazwa
- nr i-węzła

- Pamięć podręczna nazw katalogów (pamiętane są i -węzły ostatnio używanych katalogów)

4.2 BSD – grupa cylindrów (1 lub więcej sąsiadujących cylindrów)

- inf. nagłówkowe (superblok, i-węzły, blok opisu cylindrów) - w różnych odległościach od początku grupy; na różnych płytach dysku
- i-węzeł pliku – w tej samej grupie cylindrów co i-węzeł katalogu (ls z opcjami odwołuje się do i –nod'ów)
- i-węzeł nowego katalogu – w innej grupie cylindrów (z dużą liczbą wolnych i-węzłów)
- bloki przydzielane plikom w obrębie tej samej grupy cylindrów (małe pliki – minimalny ruch głowic)

FFS – 30% technicznej przepustowości dysku

wersja 7 – 3%

Linux

- Minix (nazwy 14-znakowe; max. rozmiar plik 64MB)
- ext2 – FFS (Second Extended File System - 1993)
  - bloki w pliku katalogowym – powiązana lista wpisów
    - długość wpisu
    - nazwa pliku
    - nr i-węzła
  - nie używa bloków cząstkowych (fragmentów)
  - mniejsze bloki (1, 2, 4kB)
  - operowanie gronami (1 op. we/wy dotyczy kilku bloków; logicznie sąsiadujące bloki pliku – przylegające bloki dyskowe)
  - wiele grup bloków
  - obsługa „dziurawych” plików

Linux

- blokom danych przydzielana ta grupa bloków, do której należy i-węzeł pliku
- i-węzły plików zwykłych - w grupie katalogu macierzystego
- pliki katalogowe – rozproszone
- wewnątrz grup – przydziały ciągłe (minimalizacja fragmentacji)
- występuje mapa bitowa wolnych bloków w grupie – szukanie miejsca dla pliku:
  - tworzony nowy plik – od początku grupy bloków
  - rozszerzanie pliku – od bloku przydzielonego ostatnio

Linux

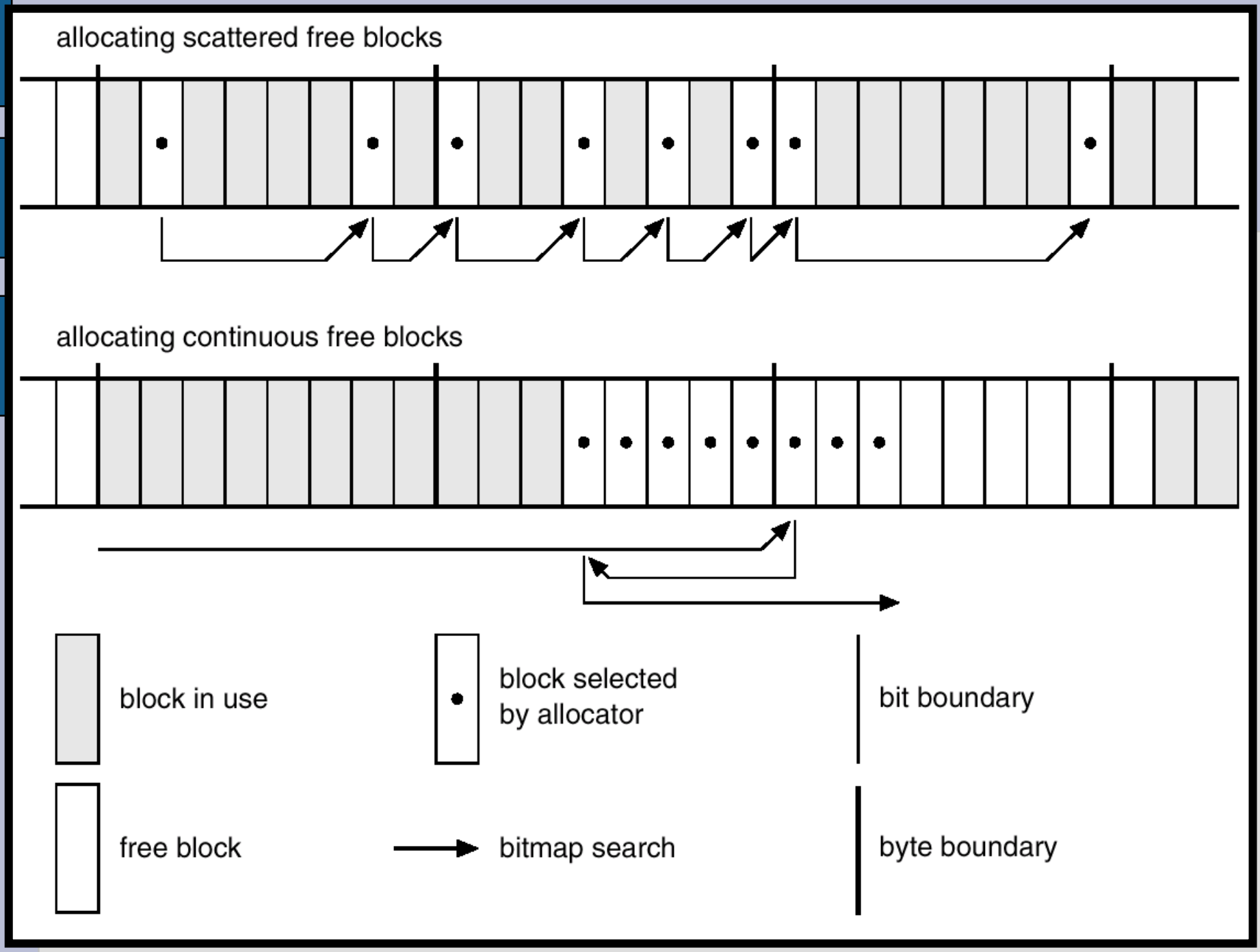
Szukanie 2-etapowe

- Całego wolnego bajta w mapie bitowej

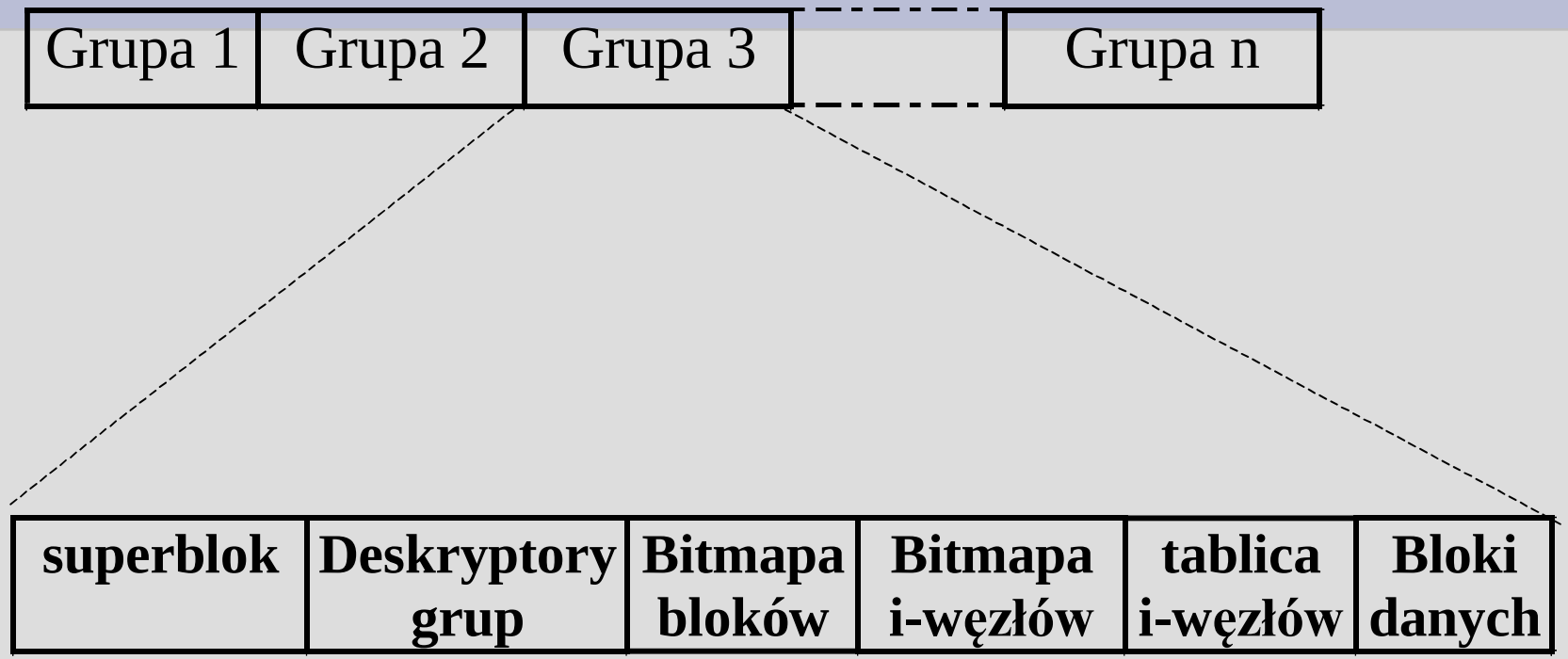
  Przydział miejsca porcjami 8-blokowymi; po znalezieniu bajta w bitmapie – przeszukiwanie wstecz dla uniknięcia dziur;

  wstępnie przydziela się 8 bloków; przy zamykaniu pliku odznacza się niezajęte bloki
- Pojedynczych wolnych bitów (jeśli 1 się nie powiedzie)

  blisko początku miejsca szukania


allocating scattered free blocks
allocating continuous free blocks
block in use
block selected by allocator
bit boundary
free block
bitmap search
byte boundary

System plików w Linuksie



superblok, deskryptory grup – w każdej grupie lub 0,1 $[3\ 5\ 7]^n$

Superblok (ext2_super_block)

- liczba i-węzłów na dysku
- liczba wolnych i-węzłów
- liczba bloków na dysku
- liczba wolnych bloków dyskowych
- liczba zarezerwowanych bloków dyskowych
- pierwszy blok z danymi
- rozmiar bloku
- rozmiar fragmentu
- liczba bloków, fragmentów i i-węzłów w grupie
- czas ostatniego zamontowania, zapisu na dysk, sprawdzenia
- maksymalna liczba zamontowań, liczba aktualnych zamontowań
- rozmiar struktury i-węzła
- pierwszy niezajęty węzeł
- domyślny identyfikator użytkownika dla bloków zarezerwowanych
- domyślny identyfikator grupy dla bloków zarezerwowanych

Deskryptory grup

- Tablica rekordów opisujących poszczególne grupy
- 1 rekord: liczba wolnych i-węzłów, liczba wolnych bloków
- Używane podczas przydzielania bloków

Grupy

- Każda grupa ma określoną wielkość (8MB - 128MB) za wyjątkiem ostatniej
- Mapa bitowa zajętości bloków ma wielkość jednego bloku
  - 1024*8*1024 = 8MB
  - $4096*8*4096=2^{12} * 2^{12} * 2^{3} = 2^{27} =$ 128MB
- Mapa bitowa zajętości i-węzłów - ma wielkość jednego bloku - dla każdego i-węzła jest przydzielony jeden bit

KATALOGI

- katalog w Linuksie jest także plikiem
- jego wewnętrzna reprezentacja danych jest uporządkowana
- każda pozycja w katalogu składa się z:
  - numeru i-węzła
  - długości pozycji katalogowej
  - długości nazwy
  - samej nazwy (do 255 znaków)….typ
- lista jednokierunkowa -> tablice haszujące

ext3

- 1999r – RedHat, Stephan Tweedie
- kompatybilny z ext2
- usprawnienia
  - mechanizm księgowania
  - indeksowane katalogi
- struktura ext3_dir_entry – pole file_type (8 bitów)

ext4

- 2008 r – od 2.6.19; stabilny od 2.6.28
- kompatybilny z ext2, ext3 (możliwe do zamontowania jako ext4 (1 bit w i-węźle)… ale
- Pliki – ciągłe porcje danych
- Używa ekstentów „*extens*” (zmiennego rozmiaru --- do 128MB każdy)
- W i-węźle max. 4 informacje o ekstentach
- Dla większych plików – adresowanie pośrednie
- Volumen – do 10^{18} B – eksa(i)bajt - 2^{60}B
- Plik – do 16TB =$16*2^{40}$B = 2^{44}B

ext4

extent: (12B) 3 wartości:

- Początkowy blok pliku w danym extencie (logiczny numer) ---- 4B
- Rozmiar mapowanego obszaru (w blokach) ------- 2B
- Początkowy blok danych na dysku ------ 2B+4B (id bloku na dysku -- 48bitów)

Dla rozmiaru bloku=4kB=2^{12} B:

rozmiar woluminu = $2^{48} * 2^{12}$ B= 2^{60} B=1EiB

rozmiar pliku = $2^{32} * 2^{12}$ B= 2^{44} B=16TiB

rozmiar extentu = $2^{15} * 2^{12}$ B= 2^{27} B=128MB (1 bit z 16 – używany w prelokacji)

Metadane:

ext2,3 --- 15 adresów po 4B=60B

ext4 ------ 4 extenty po 12B + (nagłówek=12B)= 60B

Rozmiar i-węzła:

<128B; rozmiar bloku>; domyślnie 256B

pierwsze 128B – taki sam układ

pozostałe:

- o ustalonym rozmiarze—nanosekundowe znaczniki czasu
- o zmiennym rozmiarze: sekcja EAS (*Fast Extended Attributes*)

m.in. ACL

ext4

Pliki o rozmiarze > 512B → dodatkowe poziomy indeksów (nagłówek zawiera inf. o głębokości drzewa)

- Zmniejszenie fragmentacji
- Prelokacja
- Wieloblokowa alokacja (mballock)

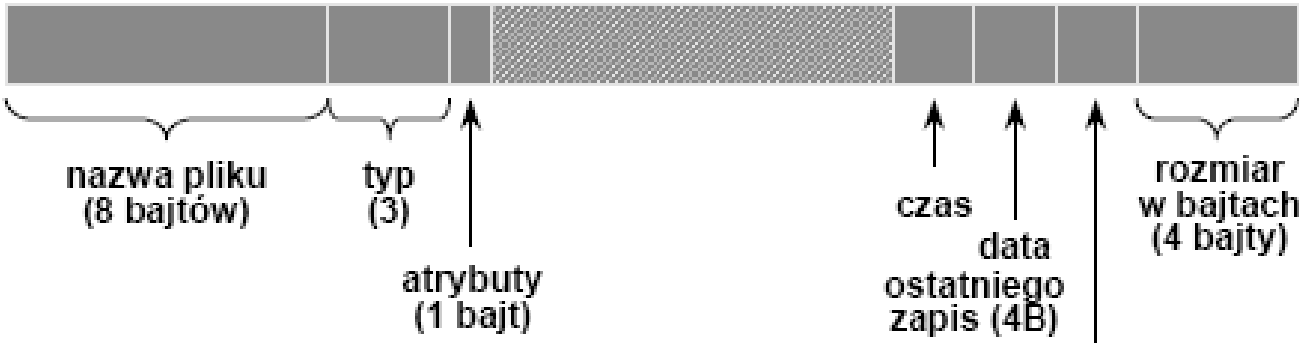
Indeksowane katalogi

- bloki 0..511 katalogu – struktura indeksująca
  - blok 0 – korzeń drzewa + nagłówek
  - blok indeksujący: 512 wpisów (klucz, adres)

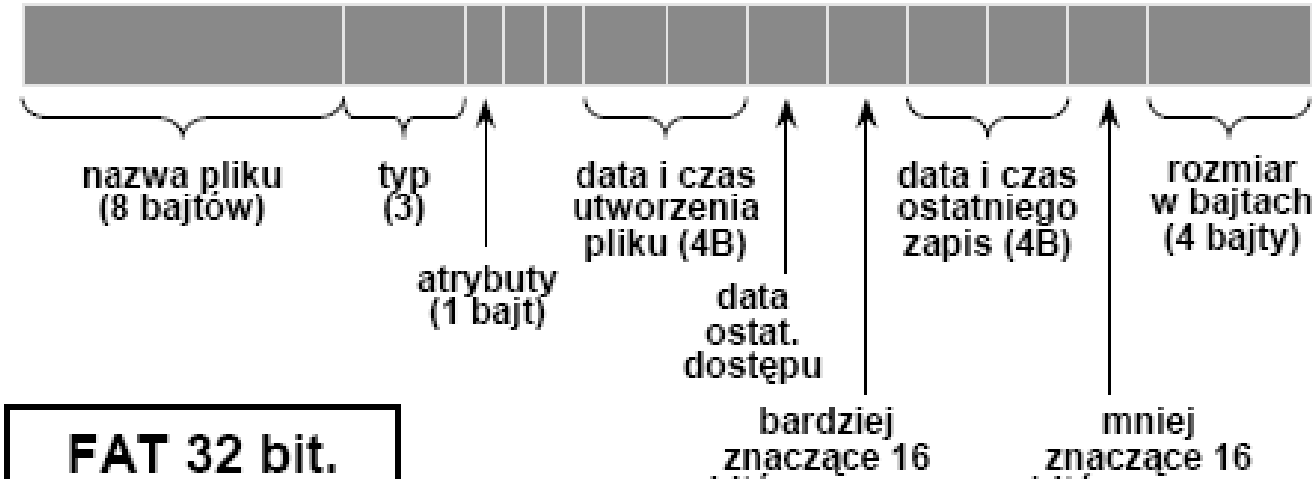
    klucz – wynik fcji haszującej + znak kolizji;

    adres – logiczny adres bloku danych lub kolejnej struktury indeksującej
- 90000 plików w katalogu ; kolejny poziom -> 50mln. wpisów
- Zwiększenie wydajności; koszty – 2MB – na strukturę
  -> indeksowanie – tylko dla dużej ilości wpisów w katalogu

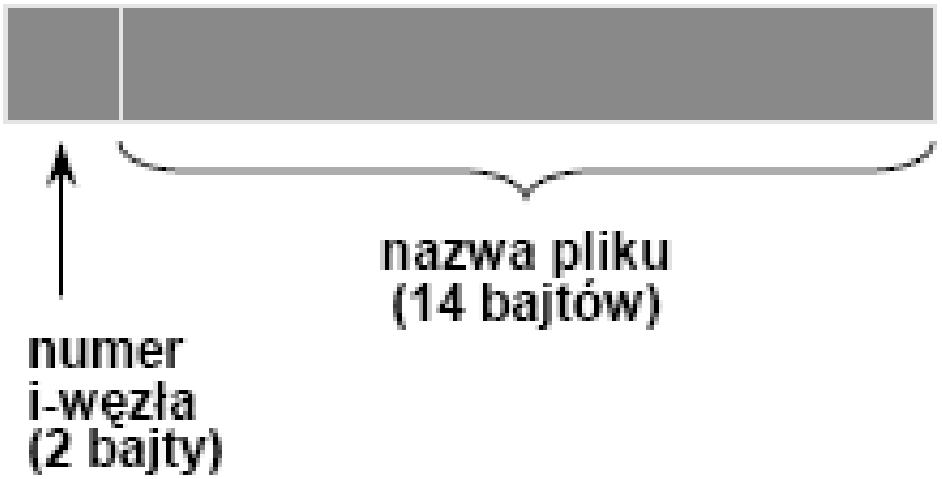

blok nadrzędny
tablica FAT
kopia tablicy FAT
katalog główny
bloki danych
blok nadrzędny
tablica i-węzłów
bloki danych
MFT
strefa MFT
bloki danych
kopia 16 pierwszych rekordów MFT
bloki danych



FAT 12/16 bit.



FAT 32 bit.


nazwa pliku
(14 bajtów)
numer
i-węzła
(2 bajty)

NTFS

- Tom (volume) – podst. jednostka
- Operowanie gronami (2^n przyległych sektorów)
- Adres dyskowy=LCN (logical cluster number)
- Plik – obiekt strukturalny, złożony z atrybutów
  - Każdy atrybut – niezależny strumień bajtów
  - Atrybuty standardowe dla wszystkich plików: nazwa; czas utworzenia; deskryptor bezpieczeństwa; liczba dowiązań; beznazwowy atrybut danych

MFT – Master File Table

- Każdy plik opisany min. 1 rekordem
- Rozmiar rekordu – parametr systemu (1-4kB)
- Małe atrybuty – rezydentne w MFT
- Wielkie atrybuty – przechowywane w rozszerzeniach na dysku; wskaźniki do nich w rekordzie MFT
- Pliki o wielu atrybutach – podstawowy rekord pliku (base file record) + wskaźniki do rekordów nadmiarowych

  - każdy plik ma 64-bitowy identyfikator (file reference);
    48 bitów – nr rek. w MFT
    16 bitów – nr kolejny (inkrementowany przy powtórnym użyciu wpisu w MFT)
- struktura katalogów – B^+ drzewo; każdy wpis – nazwa pliku, odsyłacz, znacznik czasu utworzenia, rozmiar (z MFT)

Pliki metadanych

- Tablica MFT
- MftMirr – kopia metadanych (pierwsze 16 pozycji)
- LogFile – plik dziennika transakcji
- Volume – inf. o wolumenie (nazwa, wersja NTFS)
- AttrDef – tablica definicji atrybutów
- . – katalog główny
- Bitmap – plik bitmapy klastrów (gron)
- Boot – plik inicjacyjny
- BadClus – lista złych klastrów (gron)
- Quota – ograniczenia
- Upcase – tablica konwersji małych liter na duże

ODPORNOŚĆ - TRANSAKCJE

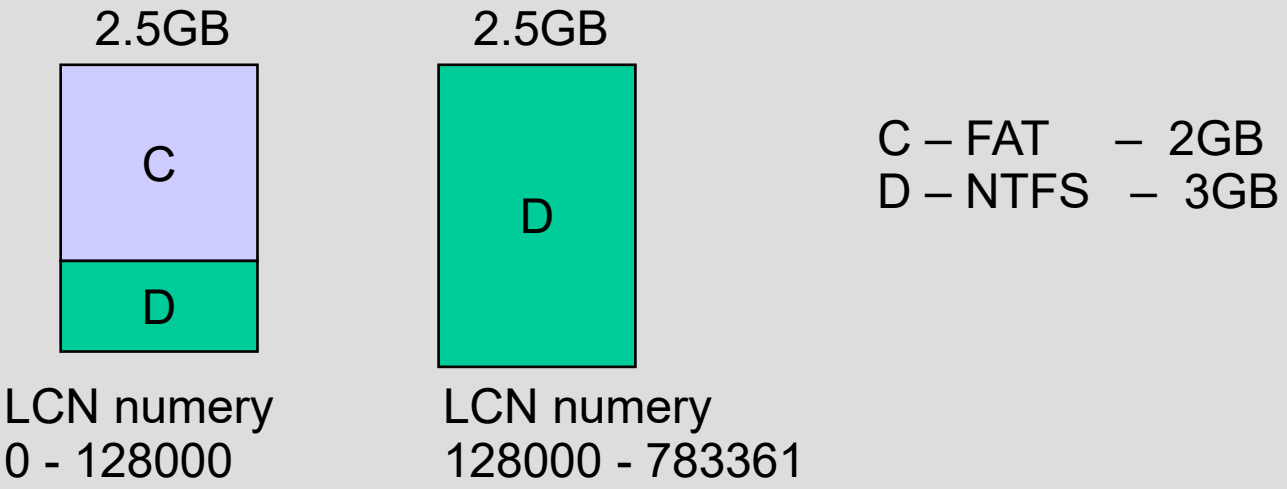
- Dla zapewnienia integralności systemu plików (struktury danych systemowych)
- Każda zmiana w systemie plików + inf. o pomyślności zakończenia – zapisywana w pliku logu (umożliwia powtórzenie lub anulowanie operacji)
- Po awarii przetwarzanie zapisów dziennika
- Okresowo – zapis do dziennika punktów kontrolnych – (*chekpoint*)

	plik	katalog
R	oglądanie zawartości	pokazywanie plików z katalogu
W	zmiana, usunięcie zawartości	dodawanie elementu do katalogu
X	uruchomienie pliku wykonywalnego	cd
D	usuwanie pliku	usuwanie katalogu (pustego)
P	zmiana praw dostepu	zmiana praw dostepu
O	otrzymanie własności	otrzymanie własności

Zarządzanie tomem (wolumenem) - 1

Łączenie wielu partycji – **tom logiczny** (do 32 stref fizycznych – partycji dysków, dysków)

Mechanizm LCN



Zarządzanie tomem (wolumenem) - 2

System plików z paskowaniem (stripping)

Schemat **RAID** poziomu **0** – paskowanie dysku

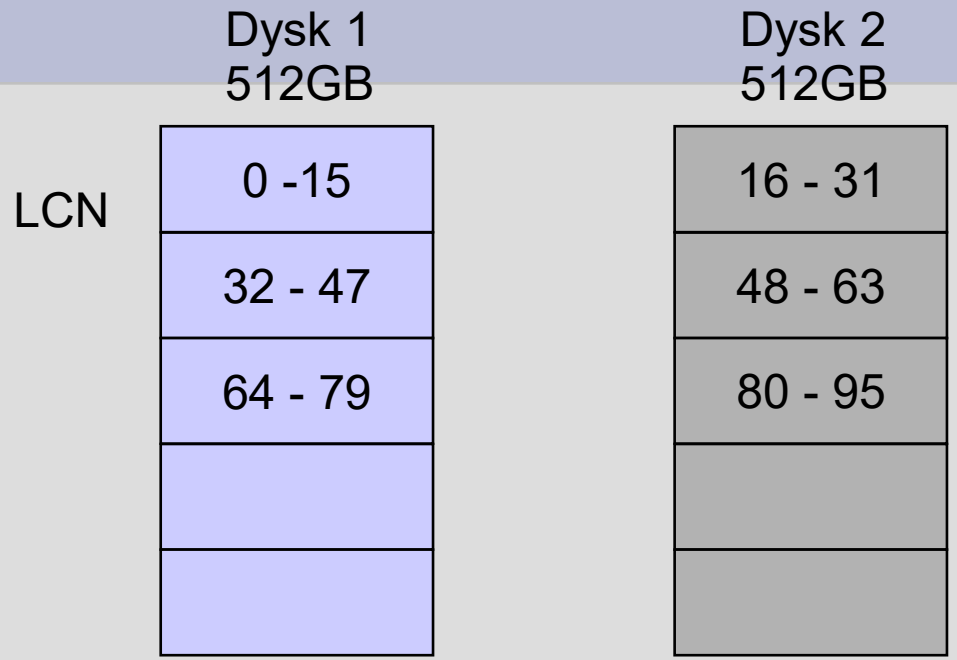
R*edundant* ***A****rray of* ***I****ndependent* ***D****isks*

Kolejne paski przydzielane kolejnym strefom fizycznym

Zbiór pasków – 1 tom logiczny

Równoległe operacje we/wy - polepszenie przepustowości we/wy (duże pliki !!!)

Napęd logiczny C: 1TB



poszczególne dyski – oddzielne kontrolery
partycje do strippingu – podobny rozmiar
dyski nie powinny być używane do innych celów
powszechnie 2 – 4 partycje (teoretycznie do 32)

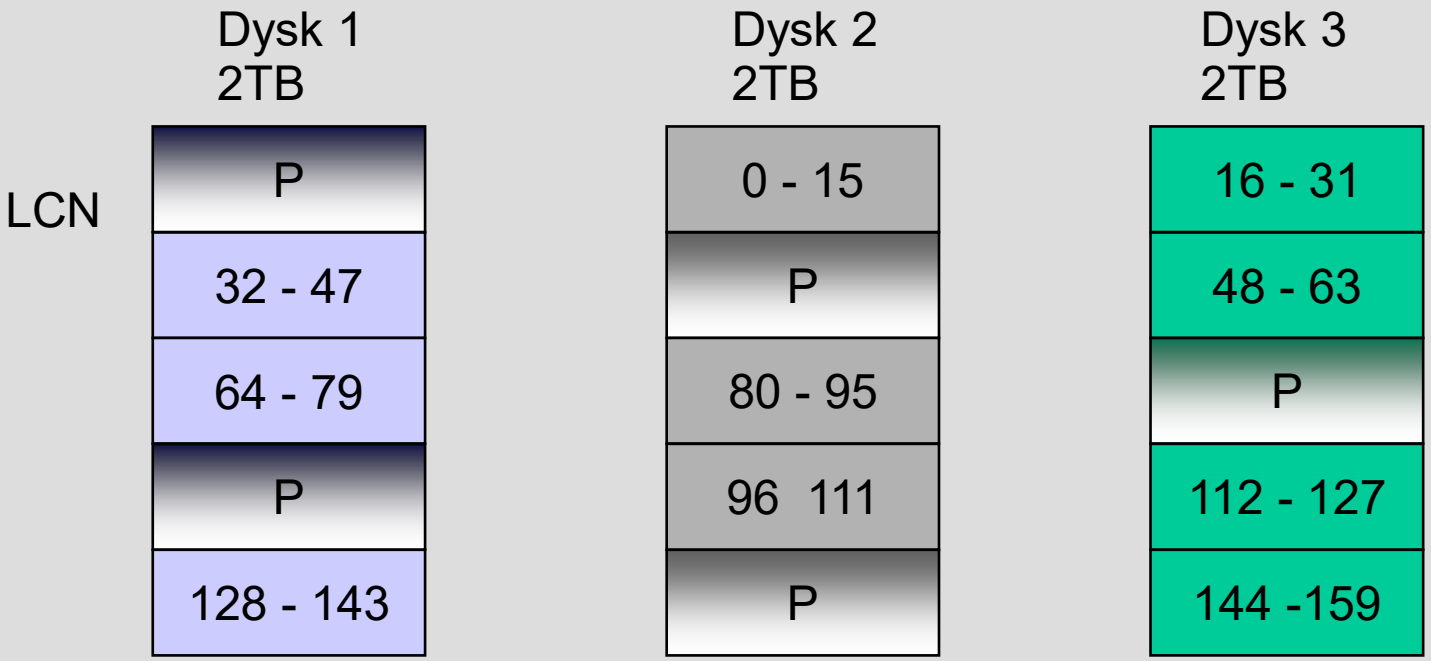
Zarządzanie tomem (wolumenem) - 3

Zbiór pasków z parzystością

RAID 5

- Odporność na uszkodzenia
- Kolejne porcje danych na kolejnych dyskach + dane o parzystości (XOR) krążą po dyskach zestawu
- Przy uszkodzeniu paska – możliwość zrekonstruowania danych
- Min. 3 jednakowe strefy na 3 dyskach

Napęd logiczny C: TB

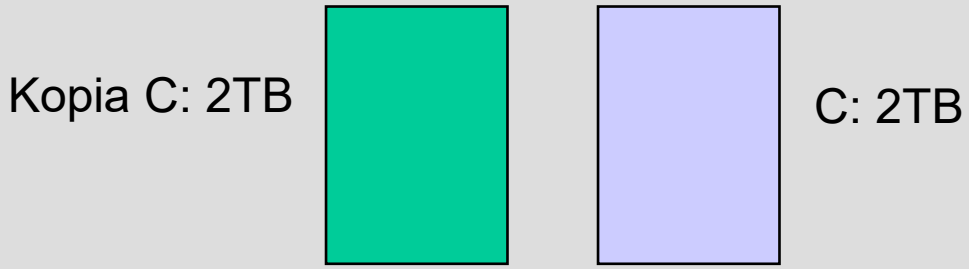


Zarządzanie tomem (wolumenem) - 4

Dyski lustrzane DISK MIRRORING

RAID 1

- **2 identyczne strefy na 2 dyskach**
- **Polepszenie bezpieczeństwa**
- **Przyspieszenie we/wy**
- **Oba dyski – osobne sterowniki (*duplex set*)**



- NT programowo implementuje RAID 0, 1, 5
- **Zapas sektorów (*sector sparing*)**

  Część sektorów nie jest ujęta w mapie dobrych sektorów - rezerwa użyta w razie awarii (wtórne odwzorowanie grona – *cluster remapping*)

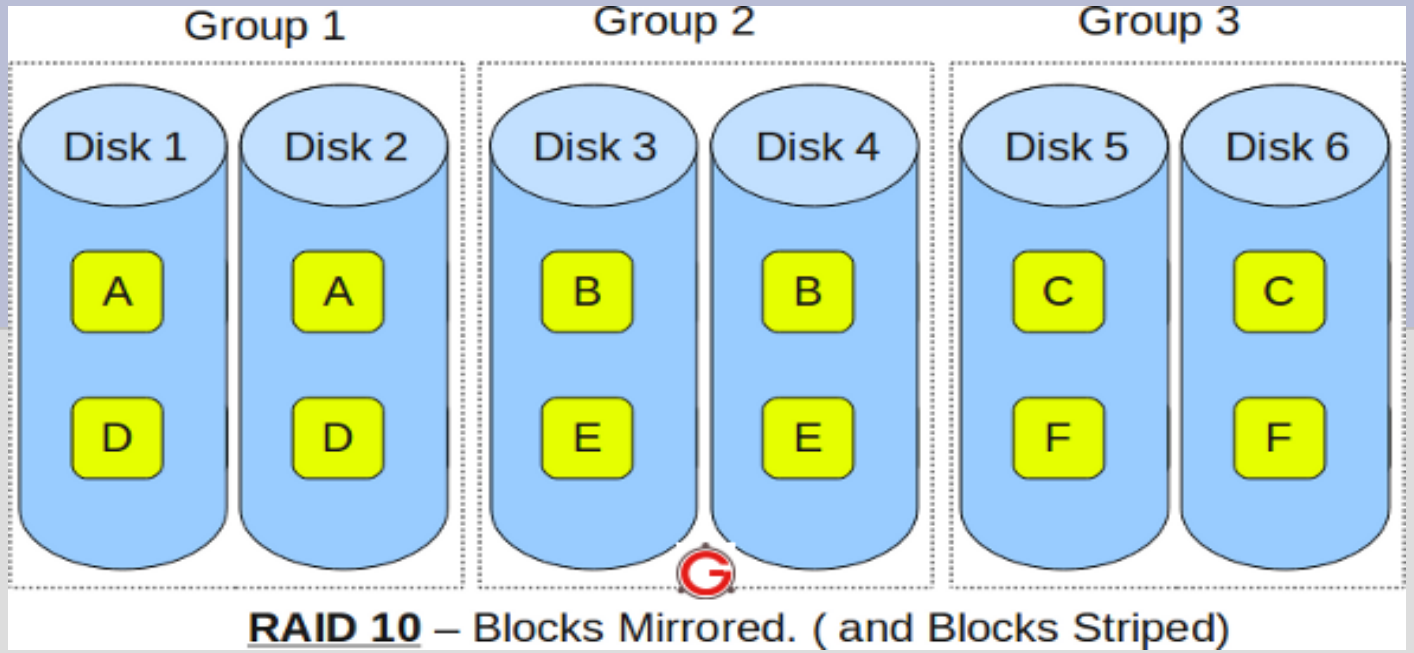
- **Upakowanie**

  automatyczna kompresja plików; NTFS dzieli plik do kompresji na jednostki upakowania złożone z 16 kolejnych gron

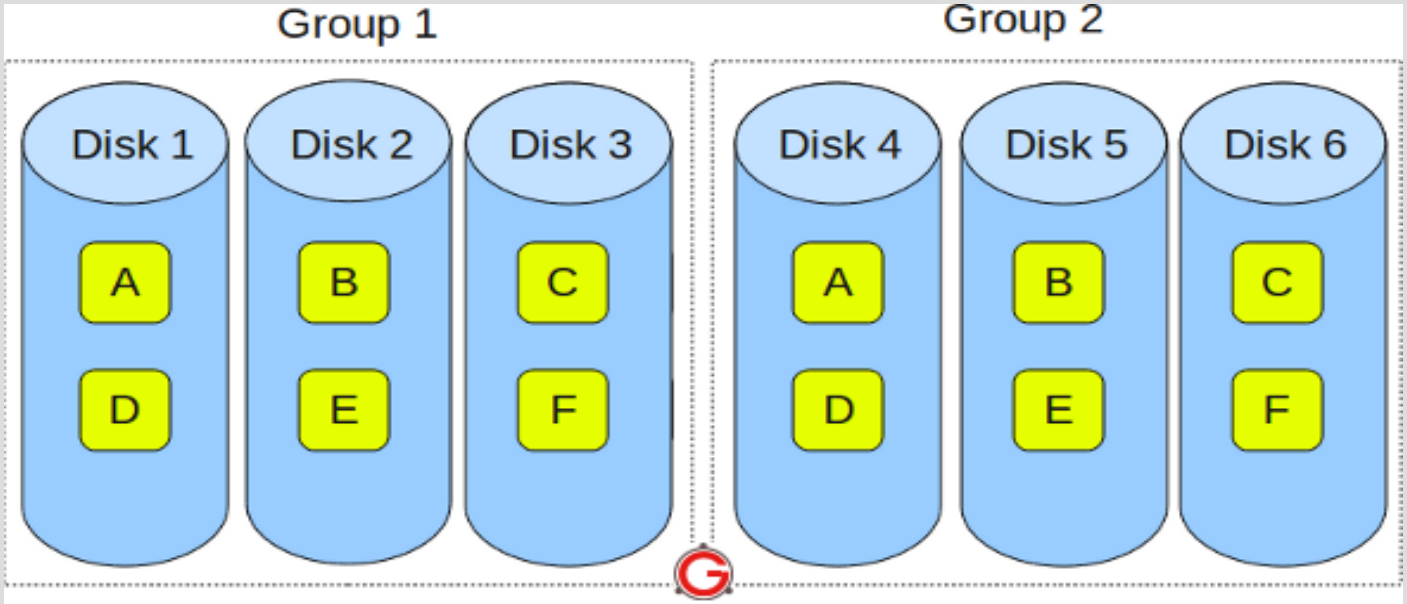
pliki rozrzedzone - grona zawierające same 0 ; system nie przydziela im miejsca na dysku (przerwy w nr gron wirtualnych); podczas czytania – uzupełnienie 0 w buforze

RAID

- RAID 0
- RAID 1
- RAID 3
- RAID 5
- RAID 6
- RAID 10 /paskowanie mirroringu
- RAID 01/mirroring z paskowania



RAID 10 – Blocks Mirrored. (and Blocks Striped)



RAID 01 – Blocks Striped. (and Blocks Mirrored)

http://www.thegeekstuff.com/2011/10/raid10-vs-

ReFS Resilient File System

Dla przetwarzania dużej liczby plików /serwery plików

$1YB = 2^{80}B = 10^{24}B$

- Samonaprawialność
- Kompatybilny z NTFS
- Nie dla dysków systemowych
- Przechowuje 64-bitowe sumy kontrolne dla danych i metadanych