

Alokator pamięci, wymagania na ocenę 4,0:

Celem jest rozbudowanie alokatora o następujące funkcjonalności:

```
void* heap_malloc_aligned(size_t count);  
void* heap_calloc_aligned(size_t number, size_t size);  
void* heap_realloc_aligned(void* memblock, size_t size);
```

Rodzina funkcji `heap*_aligned` (np. `heap_malloc_aligned`) wykonuje operacje funkcji bazowych `*` z ograniczeniem wymuszającym alokację nowych bloków **wyłącznie** na początku strony pamięci. W praktyce adres zwracany przez funkcje `_aligned` jest zawsze wielokrotnością długości strony pamięci i zaczyna się w jej zerowym bajcie.

Przyjmując założenie, że:

- strona ma długość `4096` bajtów (`0x1000`) oraz
- `ptr` to zmienna przechowująca wynik alokacji pamięci funkcjami `_aligned`

to następujące wyrażenie musi być **zawsze prawdziwe**:

```
((intptr_t)ptr & (intptr_t)(PAGE_SIZE - 1)) == 0
```

(sprawdza ono, czy 12 najmłodszych bitów adresu w `ptr` jest wyzerowanych).

Funkcje przydzielają pamięć na tej samej zasadzie, co ich zwykłe odpowiedniki.

Uwagi

- W tym zadaniu funkcja `main` nie jest testowana. Wykorzystaj ją do testów.
- Funkcja `custom_sbrk()` dostępna jest zarówno w raportach z kompilacji (Dante dołącza ją automatycznie) jak i w repozytorium GitHuba https://github.com/tomekjaworski/SO2/tree/master/heap_sbrk-sim.
- Nie używaj rzeczywistej funkcji `sbrk()`. W przypadku Biblioteki Standardowej `libc` za jej wykorzystanie odpowiada standardowa implementacja alokatora pamięci (znany już `malloc` itp). Ręczne uruchamianie `sbrk()` spowoduje desynchronizację informacji, posiadanych przez tę bibliotekę, i fizycznie przydzielonej pamięci. Uniemożliwi to poprawne działanie wszystkim funkcjom Biblioteki Standardowej, korzystającym z wbudowanego alokatora (np. `fopen`).
- Link do pliku CMake dla środowiska CLion: <https://pastebin.com/DGr27FLE>.

Przydatne informacje:

- <https://medium.com/@andrestc/implementing-malloc-and-free-ba7e7704a473>
- <https://danluu.com/malloc-tutorial/>

