

temat: Przetwarzanie wyników wyszukiwania

skład: Monika Klęsk, Sylwia Firszt, Mateusz Burghardt, Krzysztof Kabata

opis: program zajmuje się generowaniem listy wyników wyszukiwania na podstawie otrzymanych ID plików, jest częścią systemu wyszukiwania.

Technologia: c++, MPI, openMP

dane wejściowe: plik tekstowy z liczbami (tabela 2wymiarowa)

dane wyjściowe: posortowany wynik wyszukiwania w postaci tabeli rekordów, np.
Object[] = {[id:4,ocena:4.82], [id:3,ocena:3.51]}

algorytm:

- konwertowanie linii w pliku tekstowym na tabele ID
- porównywanie tabel między sobą aby wyciągnąć tylko powtarzające się ID
- zastąpienie każdego ID rekordem/obiektem w którym jest zmienna, według której następuje sortowanie wyników (np. id "12" to rekord w którym jest zmienna "ocena" = 4.83)
- sortowanie tabeli rekordów/obiektów wg wartości zmiennej
- zwrócenie wyniku

przykład:

- plik wejściowy:

```
# 1 2 4 5
# 2 4
# 1 4
# 2 4
# 1 2 4
```

- wyjście:

```
# Prace zaczyna proces: 0 (wszystkich 5)
# Prace zaczyna proces: 1 (wszystkich 5)
# Watek 1 otwiera plik dane_wejsciove.txt
# Prace zaczyna proces: 3 (wszystkich 5)
# Prace zaczyna proces: 2 (wszystkich 5)
# Prace zaczyna proces: 4 (wszystkich 5)
# EngineArray::arrayUnique() czas: 29421 qs
# Proces 3 odebral IDS
# EngineRec::generateRec() czas: 4 qs
# Tabela RECS:
# ID 4 ocena 4
```

podział pracy:

Mateusz Burghardt: generowanie tabeli kluczy (IDS) z pliku wejściowego i przesyłanie jej kolejnemu procesowi, wyświetlanie tabeli rekordów, tworzenie dokumentacji projektu

Krzysztof Kabata: generowanie tabeli unikalnych ID na podstawie otrzymanej tabeli IDS, przekazywanie wyniku do kolejnego procesu, zarządzanie projektem

Monika Klęsk: odbiór tabeli unikalnych ID, generowanie tabeli rekordów, wysyłanie do kolejnego procesu

Sylwia Firszt: odbiór tabeli rekordów, posortowanie jej, przesyłanie do procesu głównego