

Zadania - zestaw 1.

Otwarto: wtorek, 6 marca 2018, 08:10

Wymagane do: niedziela, 18 marca 2018, 08:10

Zarządzanie pamięcią, biblioteki, pomiar czasu

Zadanie 1. Alokacja tablicy z wskaźnikami na bloki pamięci zawierające znaki (25%)

Zaprojektuj i przygotuj zestaw funkcji (bibliotekę) do zarządzania tablicą bloków zawierających znaki.

Biblioteka powinna umożliwiać:

- tworzenie i usuwanie tablicy
- dodanie i usunięcie bloków na które wskazują wybrane indeksy elementów tablicy
- wyszukiwanie bloku w tablicy, którego suma znaków (kodów ASCII) w bloku jest najbliższa elementowi o zadanym numerze,

Tablice i bloki powinny być alokowane przy pomocy funkcji *calloc* (alokacja dynamiczna) jak również powinny wykorzystywać tablicę dwuwymiarową (statyczny przydział pamięci).

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

Zadanie 2. Program korzystający z biblioteki (25%)

Napisz program testujący działanie funkcji z biblioteki z zadania 1.

Jako argumenty przekaz liczbę elementów tablicy, rozmiar bloku, sposób alokacji pamięci oraz spis wykonywanych operacji. Zakładamy, że możliwe jest zlecenie trzech operacji (np. stworzenie tablicy, wyszukanie elementu oraz usunięcie i dodanie zadanej liczby bloków albo stworzenie tablicy, usunięcie i dodanie zadanej liczby bloków i naprzemienne usunięcie i dodanie zadanej liczby bloków).

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

- * *create_table rozmiar rozmiar_bloku* - stworzenie tablicy o rozmiarze "rozmiar" i blokach o rozmiarach "rozmiar bloku"
- * *search_element wartość* - wyszukanie elementu o wartości ASCII zbliżonej do pozycji "wartość"
- * *remove number* - usuń "number" bloków
- * *add number* - dodaj "number" bloków
- * *remove_and_add number* - usuwaj i dodawaj na przemian blok "number" razy

Program powinien stworzyć tablice bloków o zadanej liczbie elementów i rozmiarze bloków. Dane można wygenerować na stronach typu [generatedata.com](#) albo użyć danych losowych.

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

- stworzenie tablicy z zadaną liczbą bloków o zdanym rozmiarze i przy pomocy wybranej funkcji alokującej,
- wyszukanie najbardziej podobnego elementu z punktu widzenia sumy znaków do elementu zadanego jako argument
- usunięcie kolejno zadanej liczby bloków a następnie dodanie na ich miejsce nowych bloków
- na przemian usunięcie i dodanie 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 polecenia kompilujące program 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),
- oraz uruchamiający testy.

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 na różnych poziomach optymalizacji.

Wyniki pomiarów dodaj do pliku results3b.txt. Otrzymane wyniki krótko skomentuj.

Wygenerowane pliki z raportami załącz jako element rozwiązania.