

Programowanie w C projekt

Marcin Żurowski

4 czerwca 2025

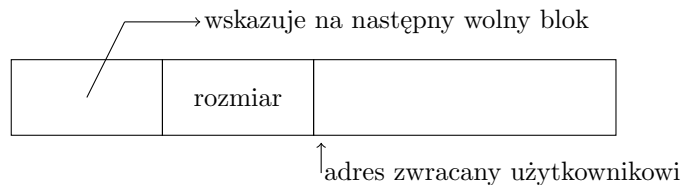
Napisz dwie funkcje które obsługują alokację pamięci:

- `void *malloc2(unsigned size);`
- `void free2(void * ap);`

Program powinien wykorzystywać następującą strukturę danych:

```
typedef long Align;
union header {
    struct {
        union header *ptr;
        unsigned size;
    } s;
    Align x;
};
typedef union header Header;
Header base;
```

Każdy blok danych powinien być reprezentowany przez nagłówek `Header` w następujący sposób:

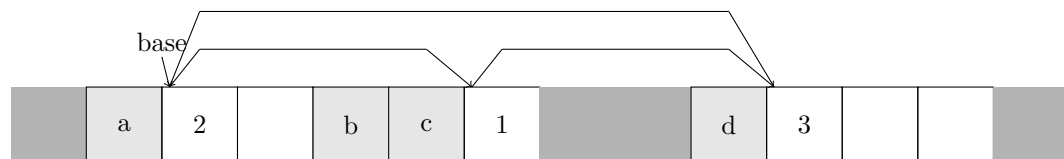


Blok ten powinien być alokowany za pomocą funkcji systemowej:

```
void *sbrk(int);
```

Ale w projekcie proszę użyć funkcji `malloc` z biblioteki `stdlib.h` (tak aby program dało się odpalić pod Windows).

Przykładowy schemat budowy pamięci:



Białe kwadraty, to wolna pamięć, jasnoszare kwadraty z literami to pamięć zajęta, ciemnoszare obszary pamięć niedostępna. Bloki są uporządkowane w kolejności rosnących adresów. Liczby określają wielkość bloku. Funkcje powinny spełniać następujące założenia:

- na początku `base` powinno być pusta struktura
- po wywołaniu funkcji `malloc2` powinien być za-alokowany obszar o łącznej wielkości minimalnej `1024 * sizeof(Header)`

- funkcja `malloc2` powinna zwracać adres pamięci o żądanej wielkości umiejscowiony na końcu bloku (na powyższym rysunku alokowanie obszaru o wielkości 1 powinno zwrócić adres białego bloku między etykietą 2 a etykietą *b*, natomiast etykieta 2 powinna się zmniejszyć o 1).
- wywołanie funkcji `malloc2(3)` dla powyższego przykładu powinno skutkować przeszukaniem całej struktury i zwróceniem adresu bloku z etykietą 3
- wywołanie funkcji `malloc2(4)` dla powyższego przykładu powinno skutkować przeszukaniem całej struktury i utworzeniem nowego bloku o odpowiedniej wielkości, podłączenie go do struktury i zwrócenie adresu obszaru o odpowiedniej wielkości
- wywołanie funkcji `free2` dla obszaru *c* powinno skutkować połączeniem tego obszaru z wolnym obszarem o wielkości 1 (i odpowiednią zmianą wielkości obszaru)
- wywołanie funkcji `free2` dla obszaru *b* powinno skutkować połączeniem tego obszaru z wolnym obszarem o wielkości 2 (i odpowiednią zmianą wielkości obszaru)