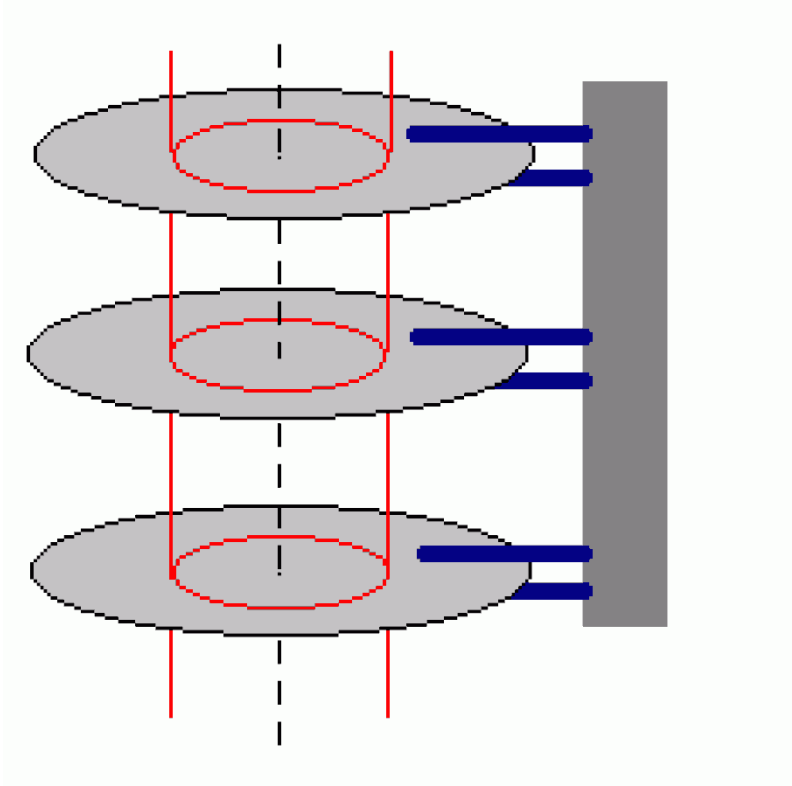
Systemy operacyjne.
System plików - interfejs i realizacja

Wykład 7

Jarosław Koźlak

Budowa dysku twardego

- Dysk twardy składa się z kilku głównych elementów. Na poniższym rysunku zaznaczono najważniejsze z nich.

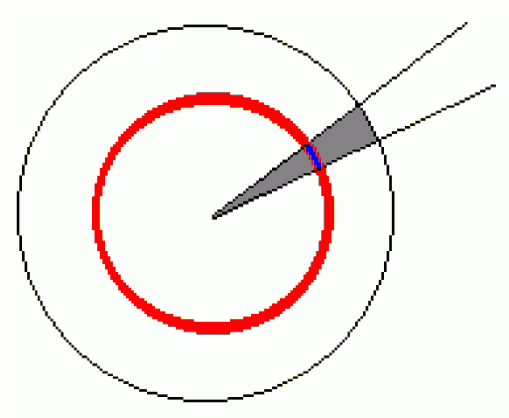


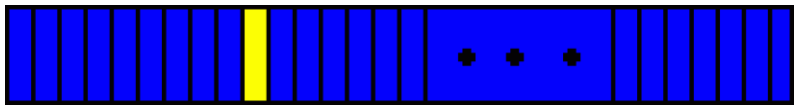
Adresacja fizyczna dysku

- Aby określić fizyczny adres konkretnego bajtu na dysku należy podać następujące wartości:
  - numer głowicy,
  - numer ścieżki (cylindra),
  - sektor fizyczny,
  - offset.

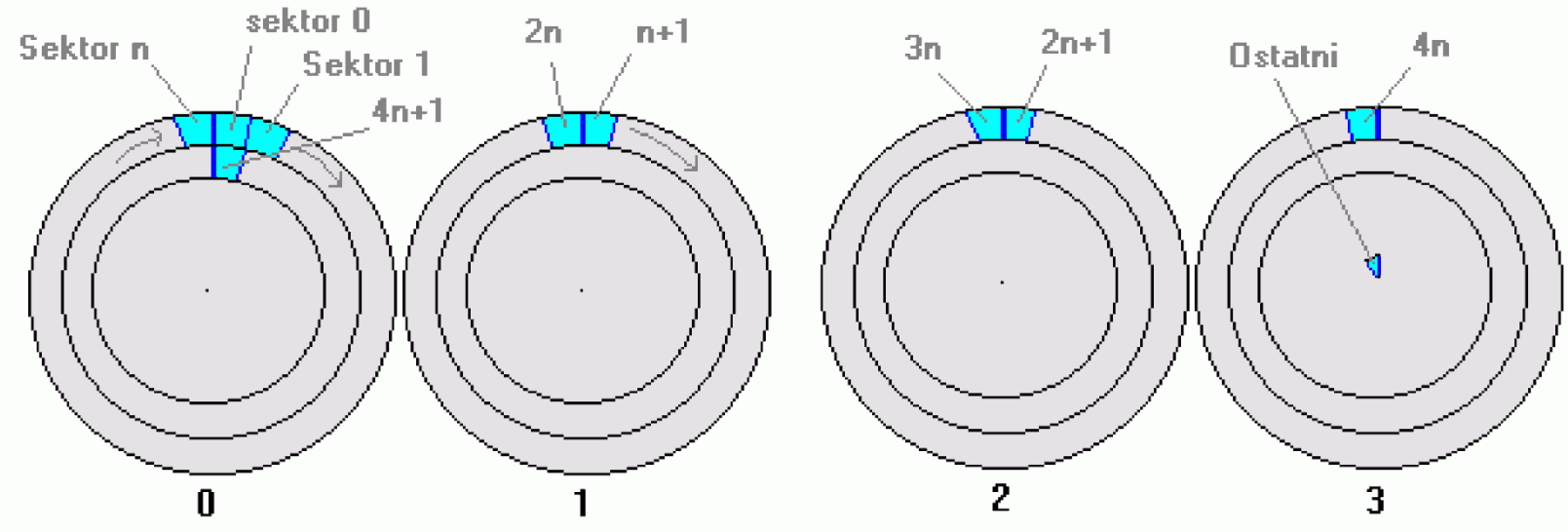
Adresacja fizyczna dysku

- Numer głowicy określa powierzchnię, na której znajduje się szukane miejsce. Mamy tyle samo powierzchni co głowic.
- Numer ścieżki określa odległość szukanego miejsca od brzegu dysku.
- Sektor fizyczny
- Offset - każdy sektor składa się z 512 bajtów Poszczególne niebieskie pola przedstawiają konkretne bajty. Żółte pole to przykładowy szukany bajt.





Adresacja logiczna



- Dwa talerze dysku, każdy z dwóch stron (4 powierzchnie).
- Sektor o numerze 0 znajduje się na powierzchni o numerze 0.
- Kolejne sektory znajdują się na tej samej ścieżce po kolei zgodnie z ruchem wskazówek zegara.
- Umieszczenie jak największej liczby kolejnych sektorów na jednym cylindrze zmniejsza liczbę potrzebnych przesunięć głowic.

Czym jest system plików?

- System plików to metody i struktury danych używane przez system operacyjny używane w celu zapisania informacji o plikach i ich zawartości na danej partycji; jest to sposób organizacji plików na dysku. Słowo to jest używane również w znaczeniu dysku, partycji. Może to być nieco mylące.

- System plików (ang. *filesystem*) to warstwa systemu operacyjnego pośrednicząca w dostępie do informacji zapisanej na pamięci masowej, a więc do plików.

Elementy składowe systemu plików

- **System plików:**
  - tworzy mechanizm przechowywania informacji i bezpośredniego dostępu do danych i programów

- **Elementy składowe systemu plików:**
  - zbiór plików - zawierających powiązane informacje; plik - logiczna jednostka magazynowania informacji
  - struktura katalogów - organizacja informacji, dostęp do plików
  - strefy (partycje, ang. *partitions*) - służą do wyodrębnienia fizycznie lub logicznie zbiorów katalogów (niektóre systemy plików)

Typowe atrybuty pliku:

- nazwa
- typ
- właściciel
- położenie (określane przez urządzenie i położenie pliku na urządzeniu)
- rozmiar (bieżący lub maksymalny dopuszczalny)
- prawa dostępu
- czasy odwołań do pliku (czas utworzenia, ostatnia zmiana, ostatnie użycie) wraz z informacją odnośnie użytkownika, który odwołań dokonał

Operacje plikowe

- tworzenie pliku (ang. *creating*)
- zapisywanie pliku (ang. *write*)
- czytanie pliku (ang. *reading*)
- zmiana pozycji w pliku (ang. *repositioning*), przeszukiwanie (ang. *seek*) pliku
- usuwanie pliku (ang. *deleting*)
- skracanie pliku (ang. *truncating*)
- dopisywanie informacji do pliku (ang. *appending*)
- przemianowywanie pliku (ang. *renaming*)

Informacje o otwartych plikach

- otwieranie/zamykanie pliku (aby uniknąć ciągłego przeszukiwania katalogu)
- tablica otwartych plików - tablica z informacjami o wszystkich otwartych plikach, przechowywana przez system operacyjny
- odwołując się do otwartego pliku używa się indeksu do tablicy otwartych plików
- czasem plik może być jednocześnie otwarty przez kilka procesów

- poziomy tablic wewnętrznych opisujących otwarte pliki
  - *procesowa tablica wszystkich plików otwartych w danym procesie* (zawiera bieżące wskaźniki dla każdego pliku i wskazania do ogólnosystemowej tablicy otwartych plików)
  - *ogólnosystemowa tablica otwartych plików* (zawiera informacje niezależne od procesów, jak położenie pliku na dysku, daty dostępu, rozmiar pliku, licznik otwarć określający w ilu procesach dany plik jest otwarty)

Metody dostępu do plików

Dostęp sekwencyjny (ang. sequential access)

- informacje przetwarzane kolejno, rekord po rekordzie
- oparty na taśmowym modelu pliku
- większość operacji na plikach używa dostępu sekwencyjnego

Dostęp bezpośredni (ang. direct access), dostęp względny (ang. relative access)

- plik jest złożony z rekordów logicznych o stałym rozmiarze
- rekordy mogą być odczytywane i zapisywane przez programy w dowolnym porządku
- oparty na dyskowym modelu pliku

Dostęp przez indeks

- indeks pliku zawierający wskaźniki do różnych bloków
- szukając pozycji w pliku najpierw przeszukuje się indeks, skąd poprzez wskaźnik dociera się do potrzebnego bloku w pliku
- indeksy mogą być kilkupoziomowe

Struktura katalogu

Podział systemu plików:

- strefy (ang. partycje), minidyski (ang. minidiscs), tomy/woluminy (ang. volumes)
- na ogół każdy dysk fizyczny zawiera co najmniej jedną strefę, mogą istnieć strefy mieszczące się na wielu dyskach

Informacje zawarte w strefie:

- zgromadzone w katalogu urządzenia (ang. device directory)/ tablicy informacje o zawartości tomu
- dla każdego pliku zawierają:
- nazwę
- położenie
- rozmiar
- typ

Operacje na katalogu

- odnajdywanie pliku
- tworzenie pliku
- usuwanie pliku
- prezentacja zawartości katalogu
- zmiana nazwy pliku
- archiwizacja plików

Katalog o strukturach drzewiastych

- rozwinięcie drzewa wyobrażającego katalog wielopoziomowy w drzewa wielopoziomowe
- katalog główny – korzeń drzewa
- katalog bieżący (ang. *current directory*), operacja zmiany katalogu
- możliwość tworzenia podkatalogów przed użytkowników – pomoc w grupowaniu plików

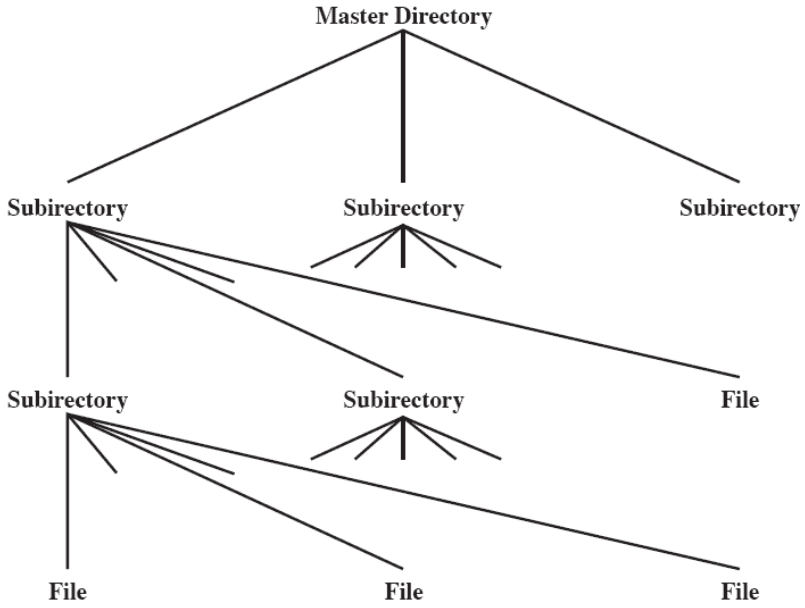


Figure 12.4 Tree-Structured Directory

Acykliczne grafy katalogów

- własności:
  - możliwość umieszczenia pliku/katalogu w wielu katalogach
  - wspiera dzielenie pliku (np. między programistów)
  - wprowadzenie dowiązań (linków) – wskaźników do innych plików i katalogów
- skutki
  - jednemu plikowi może odpowiadać wiele bezwzględnych nazw ścieżek
  - jeśli przeglądamy cały system plików, należy zadbać o uniknięcie podwójnego przeglądania,
  - usuwanie pliku powinno następować dopiero po usunięciu wszystkich dowiązań (trzeba przechowywać liczbę dowiązań do pliku) (twarde linki w systemie Unix)
- dowiązania symboliczne
  - usunięcie dowiązania nie usuwa pliku, jedynie dowiązanie
  - przykład: w Unixie dowiązania symboliczne pozostają po usunięciu plików

- **Zalety grafu acyklicznego:**
  - prosty algorytm przechodzenia i określania, że do pliku nie ma odniesień
  - podczas przeglądania unika się przechodzenia przez dzielone sekcje grafu

Graf ogólny katalogów (graf z cyklami)

- problem – zależy nam na uniknięciu wielokrotnego przeszukiwania składowej (grozi spadek wydajności i powstanie nieskończonej pętli przeszukiwania)
- licznik odniesień może być różny od 0, nawet jeżeli nie można już odwoływać się do pliku/katalogu (możliwość istnienia pętli w strukturze katalogów)
- konieczne użycie schematu łączenia nieużytków (ang. garbage collection) aby określić, kiedy usunięto ostatnie odniesienie
  - procedura obchodu systemu plików i oznaczania elementów, do których można dotrzeć
  - elementy nie oznaczone zbiera się na wykazie wolnych przestrzeni

Budowa systemu plików

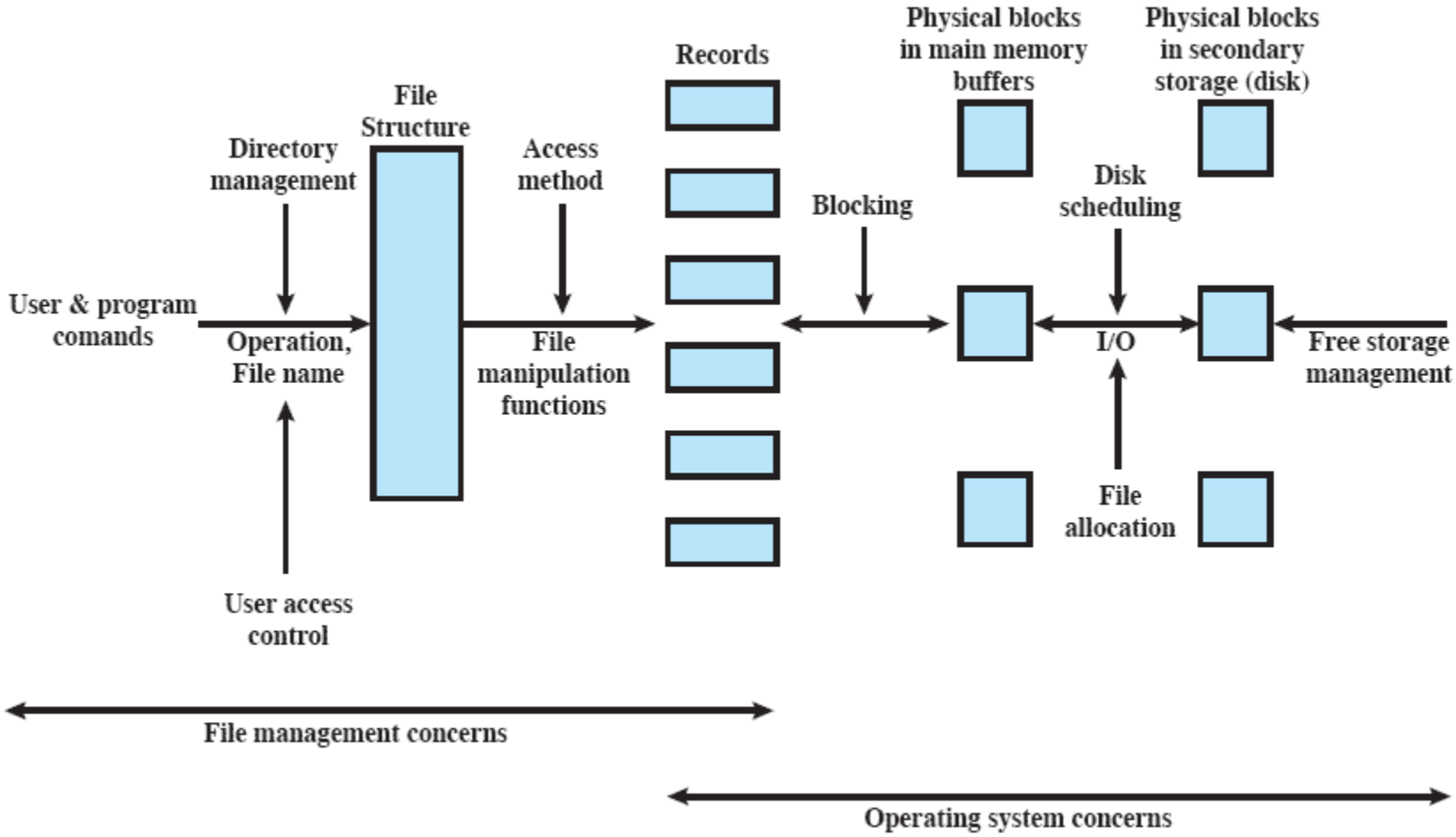
Budowa systemu plików

- rezyduje na stałe w pamięci pomocniczej
- przesyłanie informacji między pamięcią operacyjną a dyskiem odbywa się w jednostkach zwanych blokami (ang. *blocks*)
- blok zawiera jeden lub więcej sektorów
- długość sektora mieści się w granicach 32B - 4096B (zwykle: 512B)

Właściwości dysku:

- informacje mogą być uaktualniane bez zmiany miejsca (można zmodyfikować blok i zapisać go w tym samym miejscu na dysku)
- dowolny blok informacji może być bezpośrednio adresowany

Elementy zarządzania plikami



Warstwowy system plików

- Programy użytkowe
- Logiczny system plików
- Moduł organizacji pliku
- Podstawowy system plików
- Sterowanie wejściem-wyjściem
- Urządzenia

Poziomy warstwowego systemu plików

Sterowanie wejściem-wyjściem (ang. *I/O control***)** -- złożony z modułów obsługi urządzeń i procedur obsługi przerwań związanych z przesyłaniem informacji między pamięcią operacyjną a systemem dyskowym

Moduł obsługi urządzenia (ang. *device driver*) -- translator poleceń wysokiego poziomu na rozkazy zależne od sprzętu używane przez sterownik sprzętowy, łączący urządzenie we/wy z resztą systemu (zazwyczaj dokonuje zapisu /odczytu komórek pamięci sterownika)

Podstawowy system plików (ang. *basic file system*) - wydaje ogólne instrukcje odpowiedniemu modułowi obsługi urządzenia w celu czytania i pisania odpowiednich bloków na dysku, każdy fizyczny blok dysku jest adresowany poprzez swój adres (napęd, cylinder, powierzchnia, sektor,…)

Moduł organizacji pliku (ang. *file-organization module*) interpretuje pliki, ich bloki logiczne i fizyczne

- na podstawie rodzaju stosowanego przydziału miejsca na dysku i położenia pliku tłumaczy adresy logiczne bloków na adresy bloków fizycznych wykorzystywane przez podstawowy system plików; bloki logiczne każdego pliku są ponumerowane 0/1-N
- zawiera zarządcę wolnych obszarów, który odnotowuje nieprzydzielone bloki

Logiczny system plików (ang. *logical file systems*) używa struktury katalogowej, aby na podstawie symbolicznej nazwy pliku dostarczyć informacji potrzebnych modułowi organizacji pliku

Odwoływanie się do plików:

- wprowadzenia otwarcia/zamknięcia pliku - aby uniknąć konieczności odnajdywania pliku w strukturze katalogowej przed każdą operacją we/wy - dane z wpisu katalogowego są przepisywane do tablicy otwartych plików - przy zamknięciu dane o pliku są usuwane z tablicy otwartych plików

- tablica otwartych plików - zawiera informacje o wszystkich otwartych plikach:
  - nazwa
  - prawa
  - daty dostępu
  - wskaźnik do bloku dyskowego

- po otwarciu pliku do programu jest przekazywany indeks do tablicy otwartych plików
- dalsze odwołania do pliku są wykonywane przy użyciu indeksu, a nie nazwy symbolicznej nazwy indeksu:
  - deskryptor pliku (ang. *file descriptor)* - Unix
  - uchwyt plikowy (ang. *file handle)* - Windows NT
  - blok kontrolny pliku (ang. *file control block*) - inne systemy

- w Unix są tablice wielopoziomowe, każdy proces ma tablice otwartych plików, która zawiera wykaz wskaźników (indeksowany deskryptorami) do pozycji w ogólnosystemowej tablicy otwartych plików, w której są informacje o otwartych obiektach (w przypadku plików - wskaźniki do tablicy aktywnych i-węzłów)

Montowanie systemów plików

- przed użyciem system plików musi zostać zamontowany, proces montowania wymaga:
- podania nazwy urządzenia
- określenia miejsca w strukturze plików, gdzie należy podłączyć system plików (punkt montażu, ang. mount point)
- system operacyjny sprawdza, czy urządzenie zawiera właściwy system plików:
- moduł obsługi urządzenia dostaje zlecenie przeczytania katalogu urządzenia, po czym jest sprawdzane, czy posiada on odpowiedni format
- system zaznacza w swej strukturze katalogowej zamontowanie pliku w określonym punkcie montażu

Metody przydziału miejsca na dysku

należy tak przydzielać miejsce, aby:

- obszar dysku był efektywnie wykorzystywany
- dostęp do plików był szybki

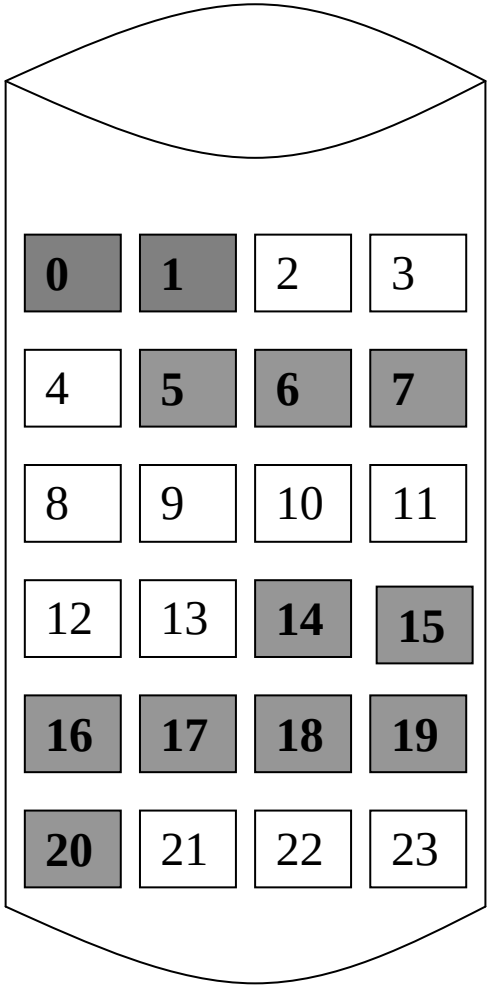
Metody przydziału miejsca na dysku. Przydział ciągły

- Przydział ciągły (ang. *continuous allocation*)
  - plik musi zajmować ciąg kolejnych bloków na dysku
  - adresy dyskowe definiują uporządkowanie liniowe
- liczba operacji przeszukiwania dysku przy dostępie do danych - minimalna
- czas przeszukiwania - minimalny
- dostęp do pliku
  - pamiętany jest adres dyskowy ostatniego bloku pliku, do którego nastąpiło odniesienie i w razie potrzeby czytany jest następny blok
  - można się odnosić bezpośrednio do i-tego bloku pliku
- problem znalezienia miejsca na nowy plik
  - różne implementacje systemów zarządzania wolnymi obszarami
  - może być rozpatrywany jako szczególny przypadek ogólnego problemu dynamicznego przydziału pamięci (ang. *dynamic storage allocation*) - najpowszechniej są stosowane strategie pierwszego i najlepszego dopasowania
  - skąd posiadać wiedzę o rozmiarze pliku podczas jego tworzenia ?
- zewnętrzna/wewnętrzna fragmentacja

Metody przydziału miejsca na dysku. Przydział ciągły

Katalog

Plik	Początek	Długość
dane	0	2
test	5	3
program.c	14	7



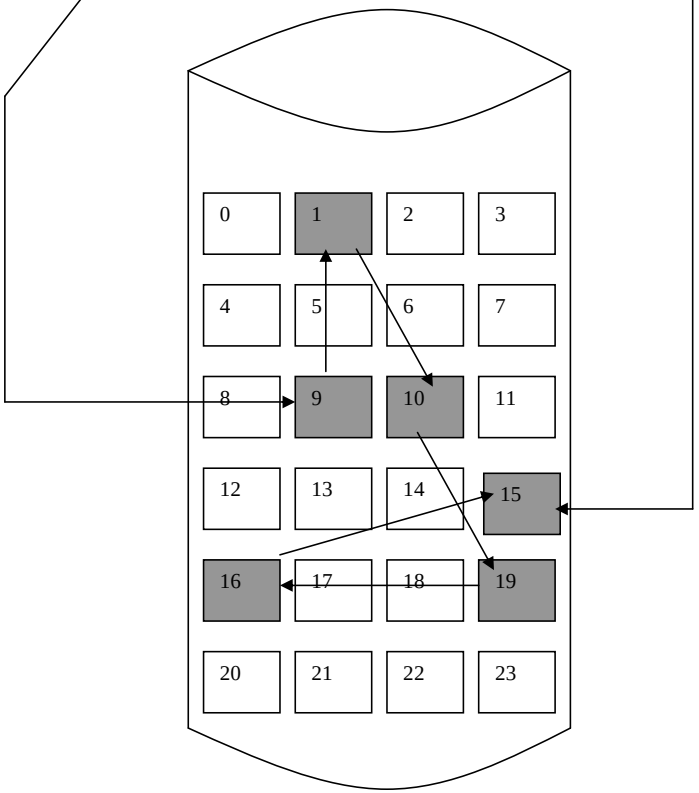
Metody przydziału miejsca na dysku. Przydział listowy (ang. linked allocation)

- każdy plik jest listą powiązanych ze sobą bloków dyskowych, które mogą być dowolnie rozmieszczone na dysku
- katalog zawiera wskaźniki do pierwszego i ostatniego bloku w pliku
- przydział listowy nie ma zewnętrznej fragmentacji, plikowi mogą być przydzielone dowolne bloki z listy wolnych bloków; nie trzeba deklarować rozmiaru pliku w momencie jego tworzenia
- wady przydziału listowego:
  - ograniczenie jego efektywnego zastosowania do plików o dostępie sekwencyjnym (aby znaleźć i-ty blok pliku trzeba zacząć od początku i dojść do i-tego bloku)
  - w każdym bloku trzeba poświęcić pewien obszar na reprezentację wskaźnika co sprawia, że pliki zajmują więcej miejsca -
  - powyższa niedogodność jest usuwana przez grupowanie bloków w tzw. grona/klastry (ang. clusters) i przydzielanie plikom klastrów zamiast bloków, co również polepsza przepustowość, ale powoduje wzrost wewnętrznej fragmentacji
  - niezawodność - błąd wskaźnika mógłby spowodować dowiązanie pliku do innego pliku, lub do listy wolnych obszarów

Metody przydziału miejsca na dysku. Przydział listowy

Katalog

Plik	Początek	Koniec
dane	9	15



Metody przydziału miejsca na dysku. Przydział listowy - użycie tablicy przydziału (rozmieszczenia) plików (ang. file allocation table - FAT)

- stosowano w systemach DOS i OS/2
- początkowa część każdej strefy na dysku jest zarezerwowana na przechowywanie tablicy
  struktura tablicy:
  - tablica ma po jednej pozycji na każdy blok dyskowy i jest indeksowana za pomocą numerów bloków
  - wpis katalogowy zawiera numer pierwszego bloku pliku
  - pozycja w tablicy indeksowana przez numer bloku zawiera numer następnego bloku w pliku (aż do końca pliku)
  - bloki nie używane są wskazywane za liczby 0 umieszczonej na ich pozycji w tablicy
  - przydzielenie nowego bloku do pliku polega na:
  - znalezieniu pierwszej zerowej pozycji w tablicy
    - zastąpieniu poprzedniej wartości symbolizującej koniec pliku przez adres nowego bloku
    - zastąpienie zera adresem końca pliku

Metody przydziału miejsca na dysku. Przydział indeksowy

- wskaźniki są skupione w jednym miejscu - bloku indeksowym (ang. index block)
- każdy plik ma własny blok indeksowy będący tablicą adresów bloków dyskowych
- pozycja o numerze i w bloku indeksowym wskazuje na blok i danego pliku
- katalog zawiera adres bloku indeksowego
- aby przeczytać blok i jest używany wskaźnik z pozycji o numerze i w bloku indeksowym i odnajduje się odpowiedni blok
- przydział indeksowy umożliwia dostęp bezpośredni bez powodowania zewnętrznej fragmentacji
- wada:
  - marnowanie przestrzeni, wskaźniki indeksowe zawierają na ogół więcej miejsca od wskaźników w przydziale listowym
- jaka powinna być wielkość bloku indeksowego ?

Rodzaje realizacji przydziału indeksowego

Schemat listowy

- blok indeksowy zawarty w jednym bloku dyskowy - może być bezpośrednio czytany i zapisywany
- aby umożliwić obsługę dużych plików - połączenie kilku bloków indeksowych

Indeks wielopoziomowy

- wariant reprezentacji listowej
- użycie bloku indeksowego pierwszego poziomu do wskazywania zbioru bloków indeksowych drugiego poziomu, których wskaźniki wskazują na bloki pliku
- można dodać 3 i 4 poziom (uwzględniając rozmiar pliku)

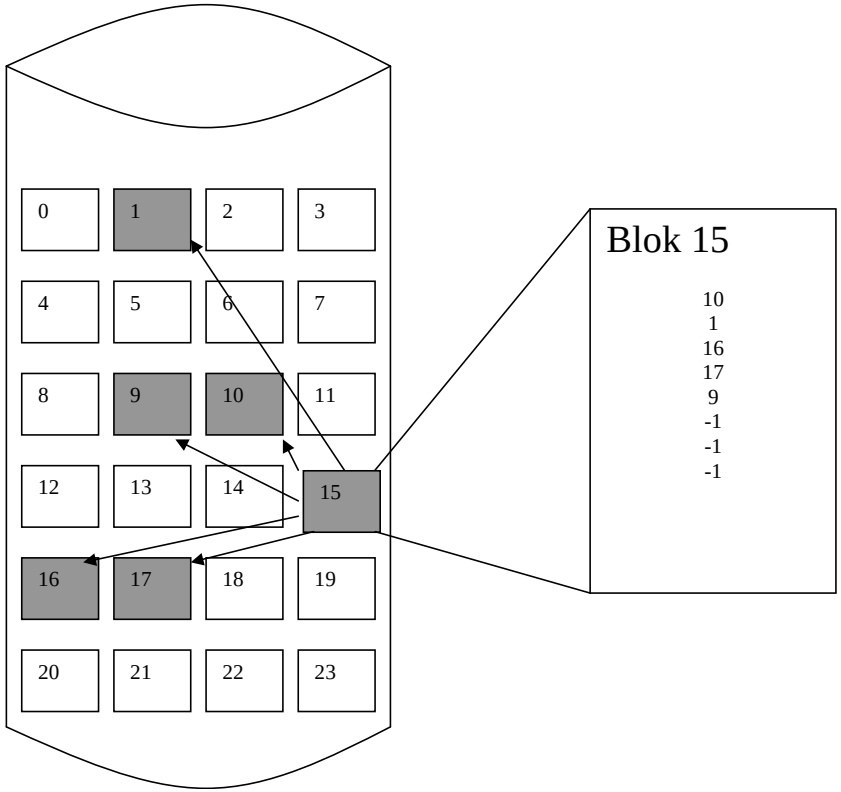
Schemat kombinowany

- stosowany w systemie w systemie Unix
- pierwsze 15 wskaźników bloku indeksowego jest przechowywane w i-węźle pliku
- 12 z nich wskazuje na bloki bezpośrednie (ang. direct block) - dane w małych plikach nie muszą posiadać oddzielnego bloku indeksowego (przy rozmiarze bloku 4KB można zaadresować dane 28KB)
- 3 wskaźniki wskazują bloki pośrednie (ang. indirect blocks)
  - 1 do bloku jednopośredniego (ang. single indirect block), który zawiera adresy bloków z danymi
  - 1 do bloku dwupośredniego - zawiera adres bloku z adresami bloków ze wskaźnikami wskaźniki d do bloków danych
  - 1mógłby zawierać wskaźnik do bloku trójpośredniego

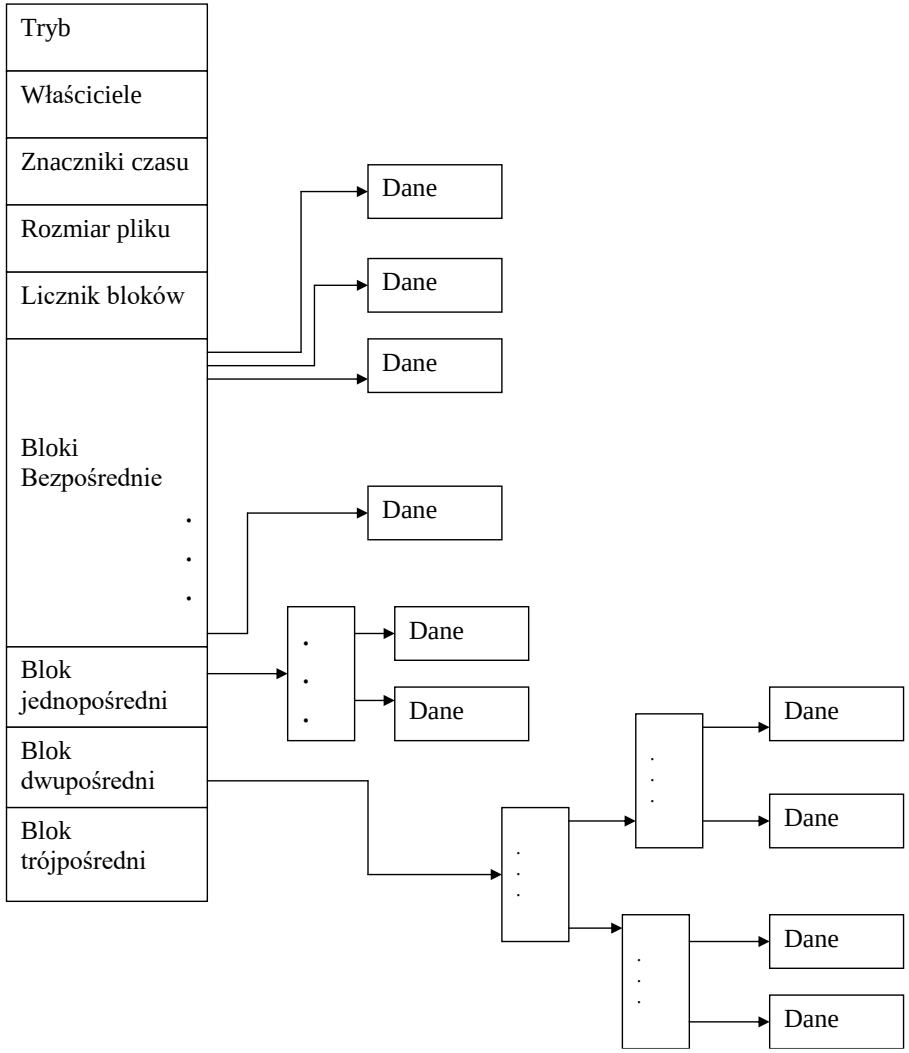
Metody przydziału miejsca na dysku. Przydział indeksowy

Katalog

Plik	Blok indeksowy
dane	16



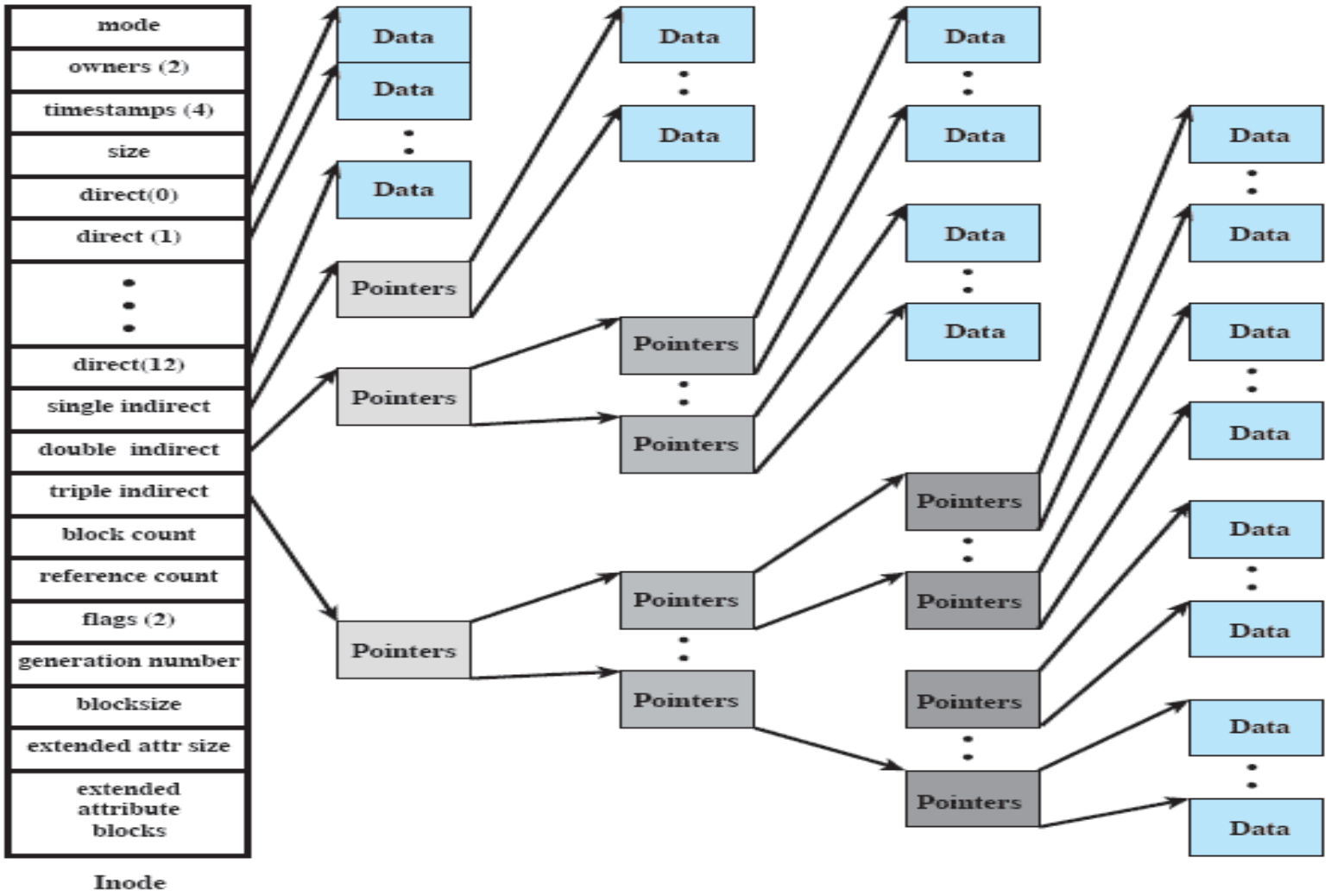
I-węzeł w systemie Unix



Free BSD: I-węzeł zawiera:

- Typ pliku i tryb dostępu
- Identyfikatory dostępu właściciela i grupy
- Czas stworzenia, czasy ostatniego odczytu i zapisu
- Rozmiar pliku
- Sekwencja wskaźników na bloki
- Liczba bloków i liczba wpisów w katalogu
- Rozmiar bloków danych
- Flagi ustawiane przez jądro i użytkownika
- Numer generacyjny przypisany plikowi
- Rozmiar rozszerzonej informacji dotyczącej atrybutów
- Zero lub więcej wpisów z rozszerzonym atrybutami

FreeBSD Inode i struktura pliku



Zarządzanie wolną przestrzenią

- lista wolnych obszarów (ang. free space list) – zawiera informacje o wolnych obszarach (blokach) dyskowych
- modyfikacje listy wolnych obszarów podczas tworzenia/usuwania/… pliku
- sposoby implementacji
  - Wektor bitowy
  - Lista powiązań

Wektor bitowy (ang. bit vector), mapa bitowa (ang. bit map)

- blok reprezentowany przez 1 bit
- blok wolny – bit ma wartość 1, blok przydzielony – bit ma wartość 0

- **Przykład:**
  - 00011100011101
  - wolne bloki: 3,4,5,9,10,11,13

- umożliwia proste i wydajne odnajdywanie pierwszego (lub n kolejnych) wolnych bloków na dysku
- przydatne wsparcie sprzętowe (dotyczące operacji na bitach)
- mało wydajne, jeśli nie można przechowywać całego wektora w pamięci operacyjnej, co w przypadku dużych dysków może wymagać zbyt wiele miejsca

Lista powiązana

- powiązanie ze sobą wszystkich wolnych bloków dyskowych
- przechowywanie wskaźnika do pierwszego wolnego bloku w specjalnym miejscu na dysku i w pamięci
- przeglądanie listy byłoby czasochłonne, ale zazwyczaj nie jest to konieczne
- np. w metodzie FAT zliczanie wolnych bloków jest włączone do struktury danych dotyczących przydziału

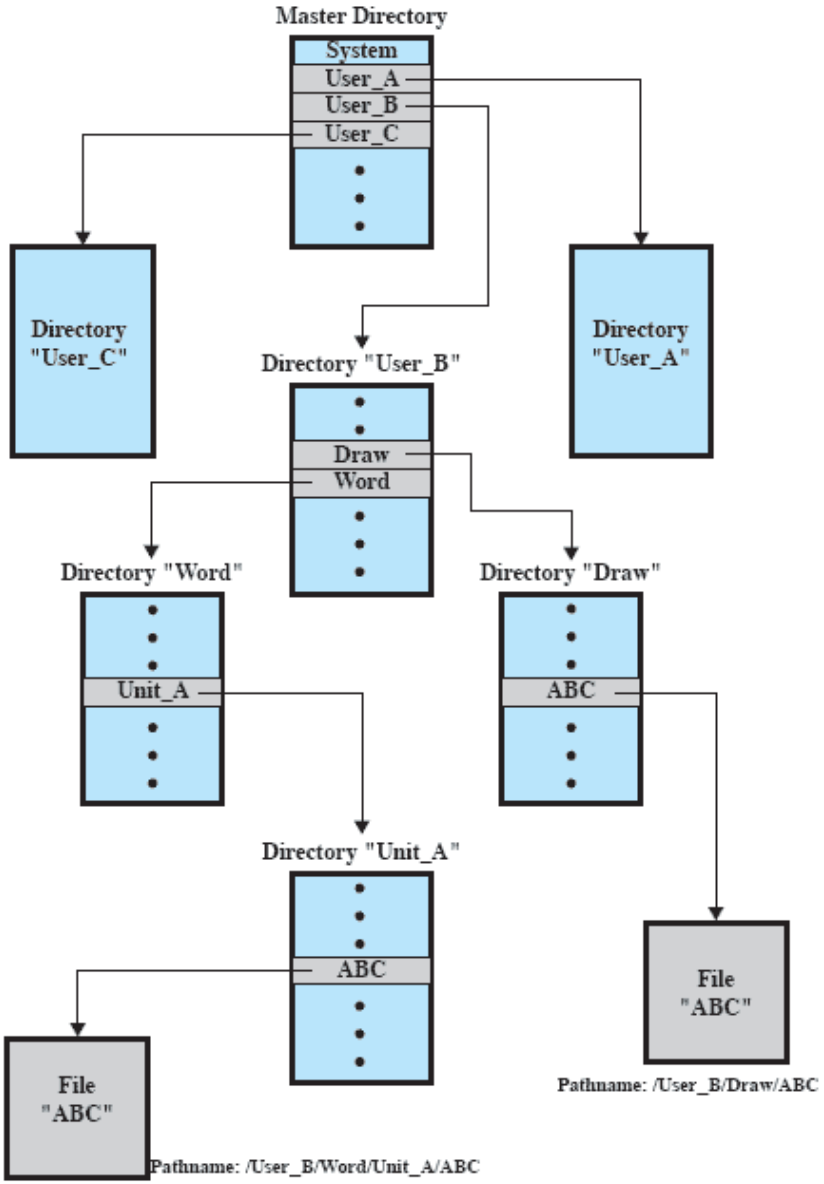
Grupowanie

- modyfikacja podejścia polegającego na zakładaniu listy wolnych obszarów
- przechowywanie adresów n wolnych bloków w pierwszym wolnym bloku
- 0…n-1 - wolne bloki
- ostatni (n-ty) blok zawiera adresy następnych n wolnych bloków itd.
- umożliwia szybkie znajdowanie adresów dużej liczby wolnych bloków

Zliczanie

- zakłada się, że kilka sąsiednich bloków można przydzielać i zwalniać jednocześnie
- przydział bloków algorytmem przydziału ciągłego lub użycie gron
- zamiast wykazu n adresów wolnych bloków wystarczy przechowywać adres pierwszego wolnego bloku i ilość kolejnych wolnych bloków

Organizacja katalogu



Implementacja katalogu

Lista liniowa

- liniowa lista nazw plików ze wskaźnikami do bloków danych
- aby znaleźć konkretną pozycję, trzeba stosować przeszukiwanie liniowe
- metoda łatwa do zaprogramowania, lecz kosztowna czasowo
- podczas utworzenia nowego pliku trzeba przeszukać całą listę, by upewnić się, że nie ma w katalogu pliku o tej samej nazwie
- nowe wpisy są dodawane na końcu listy
- aby odwołać się do pliku, trzeba przeszukać listę
- wykorzystuje się pamięć podręczną, gdzie są przechowywane ostatnie informacje katalogowe
- stosuje się listę uporządkowaną, która umożliwia przeszukiwanie binarne

Implementacja katalogu

Tablica haszująca (ang. hash table)

- wpisy katalogowe są przechowywane także w liście liniowej
- stosuję się dodatkowo haszowaną strukturę danych
- wartość obliczona na podstawie nazwy pliku jest odnoszona do tablicy haszowania, z której pobiera się wskaźnik do nazwy pliku na liście liniowej
- znaczne zmniejszenie czasu przeszukiwania katalogu
- uproszczenie wstawiania i usuwania; trzeba uwzględnić sytuacje kolizyjne

Efektywność

- zależy od doboru algorytmów przydziału miejsca na dysku i algorytmów obsługi katalogów
- i-węzły w systemie Unix są wstępnie rozmieszczane na dysku („pusty” dysk traci część swej przestrzeni na i-węzły), co polepsza wydajność systemu
- aby zmniejszyć wewnętrzną fragmentację są stosowane zmienne rozmiary gron, zależne od wzrostu pliku
- dobór rozmiaru wskaźnika dostępu do pliku
  - np. w systemie DOS początkowo był wskaźnik 12-bitowy (grono 8KB, obszar 32MB),
    - wraz ze wzrostem rozmiarów dysków trzeba było dzielić je na 32MB strefy
    - wpisy w FAT rozszerzono do 16, potem 32 bitów

Wydajność

- Przechowywanie zawartości bloków dyskowych w pamięci:
  - utrzymywanie pamięci podręcznej bloków dyskowych
  - w niektórych systemach (Unix firmy Sun) cała nieużywana pamięć operacyjna jest traktowana jako pula buforów
- różne algorytmy zastępowania bloków w zależności od rodzaju dostępu do pliku, stosowane:
  - wczesne zwalnianie (ang. free-behind) – usuwanie bloku z bufora, gdy zamówiono blok następny
  - czytanie z wyprzedzeniem (ang. read-ahead) – odczytywanie kilku kolejnych bloków
- stosowanie RAM-dysku

Rekonstrukcja. Sprawdzanie spójności

- część informacji katalogowych jest przechowywana w pamięci operacyjnej/podręcznej, w celu przyspieszenia dostępu
- informacje w pamięci operacyjnej są na ogół nowsze od informacji na dysku
- skutki awarii komputera
  - tablica otwartych plików jest z reguły tracona, wraz z nią giną zmiany wprowadzone w katalogach otwartych plików
  - system plików może być niespójny: bieżący stan pewnych plików może nie zgadzać się ze stanem zapisanym w strukturze katalogów
- podczas startu komputera często jest wykonywany jest specjalny program wykrywający i korygujący niespójności na dysku (ang. consistency checker)
  - porównywane są dane w strukturze katalogów z blokami danych na dysku
  - usuwane są niezgodności

Składowanie i odtwarzanie

- Składowanie (ang. backup) i odtwarzanie (ang. restoring) danych
- zapisywanie danych z dysku na innym nośniku
- rejestracja dat ostatniego składowania plików
- składowanie pełne (ang. full backup)
- składowanie przyrostowe (ang. incremental backup

Przykłady systemów plików

NTFS

- Szesnastobitowy system plików FAT miał wiele niedostatków:
  - wewnętrzną fragmentację
  - ograniczenie wielkości do 2GB
  - Brak ochrony dostępów do pliku
- W 32-bitowym systemie plików FAT poradzono sobie z problemami rozmiaru i fragmentacji, ale jego wydajność (i inne cechy) wciąż odstawały od nowoczesnych systemów plików
- System NTFS jest znacznie lepszy - uwzględniono takie cechy jak
  - odtwarzanie danych,
  - bezpieczeństwo,
  - tolerowanie awarii,
  - wielkie pliki i systemy plików,
  - wielość strumieni danych,
  - nazwy standardu UNICODE
  - Pliki rozrzedzone
  - Szyfrowanie
  - Prowadzenie dziennika
  - Kopie-cienie tomów dyskowych
  - Kompresja plików

Budowa wewnętrzna systemu NTFS

- Podstawową jednostką systemu NTFS jest tom (ang. volume)
- Tom jest tworzony przez program administrowania dyskiem systemu Windows XP, u jego podstaw leży logiczny podział dysku
  - Tom może zajmować część dysku lub cały dysk, może też rozciągać się na kilka dysków
  - NTFS nie operuje na poszczególnych sektorach dysku, używa gron (klastrów) jako jednostek przydziału dyskowego
  - Grono (ang. cluster) jest grupą przyległych sektorów, których liczba jest potęgą liczby 2
  - Wielkość grona jest ustalana podczas formatowania systemu NTFS,
    - Dla tomów nie przekraczających 512MB domyślna wielkość grona równa się rozmiarowi sektora
    - tomy o rozmiarach 1GB mają domyślne grona o rozmiarze 1KB
    - tomy o rozmiarach do 2GB mają domyślne grona o rozmiarze 2KB
    - większe tomy mają domyślne grona o rozmiarach 4KB
  - Takie rozmiary gron są znacznie mniejsze niż w 16-bitowym FAT, co zmniejsza wewnętrzną fragmentację
    - Przykład: dysk 1.6GB zawierajacy 16000 plików
    - FAT-16: wielkość grona 32KB, zmarnowane na wewnętrzną fragmentacje: 250MB
    - NTFS: marnuje się tylko 17MB

Budowa wewnętrzna systemu NTFS

- System NTFS używa w charakterze adresów dyskowych logicznych numerów gron (ang. logical cluster numbers - LCN)
  - Są one przypisywane przez ponumerowanie gron od początku do końca
  - Dzięki takiemu schematowi system może wyliczać fizyczną odległość na dysku (w bajtach) mnożąc numer LCN przez wielkość grona
- Plik w NTFS nie jest strumieniem bajtów jak w MS-DOS czy UNIX
  - Plik w NTFS jest obiektem strukturalnym, złożonym z atrybutów o określonych typach
  - Każdy atrybut jest pliku jest niezależnym strumieniem bajtów, który podlega tworzeniu, usuwaniu, czytaniu i zapisywaniu
  - Niektóre typy atrybutów są standardowe dla wszystkich plików np.
    - nazwa/nazwy (jeśli plik ma synonimy jak skrócone nazwy MSDOS)
    - czas utworzenia
    - Deskryptor bezpieczeństwa określający kontrolowanie dostępu
  - Dane użytkownika są pamiętane w **atrybutach danych**

Budowa wewnętrzna systemu NTFS

- Większość tradycyjnych systemów plików ma beznazwowy atrybut danych mieszczący wszystkie dane pliku
- Można jednak tworzyć dodatkowe strumienie danych o jawnych nazwach
- Na przykład, przechowywane w serwerze Windows XP pliki z komputera Macintosh mają odnogę zasobu w postaci nazwanego strumienia danych
- Interfejsy IPropo modelu obiektowego komponentu (ang. Common Object Model - COM) używają nazwanego strumienia danych do pamiętania właściwości zwykłych plików, w tym miniatur (skrawków, ang. thumbnails) obrazów
- Atrybuty można dodawać w zależności od potrzeb, dostęp do nich odbywa się dzięki składani nazwa-pliku.atrybut

MTF – główna tablica plików

- Każdy plik w systemie NTFS jest opisywany przez jeden lub więcej rekordów tablicy przechowywanej w specjalnym pliku nazwanym **główną tablicą plików** (ang. master file table)
  - Rozmiar rekordu jest określony podczas tworzenia systemu plików i waha się między 1 i 4KB
  - Małe atrybuty przechowuje się w samym rekordzie MFT i nazywa atrybutami rezydentymi
  - Wielkie atrybuty, takie jak beznazwowa masa danych – określane mianem nierezydentnych, są przechowywane w jednym lub większej ilości ciągłych **rozszerzeń** na dysku, do których wskaźniki przechowuje się w rekordzie MFT.
  - W przypadku małego pliku w rekordzie MFT może się zmieścić nawet atrybut danych
  - Jeśli plik ma wiele atrybutów lub jest bardzo pofragmentowany i wymaga zapamiętania wielu wskaźników wszystkie jego części, jeden rekord w tablicy MFT może się okazać zbyt mało
  - W tym przypadku plik jest opisany przez rekord o nazwie podstawowy rekord pliku (ang. base file record), który zawiera wskaźniki do rekordów nadmiarowych, przechowujących dodatkowe wskaźniki i atrybuty

NTFS – odsyłacz do pliku

- Każdy plik w tomie systemu NTFS ma niepowtarzalny identyfikator zwany odsyłaczem do pliku (ang. file reference).
- Odsyłacz do pliku jest wielkością 64 bitową, złożoną z 48-bitowego numeru pliku i 16-bitowego numeru kolejnego.
  - Numer pliku jest numerem rekordu (tj. Pozycji w tablicy) w strukturze MFT opisującej plik
  - Numer kolejny jest zwiększany za każdym razem, gdy wpis w tablicy MFT zostaje wtórnie wykorzystany
  - Zwiększanie to umożliwia systemowi NTFS wykonanie wewnętrznej kontroli spójności np. wychwycenie nieaktualnego odwołania do usuniętego pliku po użyciu wpisu MFT na nowy plik

Metadane NTFS

- Wszystkie metadane tomu systemu NTFS są przechowywane w plikach
- Pierwszym z takich plików jest tablica MFT
- Drugi plik, używany do działań naprawczych w przypadku uszkodzenia tablicy MFT, zawiera kopię pierwszych szesnastu pozycji tablicy MFT
- Inne pliki o specjalnym znaczeniu:
  - Plik dziennika – zawiera wszystkie uaktualnienia metadanych systemu plików
  - Plik tomu – zawiera nazwę tomu, dane o wersji systemu NTFS, który sformatował tom oraz bit informujący, czy tom uległ uszkodzeniu i wymaga sprawdzenia spójności
  - Tablica definicji atrybutów – pokazuje, jakie typy atrybutów są używane w tomie i jakie operacje można wykonywać na każdym z nich
  - Katalog główny – katalog z najwyższego poziomu hierarchii systemu plików
  - Plik z mapą bitów – wskazuje, które grona są przydzielone do plików, a które pozostają wolne
  - Plik rozruchowy – zawiera kod początkowy systemu Windows XP i musi znajdować się pod specjalnym adresem na dysku, by prosty program ładujący z pamięci ROM mógł go łatwo zlokalizować; zawiera on także fizyczny adres tablicy MFT
  - Plik złych gron – przechowuje informacje o wszelkich wadliwych obszarach tomu, z informacji tych system NTFS korzysta przy usuwaniu skutków wystąpienia błędu

Usuwanie skutków awarii

- w wielu systemach plików awaria zasilania występująca w złym czasie może uszkodzić struktury danych systemu plików tak poważnie, że zniszczeniu ulega cały tom
- Wiele wersji systemu Unix przechowuje na dysku nadmiarowe metadane i usuwa skutki awarii za pomocą programu fsck, który sprawdza wszystkie struktury danych systemu plików, siłą przywracając im spójny stan
- Odtwarzanie tych struktur często pociąga za sobą usunięcie uszkodzonych plików i zwolnienie gron danych zapisanych przez użytkownika, lecz nie zapamiętanych poprawnie w strukturach metadanych systemu plików
- Sprawdzanie takie może być powolne i powodować utratę znacznych ilości danych

Plik dziennika NTFS

- Wszystkie uaktualnienia struktur danych systemu plików NTfS odbywają się w ramach transakcji
- Zanim nastąpi zmiana struktury danych, transakcja zapisuje w dzienniku wszystkie informacje niezbędne do powtórzenia lub anulowania podjętych czynności
- Po zmianie struktury danych transakcja zapisuje w dzienniku zatwierdzenie, aby zaznaczyć, że zakończyła się pomyślnie
- Po awarii system jest w stanie przywrócić struktury danych systemu plików do stanu spójności przez przetwarzanie zapisów dziennika
  - Najpierw powtarzając czynności transakcji zatwierdzonych
  - Potem usuwając skutki działań transakcji, które nie dobiegły szczęśliwie końca przed awarią
  - Okresowo (zwykle co 5s) do dziennika zapisuje się rekord zwany punktem kontrolnych
    - System nie musi rejestrować rekordów sprzed punktu kontrolnego, by poradzić sobie z awarią
    - Można je usuwać, dzięki czemu plik dziennika nie rozrasta się w sposób nieograniczony

Plik dziennika NTFS (2)

- Schemat taki nie gwarantuje, że zawartość wszystkich plików użytkownika będzie poprawna po awarii, zapewnia tylko, że struktury danych systemu plików (pliki metadanych) będą nieuszkodzone i będa odzwierciedlały pewien spójny stan istniejący przed awarią
- System transakcyjny można rozszerzyć na pliki użytkownika (takie są plany na przyszłość)
- Dziennik jest przechowywany w trzecim pliku metadanych na początku tomu
  - Jest on tworzony ze stałym rozmiarem maksymalnym podczas formatowania systemu plików
  - Dziennik składa się z dwu części:
    - Obszaru rejestrowego (ang. logging area) – będącego cykliczną kolejką wpisów dziennika
    - Obszaru ponownego startu (ang. restart area) – zawierającego informacje kontekstowe, takie jak pozycja w obszarze rejestrowym, do której system NTFS powinien rozpocząć czytanie przy rekonstrukcji
- Czynności uaktualniania dziennika są realizowane przez obsługę dziennika (ang. log-file service) systemu Windows

Bezpieczeństwo

- Bezpieczeństwo tomu NTFS wywodzi się z obiektowego modelu systemu Windows
- Każdy plik systemu NTFS jest powiązany z deskryptorem bezpieczeństwa, który zawiera żeton dostępu właściciela pliku oraz listę kontroli dostępów, wskazującą przywileje dostępu udzielone każdemu z użytkowników, którym pozwolono na korzystanie z pliku
- Podczas normalnego działania system NTFS nie egzekwuje sprawdzania pozwoleń przy przechodzeniu katalogów w nazwach ścieżek plików
- Ze względu na zgodność ze standardem POSIX, sprawdzanie to może być dopuszczone
- Sprawdzanie przejść jest znacznie droższe, gdyż w nowoczenej analizie składni nazw ścieżek nie otwiera się nazw katalogów element po elemencie, lecz stosuje się dopasowanie przedrostków

Linux-systemy plików

- Linux zachowuje model systemu plików standardu UNIX.
- W systemie UNIX plik nie musi być obiektem przechowywanym na dysku lub ładowanym za pomocą sieci z odległego serwera

Linux-wirtualny system plików

- Wirtualny system plików (ang. virtual file system, VFS) systemu Linux jest oparty na zasadach obiektowych
  - ma dwie składowe:
    - zbiór definicji, określający, jak powinien wyglądać obiekt o nazwie plik
    - warstwę oprogramowania do działań na takich obiektach
- Podstawowe typy obiektów zdefiniowane w systemie VFS:
  - struktura obiektu i-węzła
  - struktura obiektu pliku – reprezentujące poszczególne pliki
  - obiekt systemu plików – reprezentujący cały system plików
- Dla każdego z trzech typów obiektów system VFS definiuje zbiór operacji, które muszą być dostępne do wykonania na nich
- Każdy z obiektów wymienionych typów zawiera wskaźnik do tablicy funkcji, która jest wykazem adresów rzeczywistych funkcji implementujących działania na konkretnym obiekcie
  - Umożliwia to warstwie oprogramowania wykonywanie operacji na jednym z obiektów, wywołując odpowiednią funkcję z tablicy funkcji obiektu, bez konieczności uprzedniego dokładnego poznawania, z jakim rodzajem obiektu ma się do czynienia
  - Dla VFS nie jest istotne, czy i-węzeł jest plikiem dostępnym w sieci, na dysku, gniazdem sieciowym czy katalogiem – odpowiednia operacja będzie w odpowiednim miejscu jego tablicy funkcji

Obiekt systemu plików

- Reprezentuje połączony zbiór plików tworzący zamkniętą hierarchię katalogów
- Jądro systemu operacyjnego utrzymuje dla każdego urządzenia dyskowego, zamontowanego jako system plików, jeden obiekt systemu plików, jak również czyni tak dla każdego systemu plików aktualnie podłączonego przez sieć
- Podstawowym obowiązkiem obiektu systemu plików jest udostępnianie i-węzłów
- System VFS identyfikuje każdy i-węzeł za pomocą niepowtarzalnej pary (system plików, numer i-węzła) i odnajduje i-węzeł odpowiadający danemu numerowi i-węzła, kierując do obiektu systemu plików prośbę o zwrócenie i-węzła o podanym numerze

Obiekt i-węzła i obiekt pliku

- Obiekty i-węzłów i obiekty plików są mechanizmami dostępu do plików
- Obiekt i-węzła reprezentuje plik jako całość, a obiekt pliku reprezentuje plik dostępu do danych wewnątrz pliku
- Proces nie może sięgnąć do zawartości danych i-węzła bez wcześniejszego uzyskania obiektu pliku wskazującego na i-węzeł
- Obiekt pliku
  - Obiekt pliku przechowuje informacje o aktualnym miejscu czytania lub zapisywania pliku przez proces, co umożliwia sekwencyjne operacje we/wy na pliku
  - W obiekcie pliku pamięta się również, czy proces prosił o prawo zapisywania pliku przy jego otwieraniu
  - Przechowuje się informacje o czynnościach procesu w razie konieczności wykonywania adaptacyjnego czytania z wyprzedzeniem tj. sprowadzania danych pliku do pamięci zawczasu, przed zapotrzebowaniem ich przez proces, w celu poprawienia wydajności
  - Obiekty plików należą zazwyczaj do pojedynczych procesów, inna sytuacja jest w przypadku obiektów i-węzłów
- Obiekt i-węzła
  - Nawet jeśli plik nie jest już używany przez żaden proces, system VFS może wciąż przechowywać jego i-węzeł w pamięci podręcznej w celu zwiększenia wydajności w przypadku ponownego użycia pliku w najbliższej przyszłości
  - Wszystkie przechowywane podręcznie dane plikowe są łączone w listę w obiekcie i-węzła pliku
  - I-węzeł utrzymuje też standardowe informacje o pliku, takie jak nazwa właściciela, rozmiar pliku i czas jego ostatniej modyfikacji

Pliki katalogowe

- Pliki katalogowe są obsługiwane nieco inaczej niż pozostałe pliki
- Programowy interfejs systemu Unix określa pewną liczbę operacji na katalogach, jak tworzenie, usuwanie i przemianowywanie pliku w katalogu.
- W odróżnieniu od czytania i zapisywania danych, kiedy to plik musi najpierw być otworzony, systemowe wywołania operacji na katalogach nie wymagają od użytkownika otwierania interesujących go plików
  - Dlatego system VFS definiuje operacje katalogowe w obiekcie i-węzła, a nie w obiekcie pliku

Linux: Ext2fs

- Podstawowy system plików Linuxa przez długi czas (obecny Ext3fs stanowi uzupełnienie ext2fs o dodatkową funkcjonalność związaną z dziennikami)
- Odnajdywanie bloków danych należących do danego pliku dzięki przechowywaniu wskaźników bloków danych w blokach pośrednich, spiętrzanych w całym systemie aż do trzeciego poziomu
- Pliki katalogowe są pamiętane na dysku tak jak zwykłe pliki
  - Każdy blok w pliku katalogowym składa się z powiązanej listy wpisów, z których każdy zawiera długość wpisu, nazwę pliku i numer i-węzła, do którego dany wpis się odwołuje
- Blok standardowy w systemie ext2fs ma wielkość 1KB, choć stosuje się także bloki o wielkości 2KB i 4KB
- Aby zapewnić wysoką wydajność, system operacyjny powinien starać się wykonywać operacje we/wy wielkimi porcjami, gdy tylko jest taka możliwość, łącząc zamówienia na przyległe jednostki we/wy w grona
  - Operowanie gronami zmniejsza koszt przypadający na jedno zamówienie, spowodowany pracą modułów sterujacych urządzeń, dysków itd.
  - Zamówienie wielkości 1KB jest za małe, aby utrzymywać dobrą wydajnosć, toteż system ext2fs stosuje politykę przydziału zmierzającą do umieszczania logicznie sąsiadujących bloków pliku w fizycznie przylegających blokach dyskowych, dzięki czemu może on w jednej operacji przedkładać zamówienia we-wy na kilka bloków

Linux: Ext2fs: Zasady dokonywania przydziałów

- Zasady dokonywania przydziałów w extfs można podzielić na dwie części
  - System plików extfs jest podzielony na wiele **grup bloków**
  - Przydzielając miejsce dla pliku, system ext2fs musi najpierw wybrać dla niego grupę bloków
  - System usiłuje przydzielić blokom danych tę grupę bloków, do której przydzielono i-węzeł pliku.
  - Do przydzielania i-węzłów plików nie będących katalogami system wybiera tę grupę bloków, w której mieści się macierzysty katalog pliku.
  - Pliki katalogowe nie są przechowywane razem, lecz są rozproszone w dostępnych grupach bloków
    - Zasady te przyjęto dlatego, żeby powiązane ze sobą informacje trzymać w tej samej grupie bloków, oraz by rozłożyć obciążenie dysku między jego grupy bloków w celu zmniejszenia fragmentacji poszczególnych obszarów dyskowych.
  - Wewnątrz grupy bloków system próbuje w miarę możliwości dokonywać przydziałów ciągłych fizycznie, dążąc do zmniejszenia fragmentacji
    - Utrzymywana jest mapa bitowa wszystkich wolnych bloków w grupie
    - Podczas przydzielenia pierwszego bloku nowego pliku system rozpoczyna szukanie wolnego bloku od początku grupy bloków
    - Przy rozszerzaniu pliku poszukiwania są kontynuowane od bloku przydzielonego plikowi ostatnio

Linux: Ext2fs: Zasady dokonywania przydziałów

- Szukanie bloków do przydziału
  - (1) System szuka wolnego całego wolnego bajta w mapie bitowej → ma to na celu przydzielanie, w miarę możliwości, dysku porcjami co najmniej ośmioblokowymi
  - (2) Jeśli (1) się nie powiedzie, następuje szukanie pojedynczego wolnego bitu.