Sprawozdanie: B-drzewo

Kamil Lubarski 188569

1. Wprowadzenie

Celem projektu było zaimplementowanie indeksowej organizacji pliku. Jako sposób implementacji wybrano B-drzewo. Opiera się ono na przechowywaniu w każdym węźle 2 elementów: rekordów i dzieci. Na rekord trzymany w węźle składa się jego klucz oraz lokalizacja w pliku z rekordami — dzięki temu zmniejszamy zużycie pamięci (dane rekordu odczytujemy wedle potrzeby podczas wyświetlania ich zawartości). Na każde z dzieci składa się natomiast wyłącznie jego lokalizacja w pliku z węzłami. Podczas wykonywania operacji na drzewie kolejne węzły ładowane są tylko wtedy, gdy to wymagane — ponownie zmniejszamy dzięki temu zużycie pamięci.

1. Implementacja

Projekt wykonano w języku C++. Składa się na niego 5 klas: B-drzewo, Węzeł, Element, Rekord i KeyGen oraz pomniejsze funkcje odpowiedzialne za interakcję z użytkownikiem. Rekord stanowi klucz typu int oraz trójka liczb typu double; każda z nich reprezentuje prawdopodobieństwo: P(A), P(B), P(A∪B). Poniżej zamieszczono skrócony opis każdej z klas:

* Rekord — odpowiada za trzymanie danych należących do rekordu a także jego zapis / odczyt.
* Element — służy jako łącznik pomiędzy węzłem a rekordem. Zawiera w sobie klucz i offset rekordu.
* Węzeł — składają się na niego elementy i offsety dzieci. Dla ułatwienia implementacji posiada on również wskaźnik na rodzica oraz własny offset. Służy do zapisu / odczytu informacji o węźle z dysku.
* B-drzewo — w nim znajduje się główna logika programu odpowiedzialna za odpowiednie zarządzanie węzłami i ich strukturą. Posiada funkcje takie jak: insert, update, delete, print.
* KeyGen — powstała na potrzeby losowego generowania kluczy do eksperymentu przeprowadzonego w ramach projektu.

Pozostała logika potrzebna do działania programu, taka jak menu wyboru i prezentacja wyników nie należy do żadnej z klas i występuje na tym samym poziomie co funkcja main. Program prezentuje działanie poprzez wyświetlenie obecnego stanu drzewa po każdej jego modyfikacji. Wyświetla on również dane o wykonanych zapisach / odczytach dla każdej operacji. Pliki przechowujące rekordy jak i węzły są plikami binarnymi — znajdują się w nich dane w postaci bitowej bez separatorów. W przypadku zapisu węzła za każdym razem zapisywanych jest 4 elementów oraz 5 dzieci — w przypadku ich mniejszej liczby w pliku zapisywane są wartości domyślne, odpowiednio interpretowane przez program.

1. Eksperyment

W ramach projektu przeprowadzony został eksperyment mający za zadanie określenie efektywności wykonanej implementacji. Polegał on na przeprowadzeniu serii symulacji operacji na b-drzewie i prezentacji uzyskanych wyników. Eksperyment przed wykonaniem odczytywany był z dysku — w pliku tekstowym znajdowały się 3 liczby odpowiadające kolejno za ilość dodanych, zaktualizowanych i usuniętych rekordów. Liczby te były następnie interpretowane przez program.

Poniżej przedstawiona została tabela zawierająca uzyskane podczas eksperymentu wartości. W ramach symulacji rozmiar stopień drzewa (d) został ustawiony na 2.

| wstawianie | średnich odczytów | średnich zapisów | aktualizacja | średnich odczytów | średnich zapisów | usuwanie | średnich odczytów | średnich zapisów |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 0 | 1 | 5 | 0 | 0 | 8 | 1 | 1 |
| 100 | 2 | 2 | 50 | 2 | 0 | 75 | 3 | 2 |
| 1000 | 4 | 2 | 500 | 3 | 0 | 750 | 4 | 2 |
| 10000 | 6 | 2 | 5000 | 5 | 0 | 7500 | 6 | 2 |

Poniżej przedstawiona została tabela prezentująca zależność pomiędzy ilością rekordów w drzewie, a rozmiarem plików na dysku uzyskanych podczas eksperymentu.

| ilość dodanych rekordów | rozmiar pliku z rekordami (KiB) | rozmiar pliku z węzłami (KiB) |
| --- | --- | --- |
| 10 | 0,27 | 0,20 |
| 100 | 2,73 | 1,62 |
| 1000 | 27,34 | 15,79 |
| 10000 | 273,44 | 156,66 |