

Rozszerzona Koncepcja zadanie 6

Mateusz Ostaszewski

Koncepcja realizacji kolejnych funkcji

1. Tworzenie Wirtualnego Dysku

Funkcjonalność została zaimplementowana w `disk_init.cpp`. Dysk zawiera superbloc oraz i-node'y a następnie blocki danych

```
struct Superblock {
    uint64_t diskSize;
    uint32_t blockSize;
    uint64_t diskUsage;
    uint32_t fileCount;
    uint32_t freeBlocks;
    time_t lastModified;
};

struct INode {
    time_t creationTime;
    time_t modificationTime;
    uint32_t permissions;
    std::vector<uint32_t> dataBlockPointers;
    uint64_t fileSize;
    bool isDirectory;
    uint32_t referenceCount;
};
```

2. Kopiowanie Pliku z Dysku Systemu na Dysk Wirtualny

Krok 1: Otwarcie i Weryfikacja Pliku Systemowego

Plik systemowy zostanie otwarty do odczytu, używając strumienia `std::ifstream` w trybie binarnym.

Nastąpi weryfikacja, czy otwarcie pliku przebiegło pomyślnie.

Krok 2: Ustalenie i Ustawienie Rozmiaru Pliku

Rozmiar pliku systemowego zostanie ustalony (np. przez przesunięcie wskaźnika na koniec pliku i wykorzystanie metody `tellg()`) i zapisany w zmiennej.

Sprawdzenie, czy na dysku wirtualnym jest wystarczająco miejsca do przechowania pliku.

Krok 3: Aktualizacja Superbloku

Superblok dysku wirtualnego zostanie wczytany.

W superbloc pole freeBlocks zostanie zaktualizowane poprzez odjęcie liczby bloków, które będą użyte do przechowania pliku.

Pole diskUsage w superbloc zostanie zaktualizowane, dodając rozmiar nowego pliku.

Pole fileCount w superbloc zostanie zwiększone o 1.

Data ostatniej modyfikacji (lastModified) w superbloc zostanie ustawiona na aktualny czas.

Krok 4: Inicjalizacja i Ustawienie I-node

Zostanie utworzony nowy INode dla kopiowanego pliku.

W INode, pole creationTime i modificationTime zostaną ustawione na aktualny czas.

fileSize w INode zostanie ustawione na rozmiar pliku systemowego.

referenceCount w INode zostanie ustawione na 1, ponieważ jest to nowy plik.

Pola uprawnień (permissions) zostaną ustawione zgodnie z wymaganiami systemu lub skopiowane z pliku systemowego.

Wskaźniki na bloki danych (dataBlockPointers) zostaną zainicjalizowane i ustawione na odpowiednie bloki na dysku wirtualnym.

Krok 5: Kopiowanie Danych i Aktualizacja Bloków Danych

Dane z pliku systemowego zostaną odczytane i zapisane w wolnych blokach na dysku wirtualnym.

Każdy z użytych bloków danych zostanie oznaczony jako zajęty, a odpowiednie wskaźniki zostaną zapisane w dataBlockPointers w INode.

Krok 6: Finalizacja Procesu

Plik systemowy zostanie zamknięty.

Aktualizacje INode i superbloc zostaną zapisane z powrotem na dysk wirtualny.

Plik wirtualnego dysku zostanie zamknięty.

Użytkownik zostanie poinformowany o pomyślnym zakończeniu operacji kopiowania

3. Tworzenie Katalogu na Dysku Wirtualnym

Krok 1: Sprawdzenie Dostępności i Wybór Lokalizacji

Sprawdzenie, czy istnieje wystarczająca ilość wolnych bloków na dysku wirtualnym, korzystając z pola freeBlocks w superbloc.

Wyszukanie wolnego INode w strukturze dysku wirtualnego.

Krok 2: Aktualizacja Superbloku

Zaktualizowanie superbloc o nowe informacje:

Zwiększenie fileCount o 1, ponieważ tworzony jest nowy katalog.

Aktualizacja freeBlocks, odejmując liczbę bloków przydzielonych na nowy katalog.

Ustawienie lastModified na aktualny czas, oznaczające ostatnią modyfikację na dysku.

Krok 3: Inicjalizacja I-node dla Katalogu

Utworzenie nowego INode, który będzie reprezentował katalog:

Ustawienie creationTime i modificationTime na aktualny czas.

fileSize zostanie ustawione na 0, ponieważ początkowo katalog nie zawiera plików.

referenceCount zostanie ustawiony na 1 (sam katalog).

permissions zostaną ustawione zgodnie z wymaganiami systemu lub domyślnymi uprawnieniami katalogu.

dataBlockPointers zostaną zainicjalizowane, ale na początku będą wskazywać na puste bloki, ponieważ katalog jest pusty.

Flaga isDirectory zostanie ustawiona na true, aby oznaczyć, że ten i-node reprezentuje katalog.

Krok 4: Przydzielenie Bloków Danych

Jeśli struktura katalogu wymaga fizycznego przydzielenia przestrzeni (np. dla wpisów katalogowych), odpowiednie bloki zostaną przydzielone.

Wskaźniki na te bloki zostaną zapisane w dataBlockPointers w INode.

Krok 5: Finalizacja Procesu

Aktualizacje INode i superbloc zostaną zapisane na dysk wirtualny.

Struktura katalogu zostanie zainicjalizowana w przydzielonych blokach danych (jeśli jest to wymagane).

Informacja o pomyślnym utworzeniu katalogu zostanie przekazana użytkownikowi.

4. Usuwanie Katalogu z Dysku Wirtualnego

Krok 1: Lokalizacja i Weryfikacja Katalogu

Wyszukanie INode odpowiadającego katalogowi, który ma zostać usunięty.

Sprawdzenie, czy znaleziony i-node faktycznie reprezentuje katalog.

Krok 2: Rekurencyjne Usuwanie Zawartości Katalogu

Przejdźcie przez wszystkie wpisy w katalogu (używając informacji z dataBlockPointers w i-node katalogu):

Dla każdego wpisu, który jest katalogiem, proces zostanie powtórzony rekurencyjnie (powrót do Kroku 1).

Dla każdego wpisu, który jest plikiem, plik zostanie usunięty, aktualizując odpowiednio i-node pliku i zwalniając przypisane mu bloki danych.

Krok 3: Usunięcie i-node Katalogu

Po usunięciu wszystkich plików i podkatalogów, i-node reprezentujący katalog zostanie usunięty.

Zmniejszenie fileCount w superblocie o 1.

Krok 4: Zwolnienie Przypisanych Bloków Danych Katalogu

Bloki danych przypisane do katalogu (wskazywane przez dataBlockPointers w i-node) zostaną zwolnione.

Aktualizacja freeBlocks w superblocie.

Krok 5: Aktualizacja Superbloku

aktualizowanie lastModified w superblocie na aktualny czas.

Krok 6: Zapisanie Zmian na Dysku

Zapisanie zaktualizowanego superbloku

i zmodyfikowanych informacji o i-node'ach na dysk wirtualny.

Krok 7: Finalizacja Procesu

Zamknięcie pliku wirtualnego dysku.

Informowanie użytkownika o pomyślnym usunięciu katalogu i jego zawartości.

5. Kopiowanie Pliku z Dysku Wirtualnego na Dysk Systemowy

Krok 1: Lokalizacja Pliku na Dysku Wirtualnym

Wyszukanie INode odpowiadającego plikowi, który ma zostać skopiowany, upewniając się, że jest to plik, a nie katalog (sprawdzając flagę isDirectory).

Krok 2: Otwarcie Pliku Docelowego

Utworzenie lub otwarcie pliku systemowego, do którego dane będą zapisywane, używając strumienia wyjściowego std::ofstream w trybie binarnym.

Sprawdzenie, czy plik został otwarty poprawnie.

Krok 3: Czytanie Danych z Dysku Wirtualnego

Przejdźcie przez wskaźniki w dataBlockPointers w i-node pliku.

Odczyt danych z każdego bloku danych wskazywanego przez dataBlockPointers i zapisanie ich do pliku systemowego.

Krok 4: Zapisanie Danych na Dysk Systemowy

Zapisanie odczytanych danych do pliku docelowego na dysku systemowym.

Upewnienie się, że wszystkie dane zostały poprawnie zapisane i że plik jest kompletny.

Krok 5: Zakończenie Procesu

Zamknięcie pliku systemowego.

Informowanie użytkownika o pomyślnym zakończeniu procesu kopiowania.

6. Wyświetlanie Katalogu Dysku Wirtualnego

Krok 1: Lokalizacja i Weryfikacja Katalogu

Wyszukanie INode odpowiadającego katalogowi, który ma zostać wyświetlony.

Sprawdzenie, czy znaleziony i-node faktycznie reprezentuje katalog (sprawdzając flagę isDirectory).

Krok 2: Odczytanie Zawartości Katalogu

Dostęp do Listy Wskaźników w INode Katalogu

Iteracja przez Wskaźniki Bloków Danych

Odczytanie i-node'ów dla Każdego Pliku/Podkatalogu

Zbieranie Informacji o Zawartości Katalogu

Krok 3: Wyświetlenie Informacji o Plikach i Podkatalogach

Dla każdego znalezionej pliku lub podkatalogu:

Wyświetlenie nazwy, daty modyfikacji, rozmiaru i innych istotnych metadanych (np. uprawnień) związanych z i-node'm.

Jeśli jest to katalog, zaznaczenie tego faktu, np. poprzez dodanie '/' po nazwie.

Krok 4: Dodatkowe Informacje o Katalogu

Wyświetlenie ogólnych informacji o samym katalogu, takich jak całkowita liczba plików/podkatalogów, całkowity rozmiar plików w katalogu, i inne metadane, jeśli są dostępne.

Krok 5: Zakończenie Procesu

Zakończenie procesu i ewentualne zamknięcie strumieni czy plików, jeśli to konieczne.

7. Tworzenie Twardego Dowiązania

Krok 1: Lokalizacja Oryginalnego Pliku

Wyszukanie INode odpowiadającego plikowi, do którego ma zostać utworzone twarde dowiązanie.

Sprawdzenie, czy znaleziony i-node faktycznie reprezentuje plik (a nie katalog).

Krok 2: Utworzenia Dowiązania

Sprawdzenie, czy istnieje wolny i-node do reprezentowania nowego dowiązania.

Krok 3: Utworzenie Nowego I-node dla Dowiązania

Utworzenie nowego INode dla twardego dowiązania:

Skopiowanie metadanych z oryginalnego i-node'a, w tym fileSize, permissions, creationTime, modificationTime, itp.

Ustawienie referenceCount w obu i-node'ach (oryginalnym i nowym) na wartość o 1 większą, ponieważ istnieje dodatkowe odwołanie do pliku.

Krok 4: Ustawienie Wskaźników na Bloki Danych

W nowym i-node dowiązania ustawienie wskaźników dataBlockPointers tak, aby wskazywały na te same bloki danych, co i-node oryginalnego pliku.

Krok 5: Zapisanie Zmian na Dysku

Zapisanie zaktualizowanego superbloc oraz zmodyfikowanych informacji o i-node'ach na dysk wirtualny.

Krok 6: Finalizacja Procesu

Potwierdzenie dla użytkownika, że twarde dowiązanie zostało pomyślnie utworzone.

8. Usuwanie Pliku lub Dowiązania z Wirtualnego Dysku

Krok 1: Lokalizacja i Weryfikacja Pliku/Dowiązania

Rozpoczęcie przeszukiwania od katalogu głównego lub wskazanego katalogu startowego.

Iteracja przez zawartość każdego katalogu, korzystając z i-node'ów reprezentujących katalogi i ich dataBlockPointers, które wskazują na bloki danych zawierające informacje o plikach i podkatalogach.

W każdym przeszukiwanym katalogu, odczytanie listy plików i podkatalogów z bloków danych wskazywanych przez dataBlockPointers w i-node katalogu.

Porównanie nazw plików i dowiązań znajdujących się w katalogu z nazwą szukanego pliku/dowiązania.

Jeśli nazwa pliku/dowiązania zostanie znaleziona, proces przejdzie do identyfikacji i-node'a odpowiadającego temu plikowi/dowiązaniu.

Uzyskanie dostępu do i-node'a reprezentującego znaleziony plik/dowiązanie.

Sprawdzenie, czy i-node faktycznie reprezentuje plik (a nie katalog), sprawdzając flagę isDirectory. Krok

2: Aktualizacja Liczby Referencji w I-node

Zmniejszenie referenceCount w i-node. Jeżeli po zmniejszeniu referenceCount wynosi zero, oznacza to, że nie ma więcej dowiązań do tego pliku, i można przejść do jego usunięcia.

Jeżeli referenceCount jest większe niż zero, oznacza to, że istnieją inne dowiązania do tego pliku, i w tym przypadku proces zostaje zakończony (plik/dowiązanie nie jest usuwane).

Krok 3: Usunięcie Pliku/Dowiązania

Jeśli referenceCount wynosi zero, plik lub dowiązanie zostaje usunięte:

Zwalniane są bloki danych wskazywane przez dataBlockPointers w i-node pliku/dowiązania.

Aktualizacja listy wolnych bloków na dysku wirtualnym oraz freeBlocks w superbloc.

Usunięcie informacji o i-node z listy i-node'ów.

Krok 4: Aktualizacja Superbloku

Zaktualizowanie fileCount w superblock, jeśli usunięty został plik (a nie tylko dowiązanie).

Aktualizacja lastModified w superblock na aktualny czas.

Zapisanie zmian w superblock na dysk wirtualny.

Krok 5: Zakończenie Procesu

Zapisanie zmian na dysku wirtualnym, w tym aktualizacji struktury i-node'ów i mapy bloków.

Informowanie użytkownika o pomyślnym usunięciu pliku/dowiązania.

9. Dodanie do Pliku o Zadanej Nazwie n Bajtów

Krok 1: Lokalizacja Pliku- taki sam jak w **Usuwanie Pliku lub Dowiązania z Wirtualnego Dysku**

Krok 2: Sprawdzenie Dostępności Miejsca

Określenie, czy istnieje wystarczająco wolnego miejsca na dysku wirtualnym do dodania n bajtów do pliku.

Sprawdzenie, czy istniejące bloki danych pliku mogą pomieścić dodatkowe bajty; jeśli nie, należy znaleźć dodatkowe wolne bloki.

Krok 3: Odczytanie i Modyfikacja Zawartości Pliku

Odczytanie danych z bloków danych wskazywanych przez dataBlockPointers w i-node pliku.

Dodanie n bajtów do końca zawartości pliku. Bajty mogą być puste (np. zera).

Krok 4: Zapisanie Zmodyfikowanych Danych

Zapisanie zmodyfikowanych danych z powrotem do istniejących bloków danych. Jeśli wymagane są dodatkowe bloki, zostaną one przydzielone i zapisane.

Aktualizacja dataBlockPointers w i-node pliku, aby uwzględnić wszelkie nowe bloki danych.

Krok 5: Aktualizacja I-node Pliku

Zaktualizowanie pola fileSize w i-node pliku, dodając do niego liczbę dodanych bajtów.

Aktualizacja pola modificationTime w i-node, aby odzwierciedlić czas modyfikacji pliku.

Krok 6: Aktualizacja Superbloku

Jeśli konieczne było przydzielenie dodatkowych bloków, aktualizacja freeBlocks oraz diskUsage w superblock.

Aktualizacja lastModified w `superblock` na aktualny czas, aby odzwierciedlić zmianę na dysku wirtualnym.

Krok 7: Zakończenie Procesu

Zapisanie wszystkich zmian na dysku wirtualnym, w tym aktualizacji i-node pliku oraz superblock.

Informowanie użytkownika o pomyślnym dodaniu n bajtów do pliku.

10. Skrócenie Pliku o Zadanej Nazwie o n Bajtów

Krok 1 taki sam jak w poprzednim

Krok 2: Obliczenie Nowego Rozmiaru Pliku

Obliczenie nowego rozmiaru pliku po skróceniu o n bajtów.

Sprawdzenie, czy nowy rozmiar pliku jest logicznie poprawny (np. czy nie jest mniejszy niż zero).

Krok 3: Aktualizacja I-node Pliku

Zaktualizowanie fileSize w i-node pliku do nowego rozmiaru.

Jeśli nowy rozmiar pliku jest mniejszy niż poprzedni, określenie, które bloki danych należy zwolnić.

Krok 4: Zwolnienie Nadmiarowych Bloków Danych

Jeśli konieczne, zwolnienie nadmiarowych bloków danych, które nie są już potrzebne do przechowywania zawartości pliku.

Aktualizacja dataBlockPointers w i-node, aby odzwierciedlić zmiany w alokacji bloków.

Krok 5: Aktualizacja Superbloku

Jeśli zwolnione zostały bloki danych, zaktualizowanie freeBlocks w superblock.

Aktualizacja lastModified w superblock na aktualny czas.

Krok 6: Zapisanie Zmian na Dysku

nie zaktualizowanego i-node pliku oraz zmian w superblock na dysk wirtualny.

Krok 7: Finalizacja Procesu

Potwierdzenie dla użytkownika, że plik został pomyślnie skrócony.

11. Wyświetlenie Informacji o Zajętości Dysku

Krok 1: Odczyt Superbloku

Odczytanie danych z Superblock dysku wirtualnego, w tym pola diskUsage, totalBlocks, freeBlocks i lastModified.

Krok 2: Obliczanie Zajętości Dysku

Wykorzystanie pola diskUsage do uzyskania informacji o całkowitym wykorzystaniu przestrzeni dyskowej na dysku wirtualnym w gigabajtach.

Obliczenie całkowitej przestrzeni dyskowej na podstawie pola totalBlocks i blockSize oraz przeliczenie jej na gigabajty.

Obliczenie liczby wolnych bloków na podstawie pola freeBlocks i blockSize oraz przeliczenie jej na megabajty.

Krok 3: Prezentacja Danych

Wyświetlenie uzyskanych informacji w czytelny sposób, z użyciem jednostek GB (gigabajtów) i MB (megabajtów) dla przestrzeni dyskowej i liczby wolnych bloków.

Przedstawienie daty ostatniej modyfikacji dysku w czytelny sposób.