Sprawozdanie: scalanie polifazowe

Kamil Lubarski 188569

1. Wprowadzenie

Celem projektu było zaimplementowanie algorytmu służącego do sortowania plików sekwencyjnych. Jako sposób implementacji wybrano scalanie polifazowe wykorzystujące 3 taśmy. Opiera się ona na dwóch etapach - dystrybucji i scalaniu. Etap dystrybucji rozdziela taśmę wejściową na dwie taśmy w ściśle określony sposób, tak, aby ilość serii na dwóch taśmach była dwiema kolejnymi liczbami z ciągu Fibonacciego; natomiast etap scalania łączy rekordy z dwóch niepustych taśm oraz wpisuje je na trzecią z taśm.

1. Implementacja

Projekt wykonano w języku C++. Składa się on z kilku funkcji, z których dwie główne odpowiadają za rozdział taśmy wejściowej na dwie taśmy oraz scalanie dwóch taśm i zapisanie wyniku na trzeciej. Taśmami nazywane są pliki o rozszerzeniu .bin, w których jeden po drugim zapisane są w reprezentacji bitowej kolejne rekordy (każdy rekord składa się z 24 bajtów). Program korzysta z dokładnie 3 taśm. Rekord stanowi trójka liczb typu double; każda z nich reprezentuje prawdopodobieństwo: P(A), P(B), P(A∪B). Program porządkuje rekordy wejściowe w kolejności niemalejącej, jako kryterium stosując P(A∩B) wyliczone ze wzoru P(A) + P(B) - P(A∪B).

Na potrzeby projektu utworzone zostały trzy klasy:

* Rekord — odpowiada za trzymanie danych należących do rekordu, wyświetlanie ich oraz zapis / odczyt danych z taśmy.
* Blok — służy jako łącznik pomiędzy warstwą fizyczną (operującej na taśmach) oraz warstwą logiczną (sortującej odczytane rekordy). Blok przechowuje w wektorze ściśle określoną, lecz modyfikowalną ilość rekordów. Na potrzeby projektu ilość ta została ustawiona na 100 rekordów.
* Fibonacci — odpowiada za generowanie liczb z ciągu Fibonacciego oraz sprawdzanie ich poprawności.

Pozostała logika potrzebna do działania programu, taka jak menu wyboru, sortowanie i prezentacja wyników nie należy do żadnej z klas i występuje na tym samym poziomie co funkcja main. Program prezentuje działanie poprzez wyświetlenie pliku wejściowego przed oraz po posortowaniu. Wyświetla on również dane potrzebne do wykonania niżej opisanego eksperymentu.

1. Eksperyment

W ramach projektu przeprowadzony został eksperyment mający za zadanie określenie efektywności wykonanej implementacji. Polegał on na przeprowadzeniu serii symulacji sortowania określonej ilości rekordów i porównania uzyskanej ilości faz oraz operacji dyskowych do ilości wyliczonej ze wzorów przedstawionych na wykładzie.

We wzorach poniżej przyjęto oznaczenia:

* N — ilość rekordów w pliku wejściowym
* r — ilość serii w pliku wejściowym (seria to niemalejący ciąg rekordów)
* b — ilość rekordów w bloku (rozmiar bloku podzielony przez rozmiar rekordu)

Teoretyczna liczba faz = 1,45 log₂r

Teoretyczna liczba operacji dyskowych = 2N(1,04 log₂r + 1) / b

Poniżej przedstawiona została tabela zestawiająca uzyskane podczas eksperymentu wartości z wartościami teoretycznymi. W ramach symulacji rozmiar bloku (b) został ustawiony na 100 rekordów (2400 bajtów).

| N | r | fazy | operacje dyskowe | oczekiwane fazy | oczekiwane operacje dyskowe |
| --- | --- | --- | --- | --- | --- |
| 100 | 50 | 8 | 27 | 8 | 14 |
| 1000 | 498 | 13 | 230 | 13 | 206 |
| 10000 | 4892 | 18 | 2842 | 18 | 2749 |
| 100000 | 48886 | 23 | 35466 | 23 | 34400 |
| 1000000 | 488902 | 27 | 413899 | 27 | 413103 |
| 10000000 | 4890927 | 32 | 4851336 | 32 | 4822109 |

Na podstawie uzyskanych wyników można zauważyć, że ilość faz zgadza się z oczekiwaną, natomiast ilość operacji dyskowych jest od oczekiwanej większa. Zgodność ilości faz wynika z założeń metody polifazowej, która konkretnie określa ich przebieg. Rozbieżność ilości operacji dyskowych może być natomiast skutkiem nierównej długości serii na obu taśmach. Dodatkowo można zauważyć, że ilość serii oscyluje w okolicy połowy ilości rekordów.

Poniżej przedstawione zostały te same dane na dwóch wykresach: na pierwszym przedstawiono zależność ilości faz od ilości rekordów; na drugim zależność ilości operacji dyskowych. W przypadku drugiego wykresu ustawiona została skala logarytmiczna.



