Zadania - zestaw 1.

—

Zarządzanie pamięcią, biblioteki, pomiar czasu

Zadanie 1. Alokacja tablicy ze wskaźnikami na bloki pamięci zawierające tablicę wskaźników (25%)

Zaprojektuj i przygotuj zestaw funkcji (bibliotekę) do zarządzania tablicą bloków, w których to blokach pamięci zapisywane są rezultaty operacji porównywania plików wiersz po wierszu (poleceniem `diff`) sekwencji par plików — sekwencja ich nazw jest parametrem funkcji.

Biblioteka powinna umożliwiać:

- Utworzenie tablicy wskaźników (tablicy głównej) — w tej tablicy będą przechowywane wskaźniki na bloki operacji edycyjnych — pierwszy element tablicy głównej zawiera wykaz operacji edycyjnych dla pierwszej pary plików, drugi element dla drugiej pary, itd. Pojedynczy blok operacji edycyjnych (element wskazywany z tablicy głównej), to tablica wskaźników na poszczególne operacje edycyjne
- Definiowanie sekwencji par plików
- Przeprowadzenie porównania (dla każdego elementu sekwencji) oraz zapisanie wyniku porównania do pliku tymczasowego
- Utworzenie, na podstawie zawartość pliku tymczasowego, bloku operacji edycyjnych — tablicy wskaźników na operacje edycyjne, ustawienie w tablicy głównej (wskaźników) wskazania na ten blok; na końcu, funkcja powinna zwrócić indeks elementu tablicy (głównej), który zawiera wskazanie na utworzony blok — dla pokazanego (niżej) przykładu powinna więc zwrócić 0
- Zwrócenie informacji o ilości operacji w danym bloku operacji edycyjnych — dla przykładu, pokazanego poniżej, zwracaną wartością powinno być 3
- Usunięcie, z pamięci, bloku (operacji edycyjnych) o zadanym indeksie
- Usunięcie, z pamięci, określonej operacji dla podanego bloku operacji edycyjnych

Przykład — załóżmy, że sekwencja nazw plików zawiera tylko jedną parę ('*a.txt*', '*b.txt*').

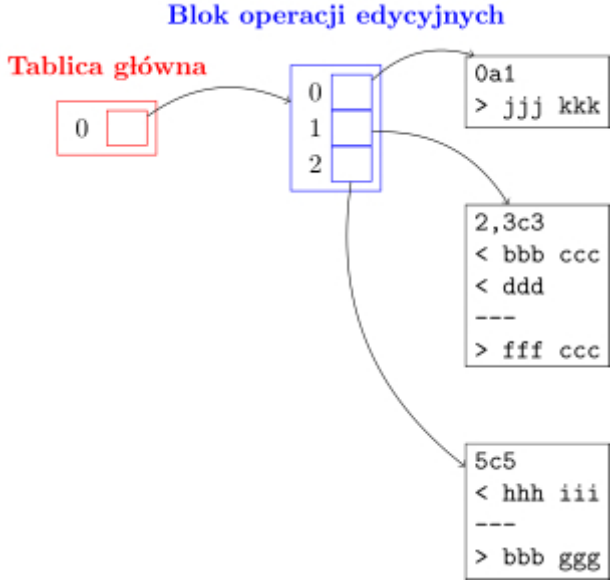
Zawartość pliku *a.txt*:

```
aaa
bbb ccc
ddd
```

```
eee
hhh iii
```
Zawartość pliku *b.txt*:

```
jjj kkk
aaa
fff ccc
eee
bbb ggg
```
Wynik wykonania `diff a.txt b.txt`:

```
0a1
> jjj kkk
2,3c3
< bbb ccc
< ddd
---
> fff ccc
5c5
< hhh iii
---
> bbb ggg
```
W tym przypadku tablica główna powinna zawierać tylko jeden wskaźnik na blok operacji edycyjnych (bo mamy tylko jedną parę plików). Blok operacji edycyjnych powinien być trzyelementową tablicą wskaźników na napisy z treścią operacji edycyjnych.



Tablice / bloki powinny być alokowane przy pomocy funkcji `calloc()` (alokacja dynamiczna).

Przygotuj plik *Makefile*, zawierający polecenia kompilujące pliki źródłowe biblioteki oraz tworzące biblioteki w dwóch wersjach: statyczną i współdzieloną.

Zadanie 2. Program korzystający z biblioteki (25%)

Napisz program testujący działanie funkcji z biblioteki z zadania 1.

Jako argumenty przekaż liczbę elementów tablicy głównej (liczbę par plików) oraz listę zadań do wykonania. Zadania mogą stanowić zadania porównania wszystkich par w sekwencji lub zadania usunięcia bloku o podanym indeksie bądź usunięcia operacji o podanym indeksie.

Operacje mogą być specyfikowane w linii poleceń na przykład jak poniżej:

- `create_table rozmiar` — stworzenie tablicy o rozmiarze "rozmiar"
- `compare_pairs file1A.txt:file1B.txt file2A.txt:file2B.txt …` — porównanie para plików: file1A.txt z file1B.txt, file2A.txt z file2B.txt, itd
- `remove_block index` – usuń z tablicy bloków o indeksie *index*
- `remove_operation block_index operation_index` — usuń z bloku o indeksie *block_index* operację o indeksie *operation_index*

Program powinien stworzyć tablice bloków o zadanej liczbie elementów

W programie zmierz, wypisz na konsolę i zapisz do pliku z raportem czasy realizacji podstawowych operacji:

- Przeprowadzenie porównania par plików — różna ilość elementów w sekwencji par (mała (np. 1-5), średnia oraz duża ilość par) oraz różny stopień podobieństwa plików w parze (pliki bardzo podobne do siebie, pliki w średnim stopniu niepodobne do siebie, pliki w znacznym stopniu niepodobne do siebie)
- Zapisanie, w pamięci, bloków o różnych rozmiarach (odpowiadających rozmiarom różnych przeprowadzonych porównań)
- Usunięcie zaalokowanych bloków o różnych rozmiarach (odpowiadających rozmiarom różnych przeprowadzonych porównań)
- Na przemian kilkakrotne dodanie i usunięcie zadanej liczby bloków

Mierząc czasy poszczególnych operacji, zapisz trzy wartości: czas rzeczywisty, czas użytkownika i czas systemowy. Rezultaty umieść pliku *raport2.txt* i dołącz do archiwum zadania.

Zadanie 3. Testy i pomiary (50%)

a. (25%) Przygotuj plik *Makefile*, zawierający polecenie uruchamiania testów oraz polecenia kompilacji programu z zad 2 na trzy sposoby:
    - Z wykorzystaniem bibliotek statycznych,
    - Z wykorzystaniem bibliotek dzielonych (dynamiczne, ładowane przy uruchomieniu programu),
    - Z wykorzystaniem bibliotek ładowanych dynamicznie (dynamiczne, ładowane przez program).

    Wyniki pomiarów zbierz w pliku results3a.txt. Otrzymane wyniki krótko skomentuj.

b. (25%) Rozszerz plik *Makefile* z punktu 3a) dodając możliwość skompilowania programu na trzech różnych poziomach optymalizacji — `-O0`…`-Os`. Przeprowadź ponownie pomiary, kompilując i uruchamiając program dla różnych poziomów optymalizacji.
   Wyniki pomiarów dodaj do pliku results3b.txt. Otrzymane wyniki krótko skomentuj.

Wygenerowane pliki z raportami załącz jako element rozwiązania.

Uwaga: Do odczytania pliku można użyć funkcji `read()` (`man read`), do wywołania zewnętrznego polecenia Unixa można użyć funkcji `system()` (`man system`).