

# System plików

1. Pojęcie pliku
2. Typy i struktury plików
3. Metody dostępu do plików
4. Katalogi
5. Budowa systemu plików
6. Przechowywanie podręczne
7. Integralność systemu plików
8. Semantyka spójności
9. Przykłady implementacji systemu plików

## Pojęcie pliku (ang. file)

- ☼ Plik jest abstrakcyjnym obrazem informacji gromadzonej i udostępnianej przez system komputerowy.
- ☼ Plik jest podstawową jednostką logiczną magazynowania informacji w systemie komputerowym, widoczną dla użytkownika.
- ☼ Plik jest nazwanym zbiorem powiązanych ze sobą informacji, zapisanym w pamięci pomocniczej (najczęściej nieulotnej, czyli trwałej).

## Zadania systemu operacyjnego

- ☼ Zadaniem systemu operacyjnego w odniesieniu do plików jest zapewnienie odwzorowania pomiędzy abstrakcyjnym obrazem pliku a jego reprezentacją na urządzeniu fizycznym.
- ☼ Wyszczególnienie zadań:
  - identyfikacja pliku (hierarchiczna struktura katalogów),
  - udostępnienie interfejsu operacji plikowych (API),
  - realizacja operacji dostępu do plików i katalogów z zapewnieniem bezpieczeństwa (synchronizacja i autoryzacja dostępu), spójności i efektywności.

## Atrybuty pliku

- ☼ Nazwa — ciąg znaków służących użytkownikowi do identyfikacji pliku
- ☼ Typ — informacja służąca do rozpoznania rodzaju zawartości pliku i tym samym sposobu interpretacji
- ☼ Lokalizacja — informacja służąca do odnalezienia pliku w systemie komputerowym (urządzenie i położenie pliku w tym urządzeniu)
- ☼ Rozmiar — bieżący rozmiar pliku w ustalonych jednostkach (bajtach, słowach, blokach itp.)
- ☼ Ochrona — informacje umożliwiające kontrolę dostępu
- ☼ Czasy dostępu — daty i czasy wykonywania pewnych operacji na pliku, typu odczyt, modyfikacja, utworzenie

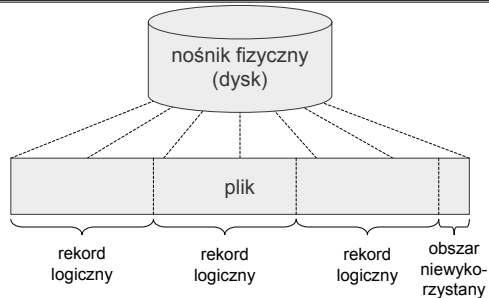
## Typy plików

- ☼ Typ pliku określa rodzaj informacji przechowywanej w pliku i tym samym sposób interpretacji jego zawartości, np. program binarny, wynik kompilacji, kod źródłowy, makrodefinicja (plik wsadowy, skrypt powłoki itp.), tekst, biblioteka programisty, grafika, dane aplikacji.
- ☼ Informacja o typie pliku może być przechowywana w strukturach wewnętrznych systemu plików, w zawartości samego pliku, w katalogach lub w nazwie pliku.
- ☼ Typ pliku może być rozpoznawany przez system operacyjny, ale może to być również tylko informacja interpretowana przez użytkownika lub aplikację.

## Struktura pliku

- ☼ Struktura pliku określa jego wewnętrzną organizację.
- ☼ Struktura może być definiowana i rozpoznawana przez system operacyjny lub może być rozpoznawana na poziomie aplikacji korzystającej z tego pliku.
- ☼ Definiowanie różnych struktur plików na poziomie systemu operacyjnego może być pomocne dla użytkownika, ale w systemie musi być wówczas zawarty kod do obsługi każdej z tych struktur.
- ☼ Wyróżniana jest też wewnętrzna (fizyczna) struktura plików, narzucana przez urządzenie, które ten plik przechowuje

## Odwzorowanie obrazu logicznego w fizyczny



Systemy operacyjne

System plików

7

## Metody dostępu do plików

- Metody dostępu do plików określają sposób wykonywania operacji na plikach w celu udostępnienia znajdującej się w nich informacji.

Systemy operacyjne

System plików

8

## Wyszczególnienie metod dostępu

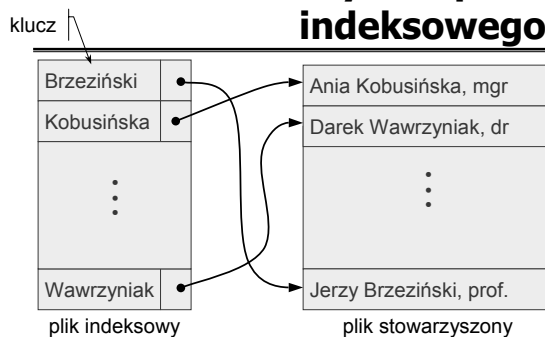
- Dostęp sekwencyjny (ang. sequential access) — informacje w pliku przetwarzane są rekord po rekordzie, tzn. po wykonaniu operacji na określonym rekordzie system przygotowuje się do wykonania operacji na kolejnym rekordzie w pliku.
- Dostęp bezpośredni (ang. direct access) — lokalizacja rekordu do przetwarzania podawana jest jako parametr odpowiedniej operacji.
- Dostęp indeksowy — rekord, na którym ma być wykonana operacja identyfikowany jest przez klucz, odwzorowywany na konkretny rekord w pliku stowarzyszonym poprzez plik indeksowy.

Systemy operacyjne

System plików

9

## Przykład pliku indeksowego



Systemy operacyjne

System plików

10

## Podstawowe operacje na plikach (1)

- Tworzenie pliku — konieczne jest określenie podstawowych atrybutów pliku, znalezienie miejsca na ten plik w systemie komputerowym oraz jego zaewidencjonowanie (utworzenie wpisu katalogowego)
- Zapis do pliku — konieczne jest określenie, co ma być zapisane i gdzie ma być zapisane (w którym pliku i w jakim miejscu tego plik, zależnie od sposobu dostępu)
- Odczyt z pliku — konieczne jest określenie, co ma być odczytane (z którego pliku i z jakiego miejsca tego plik, zależnie od sposobu dostępu) i gdzie mają być umieszczone odczytane dane

Systemy operacyjne

System plików

11

## Podstawowe operacje na plikach (2)

- Usuwanie informacji z pliku — należy określić jaki fragment pliku (i którego pliku) ma być usunięty. Najczęściej możliwe jest tylko skracanie pliku, czyli usuwanie jego końcowej zawartości lub całej jego zawartości.
- Usuwanie pliku — należy określić plik do usunięcia. Usuwana jest zawartość oraz wpis ewidencyjny pliku.
- Dodatkowe operacje na plikach, wykonywane w celu uzyskania dostępu do zawartości pliku:
  - otwieranie,
  - zamykanie,
  - przesuwanie wskaźnika bieżącej pozycji.

Systemy operacyjne

System plików

12

## Interfejs dostępu do pliku w systemie UNIX (1)

- ✿ Tworzenie pliku — funkcja systemowa **creat**
- ✿ Usuwanie dowiązania do pliku — funkcja systemowa **unlink**
- ✿ Odczyt z pliku — funkcja systemowa **read**
- ✿ Zapis do pliku — funkcja systemowa **write**
- ✿ Skracanie pliku (usuwanie końcowej zawartości, obcinanie) — funkcja systemowa **truncate**

## Interfejs dostępu do pliku w systemie UNIX

- ✿ Przesunięcie wskaźnika bieżącej pozycji — funkcja systemowa **lseek**
- ✿ Otwieranie pliku — funkcja systemowa **open**
- ✿ Zamykanie otwartego pliku — funkcja systemowa **close**

## System UNIX — tworzenie pliku

- ✿ Funkcja systemowa: **creat**
- ✿ Parametry:
  - nazwa pliku (ścieżka)
  - prawa dostępu
- ✿ Wartość zwrotna: deskryptor pliku lub **-1** w przypadku błędu.
- ✿ Uwaga: funkcja tworzy plik i otwiera go do zapisu.

## System UNIX — otwieranie pliku

- ✿ Funkcja systemowa: **open**
- ✿ Parametry:
  - nazwa pliku (ścieżka)
  - tryb otwarcia (do zapisu, do odczytu itp.)
- ✿ Wartość zwrotna: deskryptor pliku lub **-1** w przypadku błędu.
- ✿ Uwaga: funkcja występuje również w wersji z 3-argumentowej, umożliwiającej tworzenie pliku.

## System UNIX — zamykanie deskryptora pliku

- ✿ Funkcja systemowa: **close**
- ✿ Parametry:
  - deskryptor
- ✿ Wartość zwrotna: 0 w przypadku poprawnego zakończenia lub **-1** w przypadku błędu.

## System UNIX — usuwanie dowiązania do pliku

- ✿ Funkcja systemowa: **unlink**
- ✿ Parametry:
  - nazwa pliku (ścieżka)
- ✿ Wartość zwrotna: 0 w przypadku poprawnego zakończenia lub **-1** w przypadku błędu.

## System UNIX — skracanie pliku

- ❁ Funkcja systemowa: **truncate**
- ❁ Parametry:
  - nazwa pliku (ścieżka) lub deskryptor
  - wielkość w bajtach do jakiej ma nastąpić skrócenie (docelowa wielkość pliku)
- ❁ Wartość zwrrotna: 0 w przypadku poprawnego zakończenia lub -1 w przypadku błędu.

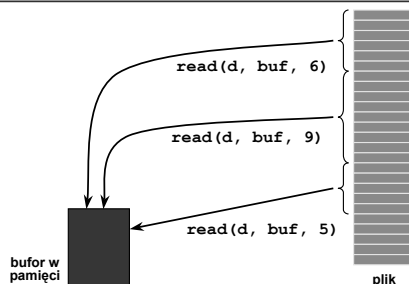
## System UNIX — odczyt zawartości pliku

- ❁ Funkcja systemowa: **read**
- ❁ Parametry:
  - deskryptor pliku
  - adres w pamięci, pod którym zostaną zapisane dane odczytane z pliku
  - liczba odczytywanych bajtów
- ❁ Wartość zwrrotna: liczba rzeczywiście odczytanych bajtów lub -1 w przypadku błędu.

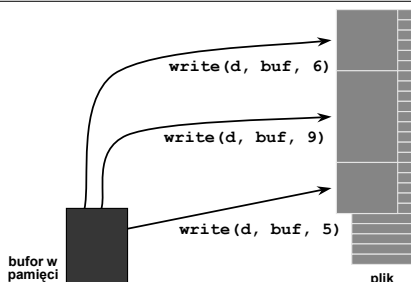
## System UNIX — zapis zawartości pliku

- ❁ Funkcja systemowa: **write**
- ❁ Parametry:
  - deskryptor pliku
  - adres w pamięci, spod którego zostaną pobrane dane do zapisu
  - liczba zapisywanych bajtów
- ❁ Wartość zwrrotna: liczba rzeczywiście zapisanych bajtów lub -1 w przypadku błędu.

## Przykład realizacji operacji odczytu w systemie UNIX



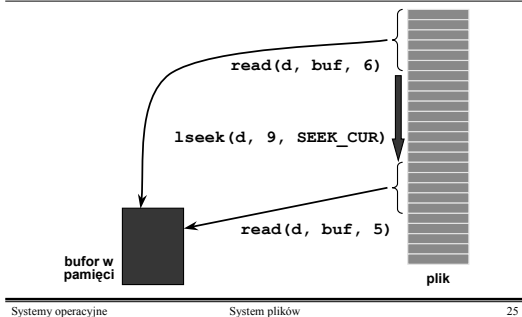
## Przykład realizacji operacji zapisu w systemie UNIX



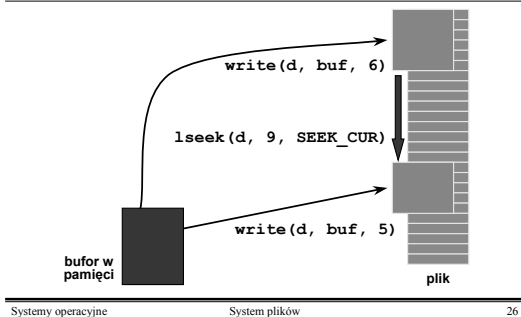
## System UNIX — zmiana wskazania bieżącej pozycji

- ❁ Funkcja systemowa: **lseek**
- ❁ Parametry:
  - deskryptor pliku
  - wielkość przesunięcia w bajtach
  - punkt odniesienia (początek pliku, bieżąca pozycja, lub koniec pliku)
- ❁ Wartość zwrrotna: położenie wskaźnika po przesunięciu lub -1 w przypadku błędu.

## Zmiana wskazania bieżącej pozycji (1)



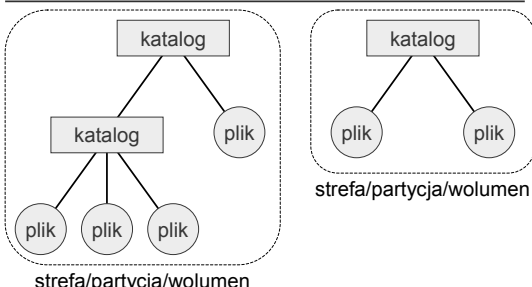
## Zmiana wskazania bieżącej pozycji (2)



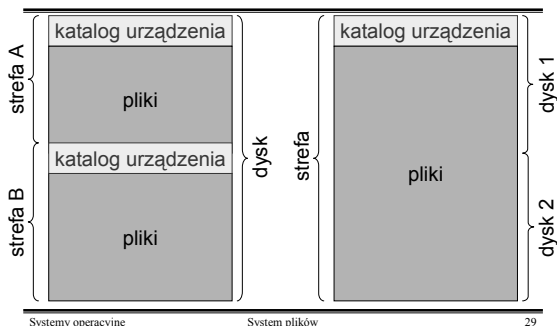
## Organizacja systemu plików (1)

- ☼ Podział na strefy (wolumeny, tomy, partycje)
  - strefa obejmuje część dysku, jeden dysk lub kilka dysków
  - strefa zawiera pliki i katalogi
- ☼ Organizacja katalogów:
  - katalog jest tablicą kojarzącą nazwy plików z wpisami katalogowymi, obejmującymi inne atrybuty plików,
  - katalogi mogą być jedno- lub wielopoziomowe,
  - katalogi wielopoziomowe zorganizowane mogą być w różne struktury logiczne (drzewo, graf acykliczny, dowolny graf),
- ☼ Pliki identyfikowane są przez nazwy, znajdujące się w katalogach.

## Organizacja systemu plików (2)



## Podział na strefy



## Operacje na katalogu

- ☼ Tworzenie wpisu katalogowego, gdy tworzony jest plik lub alternatywnej jego nazwy
- ☼ Usuwanie wpisu katalogowego
- ☼ Odnajdowanie wpisu katalogowego
- ☼ Tworzenie wykazu wpisów katalogowych (listing zawartości)
- ☼ Przemianowanie pliku (zmiana nazwy)

## Struktura logiczna katalogów

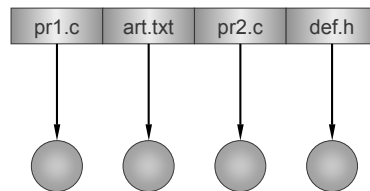
- Struktura jednopoziomowa — wpisy katalogowe poszczególnych plików znajdują się w tym samym katalogu (na tym samym poziomie).
- Struktura dwupoziomowa — wpisy katalogowe plików znajdują się w różnych katalogach, ale katalogi nie mogą zawierać innych katalogów.
- Struktura drzewiasta — w katalogach można tworzyć podkatalogi oraz pliki.
- Graf acykliczny — podkatalog (lub plik) może być umieszczony w wielu katalogach.
- Graf ogólny — dopuszcza się cykl w powiązaniach pomiędzy katalogami

Systemy operacyjne

System plików

31

## Struktura jednopoziomowa

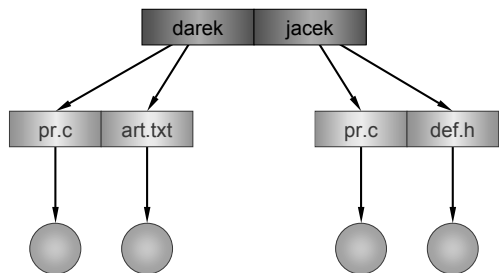


Systemy operacyjne

System plików

32

## Struktura dwupoziomowa

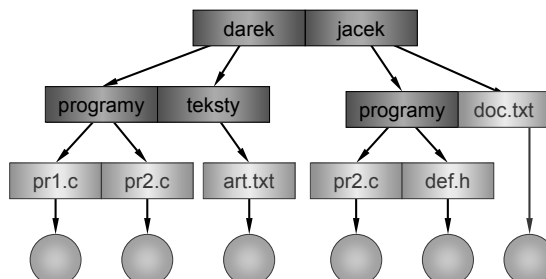


Systemy operacyjne

System plików

33

## Struktura drzewiasta

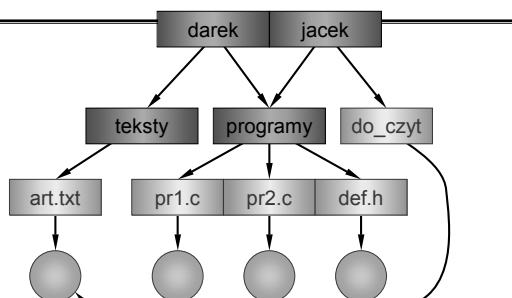


Systemy operacyjne

System plików

34

## Graf acykliczny

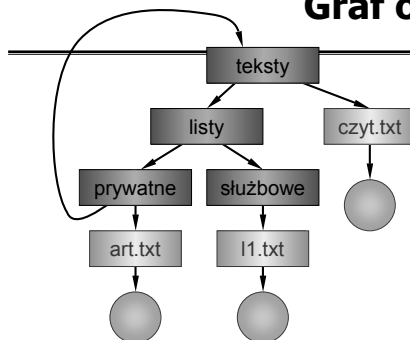


Systemy operacyjne

System plików

35

## Graf ogólny



Systemy operacyjne

System plików

36

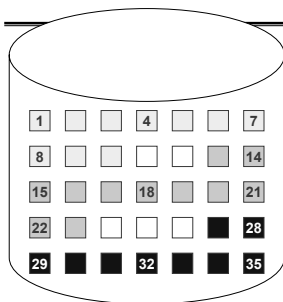
# Organizacja fizyczna systemu plików

- ✿ Przydział miejsca na dysku
  - przydział ciągły, przydział listowy, przydział indeksowy
- ✿ Zarządzanie wolną przestrzenią
  - wektor bitowy, lista powiązana, grupowanie, zliczanie
- ✿ Implementacja katalogu
  - lista liniowa, tablica haszowa, struktura indeksowa

# Przydział miejsca na dysku

- ✿ Przydział ciągły (ang. contiguous allocation) — cały plik zajmuje ciąg kolejnych bloków (jednostek alokacji)
- ✿ Przydział listowy (ang. linked allocation) — plik jest listą powiązanych bloków, dowolnie rozproszonych w dostępnej przestrzeni dyskowej
- ✿ Przydział indeksowy (ang. indexed allocation) — wskaźniki do rozproszonych bloków dyskowych (indeksy) skupione są w jednym miejscu, w tzw. bloku indeksowym
  - schemat listowy bloku indeksowego
  - indeks wielopoziomowy
  - schemat kombinowany

## Przydział ciągły



### Katalog:

blok początkowy: 1  
rozmiar: 10 bloków

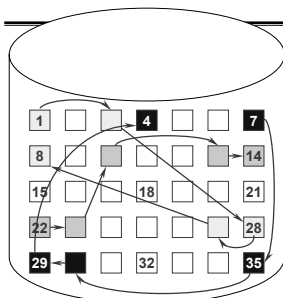
blok początkowy: 13  
rozmiar: 11 bloków

blok początkowy: 27  
rozmiar: 9 bloków

## Przydział ciągły — właściwości

- ✿ Efektywność dostępu (niewielkie ruchy głowic dysk.)
- ✿ Łatwa lokalizacja bloków pliku zarówno przy dostępie sekwencyjnym jak i swobodnym
- ✿ Problem fragmentacji zewnętrznej — po usuniętych plikach pozostają dziury, które trudno połączyć w jeden większy blok
- ✿ Problem rozszerzania pliku
  - pliku nie da się rozszerzyć,
  - będzie go trzeba przenieść w nowe miejsce (znaleźć większą dziurę)
  - będzie trzeba z góry zarezerwować więcej miejsca w przestrzeni dyskowej

## Przydział listowy



### Katalog:

blok początkowy: 1  
blok końcowy: 8

blok początkowy: 22  
blok końcowy: 14

blok początkowy: 7  
blok końcowy: 4

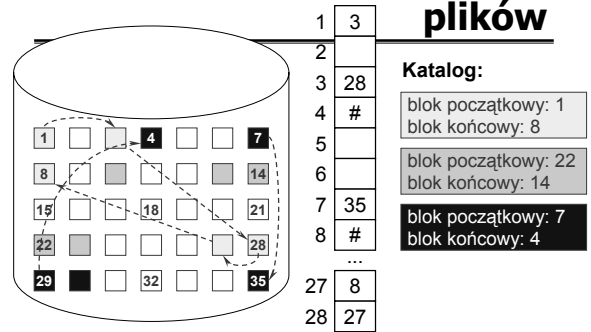
## Przydział listowy — właściwości

- ✿ Nie ma problemu fragmentacji zewnętrznej i wynikających z niej wad metody przydziału
- ✿ Łatwa realizacja dostępu sekwencyjnego
- ✿ Problem realizacji dostępu swobodnego (bezpośredniego) — przejście do poprzedniego bloku wymaga rozpoczęcia przeglądania listy bloków od początku
- ✿ Konieczność pamiętania wewnątrz bloku wskaźnika do bloku następnego
- ✿ Niezawodność — utrata jednego bloku pociąga za sobą stratę wszystkich następnych

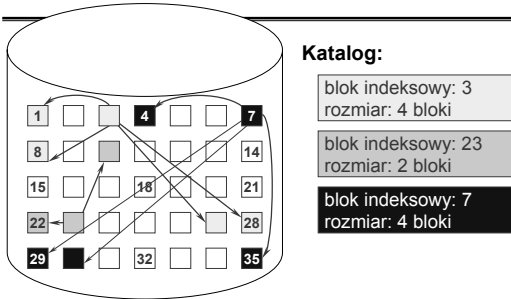
## Tablica alokacji plików — FAT (File Allocation Table)

- FAT jest dodatkową strukturą (tablicą) umieszczoną w odpowiednim obszarze na dysku
- Każdy element tablicy FAT odpowiada dokładnie jednej jednostce alokacji (blokowi) z przestrzeni bloków plikowych i indeksowany jest numerem bloku
- Element tablicy FAT zawiera indeks następnego bloku przydzielonego danemu plikowi lub pewną wartość specjalną oznaczającą wolną pozycję lub ostatnią pozycję danego pliku

## Struktury tablicy alokacji plików



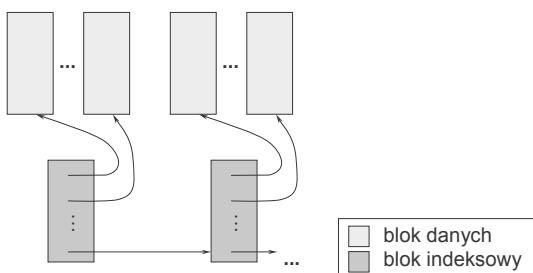
## Przydział indeksowy



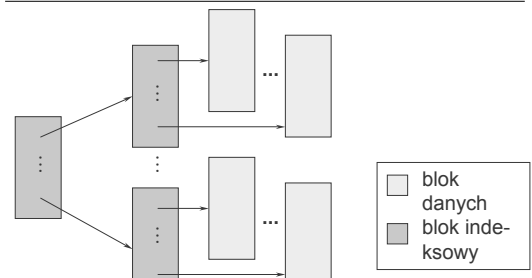
## Struktura bloku indeksowego

- Schemat listowy — w ostatnim elemencie bloku indeksowego znajduje się wskaźnik do następnego bloku indeksowego tego pliku.
- Indeks wielopoziomowy — blok indeksowy pierwszego poziomu zawiera wskaźnik do bloków drugiego poziomu itd.
- Schemat kombinowany — zastosowanie do pewnej liczby bloków indeksu pierwszego poziomu, dla następnych bloków indeksu dwupoziomowego itp.

## Struktura bloku indeks. — schemat listowy



## Struktura bloku indeks. — indeks wielopoziomowy





## Struktura bloku indeks. — schemat kombinowany



Systemy operacyjne

System plików

49

## Przydział indeksowy — właściwości

- Łatwa lokalizacja bloków pliku zarówno przy dostępie sekwencyjnym jak i swobodnym
- Łatwa realizacja dostępu swobodnego (bezpośredniego)
- Brak problemu fragmentacji zewnętrznej
- Konieczność przeznaczenie pewnej części przestrzeni dyskowej na bloki indeksowe

Systemy operacyjne

System plików

50

## Zarządzanie wolną przestrzenią

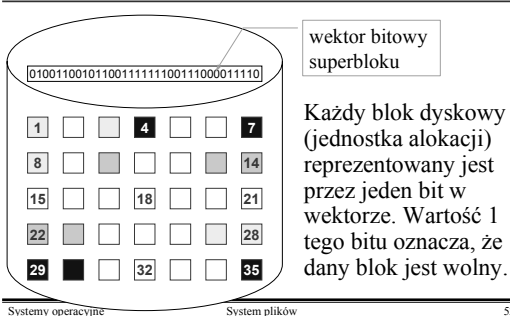
- Wektor bitowy — każdy bit odpowiada jednemu blokowi, wartość 1 oznacza wolny blok.
- Lista powiązana — każdy wolny blok zawiera indeks następnego wolnego bloku.
- Grupowanie — niektóre wolne bloki wypełnione są w całości indeksami innych wolnych bloków, ostatni indeks wskazuje na kolejny blok wypełniony w całości indeksami.
- Zliczanie — wykaz wolnych bloków obejmuje indeks pierwszego wolnego bloku oraz liczbę wolnych bloków znajdujących się za nim, stanowiących ciągły obszar.

Systemy operacyjne

System plików

51

## Zarządzanie wolną przestrzenią — wektor bit.

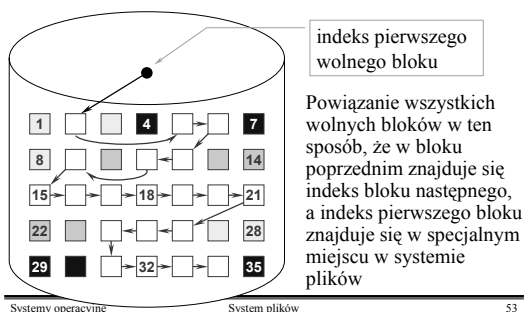


Systemy operacyjne

System plików

52

## Zarządzanie wolną przestrzenią — lista pow.

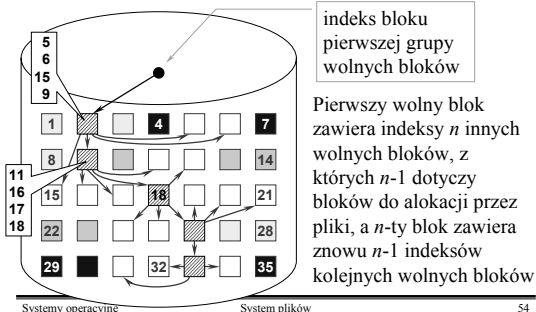


Systemy operacyjne

System plików

53

## Zarządzanie wolną przestrzenią — grupowanie



Systemy operacyjne

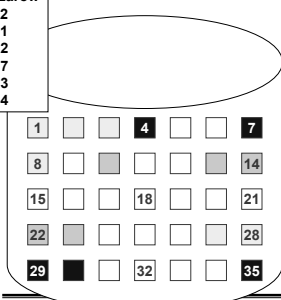
System plików

54

## Zarządzanie wolną przestrzenią — zliczanie

Wykaz wolnych obszarów

5, 2  
9, 1  
11, 2  
15, 7  
24, 3  
31, 4



W przypadku kilku kolejnych (przylegających do siebie) wolnych bloków pamiętany jest tylko indeks pierwszego z nich oraz liczba wolnych bloków znajdujących się bezpośrednio za nim. Wykaz wolnych obszarów jest ciągiem wpisów, składających się z indeksu bloku oraz licznika.

Systemy operacyjne

System plików

55

## Implementacja katalogu — lista liniowa

- Katalog składa się z ciągu wpisów katalogowych ogólnej postaci:

nazwa pliku	inne atrybuty
-------------	---------------

- Lokalizacja wpisu polega na przeszukiwaniu liniowym (sprawdzone są kolejne pozycje, począwszy od pierwszej)
- Lokalizację wpisu można przyspieszyć poprzez posortowanie wg. nazwy, jednak utrzymanie takiej struktury jest kosztowne.

Systemy operacyjne

System plików

56

## Implementacja katalogu — tablica haszowa

- Wpisy ułożone są na pozycjach odpowiadających wartościom funkcji haszującej.
- Funkcja haszująca odwzorowuje nazwę pliku na wartość z określonego przedziału, traktowaną jako indeks wpisu.
- Ta sama funkcja haszująca wykorzystywana jest do lokalizacji wpisu.
- W katalogu mogą być potrzebne dodatkowe struktury pomocne przy usuwaniu konfliktów.

Systemy operacyjne

System plików

57

## Implementacja katalogu — struktura indeksowa

- Wpisy katalogowe powiązane są w strukturę drzewiastą przyspieszającą wyszukiwanie (np. drzewo binarne, B-drzewo, B+-drzewo).
- Lokalizacja wpisu polega na przejściu drzewa zgodnie z zasadami jego budowy.
- Struktura drzewa jest zoptymalizowana w taki sposób, żeby minimalizować liczbę operacji dyskowych podczas przeszukiwania.

Systemy operacyjne

System plików

58

## Przechowywanie podręczne w sys. plików

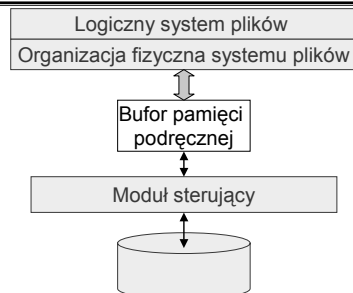
- Operacja dostępu do danych w pliku wymaga ich sprowadzenia do pamięci operacyjnej, gdzie można je testować, zmieniać, po czym zażądać ponownego ich zapisania do systemu plików.
- Czytanie i pisanie bezpośrednio z / na dysk podczas wszystkich operacji dostępu do plików, jest nieefektywne ze względu na czas reakcji dysku oraz relatywnie małą szybkość transmisji dyskowych.
- Minimalizacja dostępu do dysku możliwa jest przez utrzymywanie puli wewnętrznych buforów, zwanych podręczną pamięcią buforową (ang. buffer cache), które zawierają dane z ostatnio używanych bloków dyskowych.

Systemy operacyjne

System plików

59

## Przechowywanie podręczne w sys. plików



Systemy operacyjne

System plików

60

## Zasady przechowywania podręcznego

- ❁ Bloki dyskowe aktualnie wykorzystywane są utrzymywane w pamięci buffer cache — dane w jednym buforze odpowiadają danym z jednego bloku dyskowego.
- ❁ Obsługa żądania odczytu bloku polega najpierw na sprawdzeniu czy dany blok znajduje się w pamięci buffer cache.
- ❁ Jeśli blok nie znajduje się w pamięci buffer cache jest czytany z dysku do tej pamięci, a następnie kopiowany w odpowiednie miejsce w przestrzeni adresowej procesu.
- ❁ Dane zapisywane na dysk są również zapamiętywane w pamięci buforowej, by były tam dostępne dla ewentualnych kolejnych operacji odczytu.

## Przechow. podręczne w realizacji operacji odczytu

1. Znajdź adres bloku dyskowego, zawierającego fragment pliku, którego odczytu zażądano.
2. Przekopiuj zawartość tego bloku do bufora systemu plików w pamięci głównej (jeśli ten blok się tam jeszcze nie znajduje).
3. Przekopiuj żądany fragment z bufora do przestrzeni adresowej procesu.

## Przechow. podręczne w realizacji operacji zapisu

1. Znajdź adres bloku dyskowego, zawierającego fragment pliku, którego zapisu zażądano.
2. Przekopiuj zawartość tego bloku do bufora w pamięci głównej (jeśli ten blok się tam jeszcze nie znajduje).
3. Przekopiuj żądany fragment z przestrzeni adresowej procesu do bufora.
4. Zapisz na dysk uaktualniony blok z bufora (albo w trybie natychmiastowym albo z opóźnieniem)

## Integralność systemu plików

- ❁ W wyniku awarii systemu zawartość podręcznej pamięci buforowej może nie zostać zapisana na dysku lub może zostać zapisana tylko częściowo.
- ❁ Skutkiem w/w awarii może być pozostawienie systemu plików w stanie niespójnym.
- ❁ Większość systemów operacyjnych dostarcza odpowiednie narzędzie do sprawdzania integralności systemu plików, uruchamiane w ramach restartu systemu po awarii.

## Przejawy braku integralności sys. plików

- ❁ Brak bloku zarówno w wykazie bloków zaalokowanych jak i bloków wolnych
- ❁ Obecność bloku zarówno w wykazie bloków zaalokowanych jak i bloków wolnych
- ❁ Wielokrotne powtórzenie się bloku w wykazie bloków wolnych (duplikacja wolnego bloku)
- ❁ Wielokrotne powtórzenie się bloku w wykazie bloków zaalokowanych (duplikacja zaalokowanego bloku)
- ❁ Niespójność informacji we wpisach katalogowych (np. niezgodność licznika dowiązań w systemie UNIX).

## Semantyka spójności

- ❁ Semantyka spójności określa sposób postrzegania zmian zawartości pliku, dokonywanych przez współbieżnie działające (i korzystające z danego pliku) procesy.
- ❁ Przykłady semantyki spójności:
  - semantyka spójności systemu UNIX — wynik operacji zapisu jest natychmiast widoczny dla innych procesów, które otworzyły dany plik,
  - semantyka sesji — zmiany w pliku stają się widoczne tylko dla procesów, które ten plik otworzą po jego zamknięciu przez proces zapisujący,
  - semantyka stałych plików dzielonych — plik dzielony nie może podlegać modyfikacjom, czyli może być tylko czytany.

## Synchronizacja dostępu do plików

- Współbieżny dostęp do zawartości pliku można synchronizować na poziomie całego pliku lub poszczególnych jego fragmentów (zajmowanie rekordów).
- Najczęściej dopuszcza się dwa rodzaje blokad
  - blokada współdzielona (shared lock, read lock)
  - blokada wyłączna (exclusive lock, write lock)

## Synchronizacja dostępu do plików — zgodność blokad

	shared	exclusive
shared	zgodne	wyklucz.
exclusive	wyklucz.	wyklucz.

## Przykłady implementacji systemu plików

- CP/M — katalog zawiera blok kontrolny pliku (FCB), identyfikujący 16 jednostek alokacji (zawierający indeksy tych jednostek alokacji).
- DOS — wpis katalogowy zawiera indeks pierwszej jednostki alokacji, a pozostałe jednostki wynikają z tablicy FAT.
- ISO 9660 (CD ROM) — przydział ciągły, wpis katalogowy zawiera indeks pierwszej jednostki alokacji oraz rozmiar pliku, wpisy katalogowe są posortowane alfabetycznie.

## Przykłady implementacji systemu plików

- UNIX — pliki identyfikowane są przez i-węzły, wpis katalogowy zawiera indeks i-węzła, który z kolei zawiera indeks (kombinowany) jednostek alokacji.
- NTFS — pliki identyfikowane są przez referencje, która jest indeksem rekordu w tablicy MFT, rekord zawiera atrybuty pliku (w szczególności dane) lub odnośniki do bloków z atrybutami.

## CP/M — informacje ogólne

- Jednopoziomowy katalog zawiera 32-bajtowe wpisy, tzw. bloki kontrolne plików (FCB), obejmujące prawie wszystkie (w niektórych przypadkach wszystkie) atrybuty pliku.
- W katalogu znajdują się wpisy wszystkich plików w systemie.
- Dane (zawartość pliku) znajdują się w 128-bajtowych sektorach, alokacji podlegają jednak bloki o rozmiarach 1KB.
- Z każdym wpisem związany jest jednobajtowy kod użytkownika, w związku z czym każdy użytkownik widzi tylko swoje pliki (katalog dwupoziomowy).
- Zarządzanie wolną przestrzenią przy użyciu wektora bitowego przechowywanego w pamięci wyliczanego każdorazowo po zmianie nośnika.

## CP/M — format partycji





## UNIX — informacje ogólne

- ✿ Z każdym plikiem związany jest i-węzeł, który przechowuje wszystkie atrybuty pliku z wyjątkiem nazwy.
- ✿ Nazwa znajduje się w katalogu obok numeru i-węzła danego pliku.
- ✿ Katalogi tworzą strukturę wielopoziomową (katalog zawiera wpis specyfikujący inny katalog).
- ✿ Dane (zawartość pliku) znajdują się w blokach (jednostkach alokacji) o ustalonym rozmiarze.
- ✿ Bloki identyfikowane są za pośrednictwem indeksu kombinowanego.
- ✿ Wolne bloki wyszczególnione są na liście powiązanej.

Systemy operacyjne

System plików

79

## UNIX — format partycji



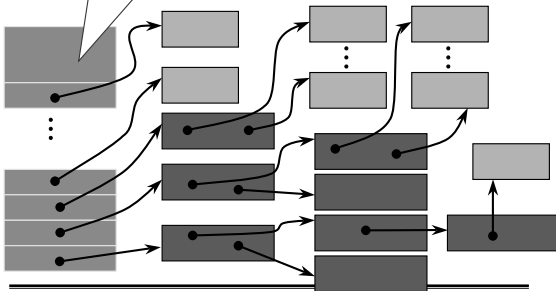
Systemy operacyjne

System plików

80

## UNIX — fizyczna struktura pliku

Atrybuty:  
id właściciela i grupy  
typ pliku  
prawa dostępu  
czaszy dostępu  
licznik dowiązań  
rozmiar w bajtach



Systemy operacyjne

System plików

81

## UNIX — struktura wpisu katalogowego



Systemy operacyjne

System plików

82

## NTFS — informacje ogólne

- ✿ Plik jest zbiorem atrybutów, obejmującym w szczególności atrybut *dane*.
- ✿ Każdy plik ma swój wpis (rekord) w głównej tablicy plików (MFT), a indeks w tej tablicy jest składnikiem identyfikatora wewnętrznego (referencji pliku).
- ✿ Tablica MFT sama jest plikiem, podobnie jak każdy inny obiekt w obrębie systemu plików.
- ✿ Wolne bloki identyfikowane są przez wektor bitowy, przechowywany w jednym z plików.

Systemy operacyjne

System plików

83

## NTFS — format partycji



Systemy operacyjne

System plików

84

## NTFS — struktura rekordu MFT

- Rekord MFT ma ustalony rozmiar 1KB
- Rekord MFT składa się z nagłówka rekordu i sekwencji atrybutów.
- Atrybut składa się z nagłówka atrybutu oraz wartości.
- Wartość znajduje się albo bezpośrednio w rekordzie MFT, albo w rekordzie znajduje się odnośnik do bloku danych, gdzie znajduje się wartość.

nagłówek rekordu MFT	
nagłówek atrybutu	wartość atrybutu
nagłówek atrybutu	wartość atrybutu
⋮	

## NTFS — zawartość nagłówka rekordu

- magiczna liczba — wartość używana do sprawdzania poprawności
- numer sekwencyjny — wartość zwiększana o 1 za każdym razem, gdy rekord używany jest dla nowego pliku
- licznik odniesień do pliku w katalogach
- liczba wykorzystywanych bajtów przez rekord
- identyfikator rekordu bazowego (referencja) w przypadku rekordu rozszerzeń

## NTFS — identyfikacja rekordu MFT

- Rekord MFT i tym samym plik identyfikowany jest wewnętrznie przez 64-bitową referencję.
- 48 bitów referencji jest numerem rekordu w tablicy MFT.
- 16 bitów referencji jest numerem sekwencyjnym zwiększanym o 1 po każdym przydzieleniu rekordu do nowo utworzonego pliku, co zapobiega błędnym odwołaniom do plików z użyciem zdezaktualizowanych referencji.

## NTFS — rekord bazowy

- Jeśli lista atrybutów pliku nie mieści się w jednym rekordzie MFT, do pliku przydzielane są dodatkowe rekordy.
- Pierwszy rekord MFT nazywany jest **bazowym**.
- Dodatkowe rekordy MFT danego pliku określane są jako rekordy rozszerzeń.

## NTFS — wybrane atrybuty pliku

- informacje standardowe — flagi, czasy itp.
- nazwa pliku
- deskryptor bezpieczeństwa
- lista atrybutów — wskazanie dodatkowych rekordów MFT
- identyfikator pliku (referencja)
- dane — strumień danych stanowiących zawartość pliku

## NTFS — wybrane metadane

- \$MFT — główna tablica plików
- \$MTFMIRR — kopia MFT
- \$LOGFILE — plik dziennika
- \$VOLUME — plik wolumenu
- \$DEFATTR — definicja atrybutów plików
- \$ — katalog główny (korzeń drzewa)
- \$BITMAP — wektor bitowy (informacja o wolnych/zajętych blokach)

## NTFS — katalogi

---

☼ Katalog jest plikiem zawierającym ciąg wpisów, w skład których wchodzi wartości wybranych atrybutów:

- referencja pliku
- nazwa
- rozmiar
- czasy dostępu

☼ Implementacja katalogu oparta jest na strukturze indeksowej (B+-drzewo).