

Proiectarea sistemelor de operare – Proiect  
Implementarea bibliotecii mymalloc in C

Grupa C113E

Sd. Sg. Vasilache Cosmin

Sd. Sg. Orsan Emanuel-Tarcisiu

Sumar:

Implementarea unei biblioteci mymalloc in limbajul de programare C care sa aiba functionalitati similare cu cele ale functiei malloc() (din biblioteca stdlib).

Obiective:

Obiectivul proiectului este de a intelege principiile alocarii dinamice a memoriei si de a implementa o biblioteca(statica) eficienta care sa faciliteze aceasta tehnica de programare.

Cerinte:

1. Implementarea functiei void\* mymalloc(size\_t size) care are rolul de a aloca o zona de memorie in heap si de a returna un pointer catre inceputul zonei alocate.
2. Implementarea functiei void myfree(void\* ptr) care sa elibereze memoria alocata de functia mymalloc().
3. Crearea functiei myrealloc(void\* ptr, size\_t size) care redimensioneaza un block de memorie alocat de noi, facandu-i dimensiunea mai mare/mica. In acelasi timp functia se asigura ca datele sunt salvate si nemodificate(in cazul maririi blocului).
4. Realizarea unui mecanism eficient de gestionare a memoriei alocate(functiile first fit, best fit), care sa se asigure si sa previna suprascrierea blocurilor de memorie noi alocate/relocate in memoria heap, totodata, verifica si da exista suficient spatiu in memorie pentru a face alocarile.

Mod de implementare:

Vrem sa implementam o structura de date sub forma unei liste dublu inlantuite prin care putem sa navigam blocurile de memorie alocate pe heap

```
typedef struct data_block  
{  
    struct data_block* next;  
    struct data_block* prev;  
    size_t dim_bloc;  
    bool liber;
```

```
}data_block;
```

Asupra acestei liste vom face functiile merge(), realloc(), etc.

Totodata mai implementam un vector care sa pointeze catre blocurile libere(dealocate) din structura pentru a putea gasi si accesa blocurile libere din urma operatiei myfree().

```
static data_block* heap_liber[SIZE_HEAP];
```