**Zarządzanie pamięcią, biblioteki, pomiar czasu**

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

**Zaprojektuj i przygotuj zestaw funkcji (bibliotekę) do zarządzania tablicą bloków, w których zapisywane są rezultaty operacji zliczania lini, słów i znaków (poleceniem *wc*) w plikach przekazywanych jako odpowiedni parametr.**

*Poniżej, jeżeli mowa o zliczaniu ilośći słów, chodzi o wykonanie programu wc w trybie domyślnym.*

**Interfejs biblioteki powinien obejmować 5 funkcji realizujących następujące zadania:**

1. Utworzenie i zwrócenie struktury zawierającej:
   * Zainicializowaną zerami (calloc()) tablicę wskaźników w której będą przechowywane wskaźniki na bloki pamięci.
   * Rozmiar tablicy, tj. maksymalna ilość bloków jakie można zapisać.
   * Aktualny rozmiar, tj. ilość zapisanych bloków.
2. Przeprowadzenie procedury zliczania ilości linii i słów dla podanego pliku:
   * Procedura przyjmuje strukturę z pkt.1 oraz nazwę pliku.
   * Uruchomienie (system()) programu wc do zliczenia lini, słów i znaków dla zadanego pliku i przesłanie wyniku do pliku tymczasowego w katalogu /tmp.
   * Zarezerwowanie bloku pamięci (calloc()) o rozmiarze odpowiadającym rzeczywistemu rozmiarowi danych znajdujących się w buforze tymczasowym i przeniesienie tych danych do nowo zaalokowanego bloku pamięci.
   * Inkrementację licznika ilości zapisanych bloków.
   * Usunięcie pliku tymczasowego.
3. Zwrócenie zawartości bloku o zadanym indeksie. Procedura przyjmuje strukturę z pkt.1
4. Usunięcie z pamięci bloku o zadanym indeksie. Procedura przyjmuje strukturę z pkt.1
5. Zwolnienie pamięci tablicy wskaźników.

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.**

Program powinien udostępniać interfejs typu REPL, tj. zczytywać komendy użytkownika ze standardowego wejścia linia po linii (fgets()).

**Program powinien udostępniać następujące komendy:**

1. init size — stworzenie tablicy o rozmiarze *size*(int)
2. count file — zliczenie słów i linii w pliku *file* zapis wyniku w tablicy
3. show index – wyświetlenie zawartości tablicy o indeksie *index*
4. delete index index — usunięcie z tablicy bloków o indeksie *index*
5. *destroy –*usunięcie z pamięci tablicy z pkt. 1

Po wykonaniu każdej z operacji program powinien wypisać na standardowe wyjście czas wykonania tej operacji: rzeczywisty, użytkownika i systemosfwy.

Przygotuj wpis w pliku *Makefile* kompilujący program w trzech wariantach:

1. Z wykorzystaniem biblioteki statycznej
2. Z wykorzystaniem bilbioteki dzielonej (dynamiczne ładowanie podczas uruchamiania programu)
3. Z wykorzystaniem biblioteki ładowanej dynamicznie (dlopen())

**Zadanie 3. Testy i pomiary (50%)**

**A)** (25%) W pliku *Makefile* stwórz wpisy przeprowadzające test programu. Test powinien uruchomić program, podać na standardowe wejście poniższe komendy, a wyniki zapisać do pliku tekstowego results\_[suffix].txt:

1. Tworząca tablicę.
2. Zliczające słowa w każdym z plików źródłowych niniejszego zadania.
3. Wypisujące rezultaty zliczania.
4. Usuwające wszystkie wpisy po kolei.
5. Usuwający tablicę z pamięci.

Utwórz wpisy w *Makefile* uruchamiający testy z pkt. a w trzech wariantach programu

1. Z wykorzystaniem biblioteki statycznej
2. Z wykorzystaniem bilbioteki dzielonej (dynamiczne ładowanie podczas uruchamiania programu)
3. Z wykorzystaniem biblioteki ładowanej dynamicznie (dlopen())

Wyniki pomiarów zbierz w plikach results\_[suffix].txt (gdzie suffix to: static, shared, dynamic).  
Otrzymane wyniki krótko skomentuj. Wygenerowane pliki z raportami załącz jako element rozwiązania w pliku report.txt.  
  
**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 results\_[suffix1]\_[suffix2].txt.  
Otrzymane wyniki krótko skomentuj. Wygenerowane pliki z raportami załącz jako element rozwiązania w pliku report.txt.