AGH

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Systemy operacyjne
Wykład 08

Wersja 2026

dr inż. Marek Wilkus **http://home.agh.edu.pl/~mwilkus**
Wydział Inżynierii Metali i Informatyki Przemysłowej
AGH Kraków

Na podstawie programu opracowanego przez dr inż. Krzysztofa Wilka

AGH

System plików

AGH

System plików

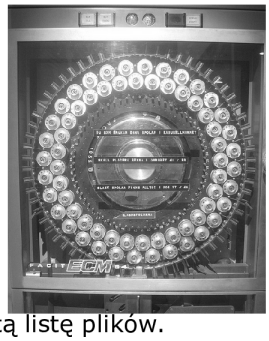
- **Plik** jest logiczną jednostką magazynowania informacji w pamięci nieulotnej.
- **Plik** jest nazwanym zbiorem powiązanych ze sobą informacji zapisanym w pamięci pomocniczej.
- **Plik** jest ciągiem bitów, bajtów, wierszy lub rekordów.

- Atrybuty pliku:
  - Nazwa (zgodna z regułami dla danego systemu operacyjnego),
  - Typ (jeżeli system tego wymaga),
  - Położenie (wskaźnik do urządzenia i położenie na tym urządzeniu),
  - Rozmiar (w bajtach, słowach lub blokach),
  - Ochrona (prawa dostępu),
  - Czas, data, identyfikator właściciela,
  - Ewentualnie inne metadane (fork zasobów (Apple), rekord metadanych (BeFS)).

AGH

Rys historyczny

- Komputery bez dysków, oparte o taśmy: Urządzenie jest plikiem.
- Później: Na urządzeniu może być kilka plików, ale i tak chodzi o zmianę taśm. Ręczną lub automatyczną (projekt Facit Carousel),
- Później: Na jednej taśmie może być kilka plików. Konieczność zwiększenia trwałości nośników taśmowych (próżnia i.t.p. rozwiązania),
- Później: Taśmy indeksowane.



- Systemy dyskowe: Dysk posiada prostą listę plików.
- Później: systemy plików.

AGH

Rys historyczny

- Lata 1970-80 - mikrokomputery:
  - Urządzenie (np. Magnetofon, drukarka, napęd - jest plikiem)
  - Później: Na urządzeniu o dostępie swobodnym może być pewna ilość osobnych plików. Poprzedni punkt pozostaje w mocy odnośnie napędów taśmowych i niektórych urządzeń I/O.
  - W takim przypadku systemem plików steruje oprogramowanie układowe napędu dyskowego - mało wydajne, ale jedyne możliwe bez ładowania dodatkowego sterownika z taśmy.
  - Maszyny z systemem CP/M: Stacja dysków staje się komputerem. Mikrokomputer staje się terminalem.

AGH

Rys historyczny

- Uniksowe mainframe: Hierarchiczny system plików.
- IBM PC (DOS 1.x) - prosta lista plików na dyskietce.
- IBM PC (DOS 2.x) - katalogi. Dyski twarde do 20-30MB (lub więcej ze sterownikami), kwestia systemu plików na większych dyskach nie jest ustandaryzowana.
- DOS 3.x-6.x - FAT - pełna obsługa zagnieżdżonych katalogów (z pewnymi limitami) i wielu partycji.
- Windows 95 - VFAT i 95 OSR2 OSR2 - FAT32 - większe dyski, rozszerzenie FATa tak, by plik miał nazwę dłuższą niż 8+3.
- Windows NT - NTFS - w systemie plików pojawiają się strumienie, rozszerzone atrybuty, sparse files, obiekty (OFS - nie dostał się do końcowej wersji WinNT).

Operacje plikowe

- **Tworzenie pliku**:
  - znalezienie miejsca w systemie plików
  - wpis do katalogu
- **Zapisywanie pliku** - podaje się nazwę (identyfikator) pliku i informację do zapisania. Istotne jest miejsce od którego piszemy (wskaźnik położenia).
- **Czytanie pliku** - podaje się nazwę pliku i bufor w pamięci. Można wykorzystać ten sam wskaźnik położenia.
- **Zmiana pozycji w pliku** - modyfikacja wskaźnika położenia.
- **Usuwanie pliku** - zwalnia się przestrzeń zajmowaną przez plik i likwiduje się wpis katalogowy.
- **Skracanie pliku** - likwidowanie części albo całej zawartości pliku bez kasowania jego nazwy i atrybutów.

Inne operacje

- **Dopisywanie** - dopisywanie nowych informacji na końcu istniejącego pliku.
- **Przemianowanie pliku** - zmiana nazwy pliku, często tą samą komendą wykonuje się przesuwanie pliku, czyli zmianę jego położenia - do innego katalogu, na inny dysk.
- **Otwieranie pliku** - stosowane w wielu systemach w celu uniknięcia wielokrotnego czytania informacji o pliku - dane z katalogu kopiowane są do tablicy otwartych plików.
- **Zamykanie pliku** - kiedy plik przestaje być potrzebny, usuwa się wpis z tablicy otwartych plików.

- **Otwieranie i zamykanie plików w systemach wielożytkownikowych musi uwzględniać równoczesne korzystanie z pliku przez kilka procesów!**

Typy plików

- System rozpoznaje typy plików poprzez:
  - Rozszerzenia - w MSDOS niektóre typy plików określane przez rozszerzenia nazwy(*.com, *.exe…),
  - Liczby magiczne - oraz typowe fragmenty początku pliku - identyfikacja w systemie Unix (komenda file, plik /etc/magic),
  - Atrybut twórcy (w Mac OS) - czyli nazwę programu, przy pomocy którego utworzono plik.

Typy dostępu do plików

- **Dostęp sekwencyjny** - informacje w pliku są przetwarzane kolejno, rekord po rekordzie,
- **Dostęp bezpośredni** - umożliwia czytanie z zapisywanie bloków w dowolnej kolejności. Rekordy muszą być stałej długości. Używany jest tam, gdzie potrzebny jest szybki dostęp do wielkich ilości informacji, np w bazach danych.
- **Dostęp indeksowy** (plik indeksowy w pamięci, lub na dysku)

Katalogi

- **Jednopoziomowy** - ograniczeniem jest konieczność spełnienia warunku niepowtarzalności nazw.
- **Dwupoziomowy** - każdy użytkownik ma własny katalog macierzysty, a w nim pliki.
- **Wielopoziomowe drzewiaste**.
- **Acykliczne grafy** - do pliku można dojść wieloma drogami.

Ochrona plików

- Można kontrolować wiele operacji:
  - **czytanie** pliku
  - **pisanie** do pliku, lub zapisywanie go na nowo
  - **wykonywanie** - załadowanie pliku do pamięci i wykonanie go
  - **dopisywanie** danych na końcu pliku
  - **usuwanie** pliku i zwalnianie obszaru przez niego zajętego
  - **opisywanie** - wyprowadzenie nazwy i atrybutów pliku

- Klasy użytkowników pliku:
  - **właściciel** - użytkownik, który utworzył dany plik
  - **grupa** użytkowników, którzy wspólnie korzystają z pliku i potrzebują podobnego zakresu dostępu
  - **wszyscy** inni.

Ochrona w Windows linii NT

- Możliwości:
  - Pełna kontrola
  - Odpowiednik +x na katalogu
  - Odpowiednik +r na katalogu
  - Odczyt/zapis atrybutów
  - Odczyt/zapis alternatywnych strumieni
  - Tworzenie/usuwanie plików
  - Tworzenie/usuwanie folderów
  - Usuwanie obiektu
  - Modyfikacja obiektu
- Każdy użytkownik może zostać tak zapisany do każdego obiektu systemu plików.
- Problem: Wydajność systemu: Spada przy >500 użytkownikach
  - W Uniksie normalne, w Windows sytuacja bardzo rzadka.

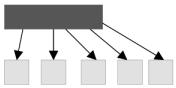
Przydział miejsca na dysku

- **Przydział ciągły** - każdy plik zajmuje ciąg kolejnych bloków na dysku.
  - zalety: minimalna liczba operacji przeszukiwania dysku, łatwość implementacji dostępu sekwencyjnego i bezpośredniego.
  - wady: trudności ze znalezieniem wolnego miejsca (fragmentacja zewnętrzna),

- **przydział listowy** - istnieje lista powiązanych ze sobą bloków dyskowych, stanowiących dany plik. Bloki te mogą się znajdować w dowolnym miejscu na dysku.
  - zalety:brak fragmentacji zewnętrznej, nie trzeba deklarować długości pliku (plik może rosnąć, dopóki są wolne bloki)
  - wady: trudność w implementacji dostępu bezpośredniego, zajęcie sporej przestrzeni przez wskaźniki, w przypadku błędu jednego wskaźnika można wejść w obszar innego pliku.

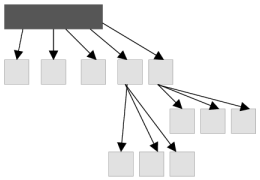
Przydział miejsca na dysku

- **Przydział indeksowy** - podobny jak przydział listowy, ale wskaźniki umieszczone w jednym miejscu - w tablicy indeksów
  - zalety: jak w przydziale listowym,
  - wady: wskaźniki bloku indeksowego zajmują zazwyczaj więcej miejsca niż wskaźniki przy przydziale listowym



Dla dużych plików...

- Schemat listowy - jeśli lista bloków jest dłuższa niż blok indeksowy, na ostatniej pozycji w bloku indeksowym podaje się adres bloku kontynuacji,
- Indeks wielopoziomowy - pozycje bloku indeksowego wskazują na bloki indeksowe poziomu drugiego.
- Schemat mieszany -pierwsze kilka, kilkanaście pozycji wskazuje bezpośrednio na bloki, a następne 2-3 na indeksy poziomu drugiego w indeksowaniu 2,3,4-poziomowym. Schemat ten zastosowany w systemie UNIX.



Przydział miejsca a wydajność

- Przykład złego doboru metody przydziału: przy przydziale listowym i dostępie bezpośrednim, dostęp do bloku n wymaga n dostępów do dysku.
- Metody poprawy wydajności
  - deklaracja typu dostępu przy tworzeniu pliku - jeśli dostęp ma być bezpośredni, stosuje się przydział ciągły (wymaga podania wielkości pliku przy tworzeniu)

  Jeśli dostęp ma być sekwencyjny, stosuje się przydział listowy

  System musi mieć zaimplementowane obie metody przydziału
  - Typ przydziału zależny od wielkości pliku - dla małych, kilku-blokowych plików przydział ciągły, dla dużych plików przydział np indeksowy.
  - Stosowanie klastrów (gron) i różnych wielkościach i stosowanie możliwie największych (np 64 kB) dla dużych plików. Uzupełnianie do końca pliku małymi klastrami.

Zarządzanie wolną przestrzenią

- **mapa bitowa** - w wektorze bitowym każdy wolny blok jest reprezentowany przez 1 a zajęty - przez 0. łatwość znalezienia wolnego bloku istnieje dzięki rozkazom procesora pokazującym pozycję pierwszego niezerowego bitu w słowie. Metoda ta może być stosowana dla małych dysków.
- **lista powiązana** - w pamięci przechowuje się położenie pierwszego wolnego bloku, a w nim - położenie następnego itd. Metoda mało wydajna - aby przejrzeć listę wolnych bloków, należy wszystkie przeczytać.
- **grupowanie** - w pierwszym wolnym bloku przechowywana jest lista n wolnych bloków. W n-tym wolnym bloku znajduje się lista następnych n bloków itd.
- **zliczanie** - przechowuje się adres pierwszego wolnego bloku i liczbę n następujących po nim wolnych bloków. I tak dla każdej grupy. N jest zazwyczaj > 1

Poprawa wydajności systemów dyskowych

- **pamięć podręczna** - przechowywanie całych ścieżek dysku w pamięci - prawdopodobnie będą z nich w niedługim czasie czytane dane. Wykorzystana do tego celu specjalna pamięć, lub nieużywana pamięć główna.
- **wczesne zwalnianie** - usuwanie bloku z bufora natychmiast, gdy pojawia się zamówienia na następny (oszczędza pamięć)
- **czytanie z wyprzedzeniem** - z zamówionym blokiem czyta się kilka następnych, gdyż prawdopodobnie zaraz będą potrzebne.
- **RAM-dysk** - wszystkie operacje dyskowe przeprowadza się w pamięci. Zawartość RAM-dysku jest pod kontrolą użytkownika. Wada - zawartość ginie po wyłączeniu zasilania, awarii.
- **Opóźniony zapis** – Zapis z pamięci podręcznej następuje później, preferuje się nawet zapis na dysk dopiero wtedy gdy potrzeba te dane odczytać. Dzięki temu opóźnia się zbędne zapisy danych tymczasowych

Czynności operatorskie

- **Sprawdzanie spójności** - po awarii systemu, czy np po wyłączeniu „z kontaktu". Program chkdsk (Windows), scandisk (DOS), fsck (UNIX). Zazwyczaj uruchamiają się automatycznie (znacznik „czystości" systemu plików)
- **Składowanie i odtwarzanie** - robienie kopii systemu plików na innym nośniku i odtwarzanie po awarii. Kopia zapasowa, Norton Ghost (DOS, Windows), tar, backup, restore (Unix).
- **Składowanie przyrostowe** - kopia całego systemu raz, a potem zapisywanie tylko zmienionych plików.

  Kopie „wieczyste" - co jakiś czas, taśmy pozostają w archiwum „na zawsze".

System Plików - DOS FAT

- **Katalog** - zawiera: nazwy plików (8 znaków), rozszerzenie (3 znaki), długość pliku, atrybuty (h s r a), datę utworzenia pliku, wskazanie pozycji FAT
- **FAT** (file allocation table) - tablica, której elementy odpowiadają kolejnym jednostkom alokacji (sektorom, blokom, klastrom). FAT jest umieszczony na początku dysku, w dwóch kopiach.

Przykład użycia: Plik Z1 zajmuje bloki: 7,8,11,3, Z2 zajmuje blok 4, a Z3 jest w blokach: 1,2,5,6,9

```
Nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 15 17 18 19 20
    02 05 ff ff 06 09 08 11 ff 00 03 00 00 00 00 00 00 00 00 00
```

FAT 12 - każda pozycja miała 12 bitów, czyli mogła zawierać wartość 0-4096, co przy klasterach 8 kB dawało 32 MB

Kolejne wersje systemu - FAT 16 (do 2 GB) i FAT 32.
Bariera - czas przeszukiwania tablicy FAT.

System Plików – NTFS (Windows)

- **Wolumin** (volume)- podstawowa jednostka dyskowa - może to być część dysku, cały dysk lub kilka dysków razem
- **Klaster** (grono, cluster) - podstawowa jednostka przydziału, jest to grupa sąsiadujących sektorów. Są znacznie mniejsze niż w systemach z FAT - 4 kB dla dużych dysków. Jako adresy dyskowe używane są logiczne numery klastrów.
- **Plik** jest obiektem strukturalnym złożonym z atrybutów. Atrybuty pliku są strumieniami bitów. Jednym z atrybutów są też dane pliku.
- **Główna tablica plików** (MFT) - przechowuje opisy plików, zawarte w jednym lub kilku rekordach dla każdego pliku.
- **Odsyłacz do pliku** - niepowtarzalny identyfikator pliku, składa się z 48-bitowego numeru pliku (pozycja w MFT) i 12-bitowego numeru kolejnego.

NTFS – inne pliki systemowe

- **Kopia MFT** - kopia pierwszych 16 pozycji MFT - do działań naprawczych,
- **Plik dziennika** - zawiera wszystkie zmiany danych w systemie plików,
- **Plik woluminu** - dane woluminu, dane o wersji NTFS, który go sformatował, bit informujący o konieczności chkdsk,
- **Tablica definicji atrybutów** - typy atrybutów i dopuszczalne dla nich operacje,
- **Katalog główny**
- **Plik mapy bitów** - wskazuje zajęte i wolne klastry,
- **Plik rozruchowy** - kod początkowy NT,
- **Plik uszkodzonych klastrów**,
- **Usuwanie skutków awarii** jest stosunkowo proste, gdyż operacje dyskowe odbywają się na zasadzie **transakcji**.

NTFS – nazwy plików systemowych

System plików	Nazwa Pliku	Rekord MFT	Funkcje
Master file table	$Mft	0	Zawiera podstawowe informacje na temat plików i folderów w systemie plików.
Master file table 2	$MftMirr	1	Kopia 4 pierwszych rekordów MFT.
Log file	$LogFile	2	Zapamiętuje ostatnie transakcje wykonane na systemie plików.
Volume	$Volume	3	Zawiera informacje nt. woluminu: nazwa, wersja.
Attribute definitions	$AttrDef	4	Tablica atrybutów, ich nazw oraz opisu.
Root file name index	$	5	Katalog główny.

NTFS – Nazwy plików systemowych

System plików	Nazwa Pliku	Rekord MFT	Funkcje
Cluster bitmap	$Bitmap	6	Przechowuje informacje o używanych klastrach.
Boot sector	$Boot	7	Boot sector
Bad cluster file	$BadClus	8	Zawiera informacje o niedziałających klastrach.
Security file	$Secure	9	Deskryptory praw dostępu.
Upcase table	$Upcase	10	Zamienia małe litery na wielkie, zgodnie z Unicode.
NTFS extension file	$Extend	11	Używana w wielu fuknejach NTFS, np.: quota, reparse point, object ID.

NTFS – system plików Windows

- Każda operacja jest wykonywana na zasadzie transakcji.
- Najpierw zmiany dokonywane na metadanych zostają zapisane w dzienniku, dopiero potem faktyczna operacja ma miejsce i jest zatwierdzana.
- W przypadku awarii systemu, uruchamiane jest odzyskiwanie systemu plików.
- Przeglądany jest dziennik i wszystkie niedokończone transakcje są usuwane bądź realizowane zgodnie z zapamiętanymi tam informacjami.
- Takie rozwiązanie gwarantuje stabilność systemu plików, jednak czasami można utracić część informacji, które były modyfikowane w trakcie awarii, bądź znajdowały się na zepsutym klastrze.

NTFS

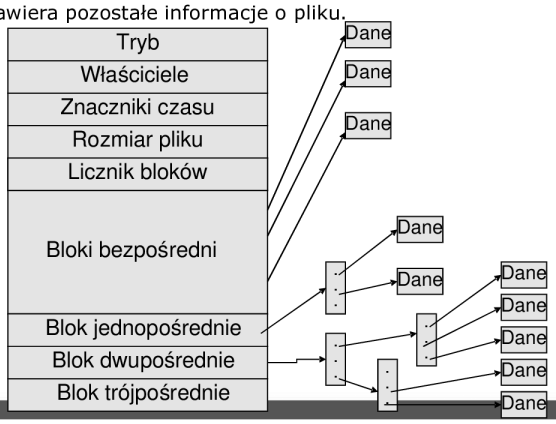
- W przypadku gdy system plików odkryje niedziałające klastry na dysku, automatycznie oznacza je jako zepsute i podstawia na jego miejsce nieużywany klaster.
- W przypadku odwołania do adresu podmienionego klastra, NTFS realizuje żądanie na nowym zmapowanym klastrze.
- W przypadku odczytu z błędnego klastra traci się informacje na nim zawarta, jednak w przypadku zapisu, użytkownik nie dowie się nawet, że wystąpił po drodze jakiś błąd.
- Powtarzające się awarie klastrów są wskazówką do wymiany nośnika na nowy.

ExFAT - FAT z przyspieszeniem

- Parsowanie tablicy FAT zajmuje mnóstwo czasu i pamięci.
- Przy zwiększeniu wartości do współczesnych liczb (np. 64-bitowe wpisy) jest jeszcze gorzej.
- Jak to usprawnić?
  - Haszowanie nazw plików - hash umożliwia szybkie wyszukiwanie na liście plików w katalogu (tu nie ma drzew jak w NTFS).
  - Zaznaczanie miejsca, które dopiero później będzie wykorzystane.
  - Możliwość zaalokowania miejsca w ciągłych blokach „a'la extent" - nie adresowanych później jak extent.
  - Pozyskiwanie informacji o dalszych blokach pozyskując jedynie fragment tablicy powiązany z hashem - mniej do poszukiwania.
- System ten jest popularny na dyskach przenośnych w systemach Windows, choć istnieje również sterownik pod Linuksa.

Systemy plików UNIX (klasyczny FS)

- Katalog - zawiera: nazwy plików (w starszych do 256 znaków), wskazanie Inode (węzła). Za wyjątkiem znacznika, nie różni się od pliku.
- I-node - zawiera pozostałe informacje o pliku.



Zawartość I-węzła

Typ	Pole	Opis
__u16	i_mode	typ pliku (dowiązanie symboliczne, zwykły plik, katalog, urządzenie znakowe, urządzenie blokowe, gniazdo, kolejka FIFO) i prawa dostępu
__u16	i_uid	Identyfikator właściciela pliku
__u32	i_size	Długość pliku w bajtach
__u32	i_atime	Czas ostatniego dostępu (w sekundach od epoki Uniksa)
__u32	i_ctime	Czas ostatniej zmiany i-węzła (jw.)
__u32	i_mtime	Czas ostatniej zmiany zawartości pliku (jw.)
__u32	i_dtime	Czas usunięcia pliku (jw.)

Zawartość I-węzła (2)

Typ	Pole	Opis
__u16	i_gid	Identyfikator grupy
__u16	i_links_count	Licznik twardych dowiązań do pliku
__u32	i_blocks	Liczba bloków danych pliku (po 512 bajtów)
_u32	i_flags	Flagi pliku ("tylko dodawanie (append only)", "nie można zmieniać (immmutable)", i inne)
union	osd1	Specyficzne informacje systemu operacyjnego
__u32	i_block	Wskaźniki do bloków danych (zwykle 15, pierwszych 12 to wskaźniki bezpośrednie, jeden pośredni, jeden podwójnie pośredni, jeden potrójnie pośredni)

Zawartość I-węzła (3)

Typ	Pole	Opis
__u32	i_version	Wersja pliku (dla NFS)
__u32	i_file_acl	Lista kontroli dostępu do pliku (ACL)
__u32	i_dir_acl	Lista kontroli dostępu katalogu
__u32	i_faddr	Adres fragmentu
union	osd2	Specyficzne informacje systemu operacyjnego

System plików UNIX

- **Superblok** - zawiera statyczne parametry systemu plików: całkowity rozmiar systemu plików, rozmiary pełnych bloków danych i ich fragmentów, dane dotyczące przydziału miejsca.
- **Grupa cylindrów** - dla zminimalizowania ruchu głowic dysku, cylindry dysku (ta sama ścieżka na wszystkich głowicach) pogrupowano po kilka, zapisując w każdej grupie:
  - superblok,
  - blok cylindra (zawierający mapę wolnych bloków, mapę wolnych i-węzłów, dane statystyczne do strategii przydziału),
  - I-węzły,
  - bloki danych (do końca grupy cylindrów).

Typowe problemy

- Klasyczny UNIX/Solaris: Ciągły FSCK i tracenie danych przy starcie na sprawnym dysku:
  - Administrator partycjonował dyski bez dokumentacji i „zaorał” superblok początkiem partycji.
- Linux: FSCK wskazujący ciągłe usuwanie nieistniejących porzuconych i-nodeów:
  - Synchronizacja przez sieciowy system plików z nieprawidłowo ustawionymi metadanymi (pliki z przyszłości).
- Implementacja FAT: Katalog urywa się w połowie:
  - Pomimo braku w większości dokumentacji: Katalog też jest zapisywany jak plik i trzeba parsować FATa.

System plików Ext4

- Domyślny system plików większości dystrybucji systemu Linux.
- Następca Ext3, Ext2, Ext, Minix FS…
- Możliwości:
  - Urządzenia do 1EB (Eksabajt),
  - Nielimitowana głębokość ścieżek katalogów,
  - Dziennik transakcji z obsługą sum kontrolnych,
  - API szyfrowania w trakcie operacji dyskowych,
  - Opóźnienie czyszczenia i-nodeów do momentu gdy zachodzi potrzeba ich zapisu.
  - Opóźnienie problemu roku 2038 o dwa bity (do roku 2446).
  - Przydziały miejsca na poziomie systemu plików, bardziej elastyczne niż partycjonowanie.

Ext4 - przydział miejsca

- Przestrzeń przydzielana jest nie w blokach, lecz w strukturach definiujących zakresy (extents).
- W metadanych jednego pliku jest miejsce na 4 extenty, maksymalny rozmiar 128MB * 4 = 512MB
- Większy plik - extenty są przechowywane w drzewiastej strukturze umożliwiającej wydajne poszukiwanie po offsecie. (najczęściej do pliku dostęp dokonujemy albo odczytując od początku do końca, albo od jakiegoś przesunięcia)

Ext4 - walka z fragmentacją

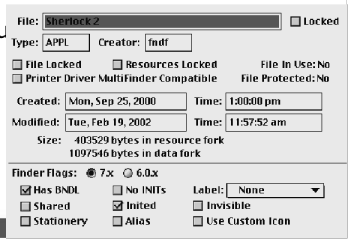
- Allocate-on-flush - Alokacja bloków dopiero w momencie zapisu bufora na dysku - zmniejsza fragmentację.
- Prealokacja - Możliwe jest jednokrotne alokowanie bloków dla pliku przed jego zapisem - dzięki temu dwa jednoczesne zapisy nie powodują problemów z fragmentacją gdy usunięto jeden plik.
- Nowe pliki zapisywane są tak, by dane były równomiernie rozmieszczone.
- Blokowanie niewielkiego % przestrzeni dyskowej celem takiego zapisywania (lub przepisywania) danych, by zminimalizować fragmentację (często 1-5%).

Ext4 - katalogi

- Specjalizowana forma drzewiastej struktury H-tree
- Istotną cechą w gałęzi B-drzewa jest nie wartość, a **zakres** wartości, jakiego możemy spodziewać się poniżej.
- Dochodzimy tak do najniższych elementów zawierających konkretne wartości.
- Dzięki temu możliwe jest bardzo szybkie zachowywanie i przetwarzanie identyfikatorów katalogów.

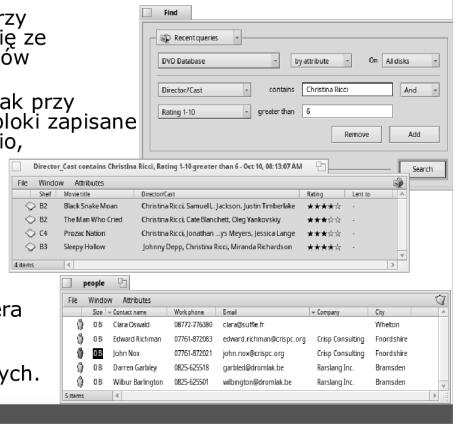
Więcej metadanych! Apple HFS i BeFS

- Apple HFS i późniejsze: Każdy plik lub katalog posiada "resource fork" – strukturę danych opisujących jak plik ma być prezentowany i traktowany przez system:
  - Czy obiekt jest łączem do innego obiektu,
  - Ikona i jej podstawowy kolor,
  - Program jaki otwiera dany plik ("Creator"),
  - Położenie ikony w oknie,
  - Czy obiekt nie posiada wyświetlanej nazwy,
  - Komentarz tekstowy,
  - Rozmiar/położenie okna programu otwierającego plik,
  - I wiele innych, w tym sporo nieużywanych.
- W Mac OS X powoli rezygnowano z forków kosztem ukrytych, niewyświetlanych plików z opisami.



BeFS – pełna dowolność w metadanych

- Zaprojektowany w 1996 roku dla systemu BeOS, używany w Haiku OS.
- Wykorzystano extenty jak przy systemach Ext, składające się ze stałych ilości fizycznych bloków dyskowych.
- Są one jednak „pakowane" jak przy systemach uniksowych - w bloki zapisane bezpośrednio, jednopośrednio, dwupośrednio itd.
- Każdy plik może posiadać dowolną ilość i dowolny typ metadanych. Są one zapisane w dodatkowych extentach opisanych w rekordzie pliku.
- Strukturę metadanych dobiera się po typie pliku.
- Dzięki temu systemu plików można używać jak bazy danych.



MS WinFS, NewFS, OFS, ???

- Nigdy nie pojawił się w ostatecznej wersji Windows.
- Relacyjna baza danych jako system plików,
- Plik: Dane, metadane, zasoby i sposób otwarcia.
  - Sposób otwarcia: Otwierający program i systemowy moduł konwersji.
- Dzięki temu uwalniamy się od pułapki niezgodności formatów...
- ...stąd skupiamy się na danych i relacjach między nimi.
- Program jest w stanie "na żywo" generować strukturę plikową (jak np.. Uniksowy ProcFS) oferując w ten sposób "chmurowe" dyski, E-mail, dokumenty, inne komputery.

Windows 95: Klient poczty Exchange udaje folder

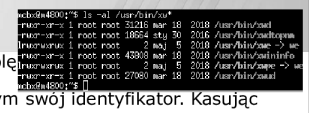
- Problemy:
  - Ciągła ewolucja formatów plików i duże różnice między ich możliwościami.
  - Niezgodności międzyplatformowe.
  - Konieczność zrezygnowania ze zgodności wstecznej

Project Longhorn (Vista): Folder jako kalendar

Skróty i łącza

- Większość współczesnych systemów plików organizuje pliki w wielopoziomowy, hierarchiczny sposób (drzewo katalogów). W tych systemach jednak mogą istnieć wyjątki.
- W MS Windows możliwe jest „podmontowanie" katalogu do innego katalogu - linki. Bardziej popularną formą są realizowane na poziomie graficznego interfejsu skróty, czyli pliki o rozszerzeniu .lnk interpretowane odpowiednio przez graficzną powłokę.
- Siłowe zrealizowanie „skrótu do skrótu" spowoduje (we wcześniejszych wersjach powłoki Windows) zrestartowanie powłoki lub (w późniejszych wersjach) jedynie sprawdzenie obecności docelowego elementu, ewentualnie (w najnowszych wersjach) bezwzględne zwrócenie błędu bez dostępu do ryzykownego pliku na prawach systemu.

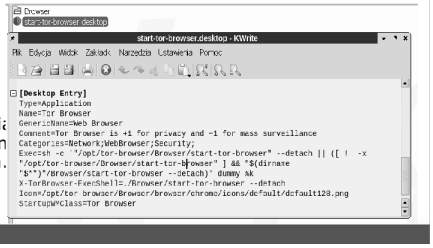
Skróty w Unix / Linux



- Soft link - łącze „miękkie” - uznawane przez konsolę i powłoki graficzne, pod pełną ich kontrolą. Jest osobnym obiektem (plikiem/folderem) posiadającym swój identyfikator. Kasując docelowy obiekt uszkadzamy link i tracimy dane.
  Tworzymy przez **ln -s źródło cel**

- Hard link - stanowi jedynie kolejne odniesienie do istniejącego identyfikatora **pliku**. Kasując jeden hard link wciąż możemy dostać się do pliku innym hard linkiem - dane tracimy dopiero po usunięciu ostatniego. Powłoka nie ma bezpośredniej możliwości kontroli hard linków.
  Tworzymy przez **ln źródło cel**

- Plik .desktop - stanowi pełny zapis wyglądu i zachowania docelowego obiektu. Nazwa, objaśnienie w menu powłoki, ikona, program docelowy i jego parametry, a także otwierane typy plików i ewentualnie tłumaczenia nazwy i opisu. Działa w związku z tym **wyłącznie** w środowisku graficznym. Tworzenie - jak pliku tekstowego.



Dziękuję za uwagę