System plików - koncepcja

Jakub Robaczewski

# Struktura dysku:

* Superblok – przechowuje informacje o rozmiarze dysku
* 1..M węzłów
  + flagi dotyczące węzła (IN\_USE, IS\_START)
  + nazwa pliku
  + rozmiar danych w blokach
  + indeks następnego węzła lub -1, jeśli ostatni
* 1..M bloków danych (każdy po 2048 bajtów)

i-ty węzeł - sizeof(superblock) + N \* sizeof(inode)

i-ty blok danych - sizeof(superblock) + M \* sizeof(inode) + N \* sizeof(datablock)

# Operacje:

* Tworzenie dysku
  + Stwórz plik dysku
  + Oblicz liczbę M
  + Przygotuj w pamięci operacyjnej M struktur
* Otwieranie dysku
  + Otwórz plik dysku
  + Porównaj superblock z realnym rozmiarem pliku (sprawdzenie błędów)
  + Oblicz liczbę M
  + Wczytaj węzły
* Zamykanie dysku
  + Zapisz węzły
  + Zamknij plik dysku
* Kopiowanie (minix -> dysk wirtualny)
  + Sprawdź, czy taki plik już nie istnieje
  + Oblicz rozmiar pliku
  + Sprawdź, czy jest wystarczająco dużo miejsca (oblicz ilość węzłów bez flagi IN\_USE)
  + Zapisz nazwę i odniesienia do węzłów i dane do bloków
* Kopiowanie (dysk wirtualny -> minix)
  + Sprawdź, czy plik istnieje
  + Przejdź po węzłach i zapisz do pliku docelowego
* Wyświetlanie plików
  + Przechodząc całą listę węzłów wypisuj te z flagami IN\_USE i IS\_START
* Usuwanie plików
  + Przejdź po węzłach i szukaj nazwy
  + Przejdź po następnych węzłach (next\_node) i kasuj flagę IN\_USE
* Usuwanie dysku
  + Usuń plik dysku
* Wyświetlanie mapy dysku
  + Superblocku
    - Adres – 0
    - Typ – superblock
    - Rozmiar – sizeof(superblock)
    - Stan – brak
  + Węzły (N-ty węzeł)
    - Adres – sizeof(superblock) + N \* sizeof(inode)
    - Typ – węzeł
    - Rozmiar – sizeof(inode)
    - Stan – flagi
  + Bloki danych (N-ty blok)
    - Adres - sizeof(superblock) + M \* sizeof(inode) + N \* sizeof(datablock)
    - Typ – dane
    - Rozmiar – sizeof(datablock)
    - Stan – w zależności od stanu węzła – IN\_USE – wolny / zajęty
  + Całkowita zajętość
    - Węzły z flagą IN\_USE / Wszystkie węzły

# Testy:

**Test poprawności odczytywania:**

* Stwórz plik z losowym ciągiem danych
* Wykonuj dużą liczbę razy np. 10000
  + Skopiuj plik do systemu plików
  + Pobierz plik z systemu plików
* Skontroluj czy plik początkowy jest równy końcowego

**Test poprawności odzyskiwania danych:**

* Sprawdź czy system plików jest pusty
* Dodaj do systemu plików pliki tak by się zapełnił
* Usuń wszystkie dodane pliki
* Sprawdź czy system plików jest pusty

**Testy momenty krytycznych**

* Dodanie do pełnego dysku
* Dodanie 2 plików o tej samej nazwie
* Usunięcie nieistniejącego pliku
* Kopiowanie nieistniejącego pliku (minix -> system, system -> minix)
* Dodawanie pliku z zerową długością