Rapport projet RS 2015-2016

Par: **-BENJEBARA MAHA < maha.benjebara@telecomnancy.eu >**

-HILAL Nada < nada.hilal@telecomnancy.eu >

TELECOM nancy

Ingénieurs du numérique • Inspiring your digital future

UNIVERSITÉ DE LORRAINE

I) Introduction :

Dans ce projet de RS il nous été demandé de coder une librairie dynamique permettant l'allocation de la mémoire en c. Pour réaliserce projet on a utilisés les notions suivantes :

- les listes chainées en C
- mmap et sbrk

II)Réalisation :

mem_init (): Initialise les structures de données qui gèrent l'allocation dynamique de mémoire en utilisant mmap.

Mem _init (): initialise les structures de données qui gèrent l'allocation dynamique de mémoire en utilisant cmap.

On a cré des Node qui sont comme des cellules contenant pointeur sur next et previous, listes chaînées qui contiennent des nodes et puis une structure mem_manager qui contient deux liste schainées un pour les nodes libres et l'autre pour les nodes alloués .

void* Mem_alloc (taille unsigned int): se réserve octets de taille de l'espace de la région de mémoire créé par mem_init. Si la mémoire est réservée (alloué) avec succès, retourne un pointeur vers la mémoire réservée. Si la mémoire ne peut pas être réservé (il n'y a pas de bloc qui est assez grand pour contenir la taille octets), retourne NULL.

int Mem_Free (void * addr): Rendement mémoire allouée par mem_alloc à la liste des blocs libres, afin qu'il puisse être réutilisé plus tard.

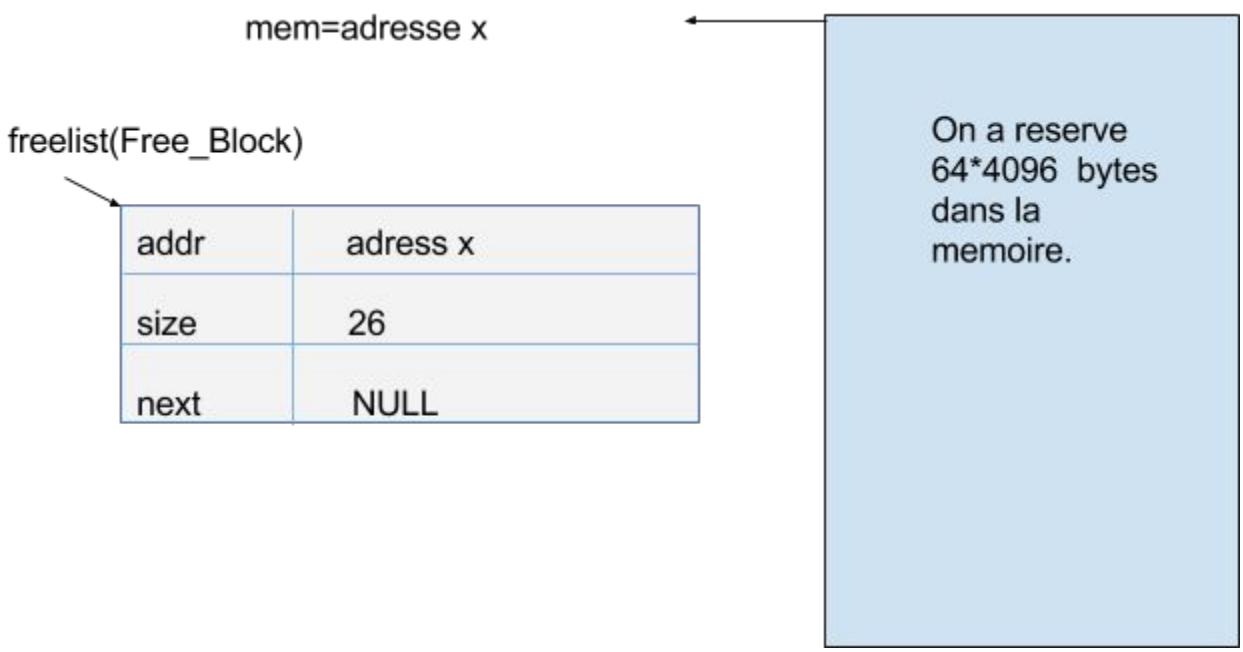
Il ya trois variables globales qui définissent les structures de données nécessaires:

-mem:stocke l'adresse de début de la région de mémoire qui est réservée par Mem_Init.

-freelist(Free_Block): Une liste liée de blocs struct qui identifient les parties de la région de mémoire qui sont libres (non utilisé). Blocs dans cette liste sont stockés dans l'ordre croissant des adresses.

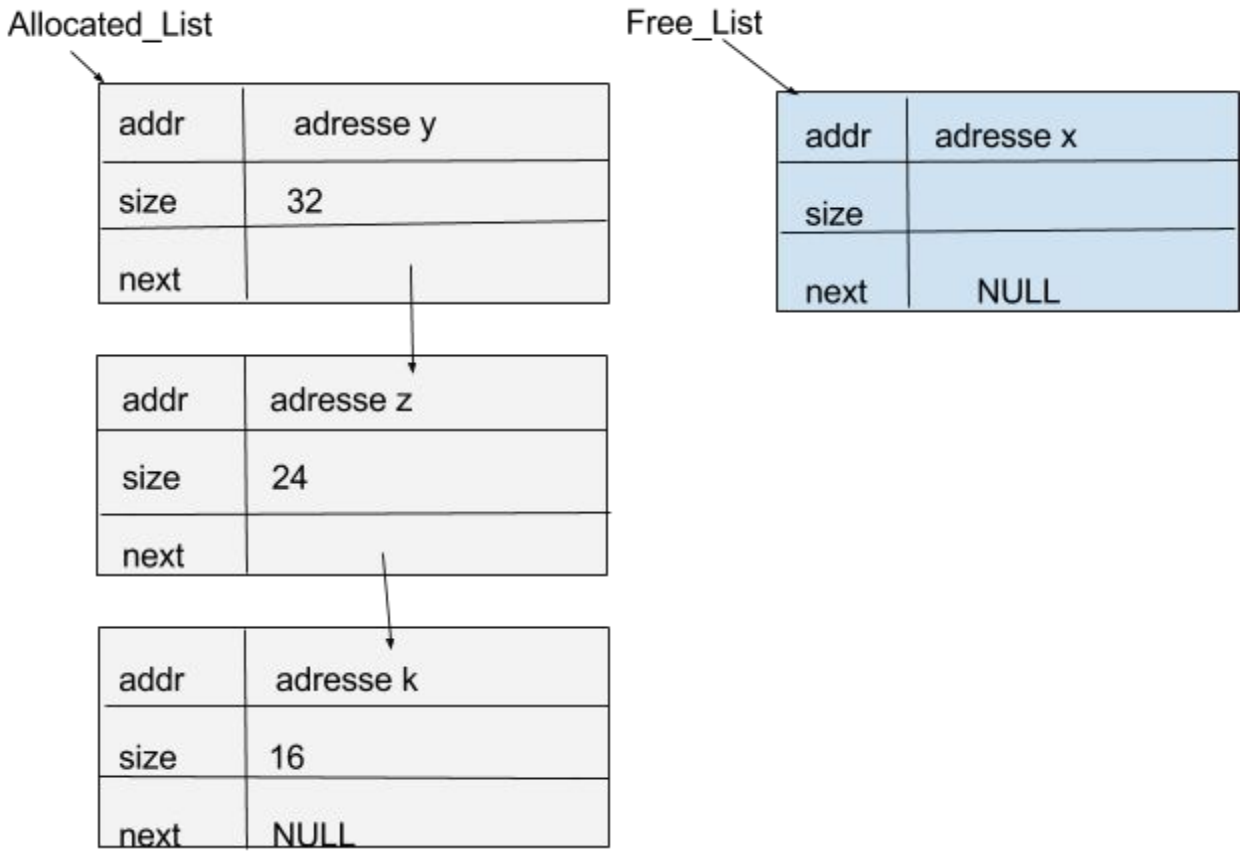
-allocated_list(Allocated_Block): Une liste liée de blocs struct qui identifient des parties de la mémoire qui ont été réservés par des appels à Mem_Alloc. Quand un bloc est alloué, il est placé à l'avant de cette liste, la liste est non ordonnée.

Explication de notre methode par des exemples:

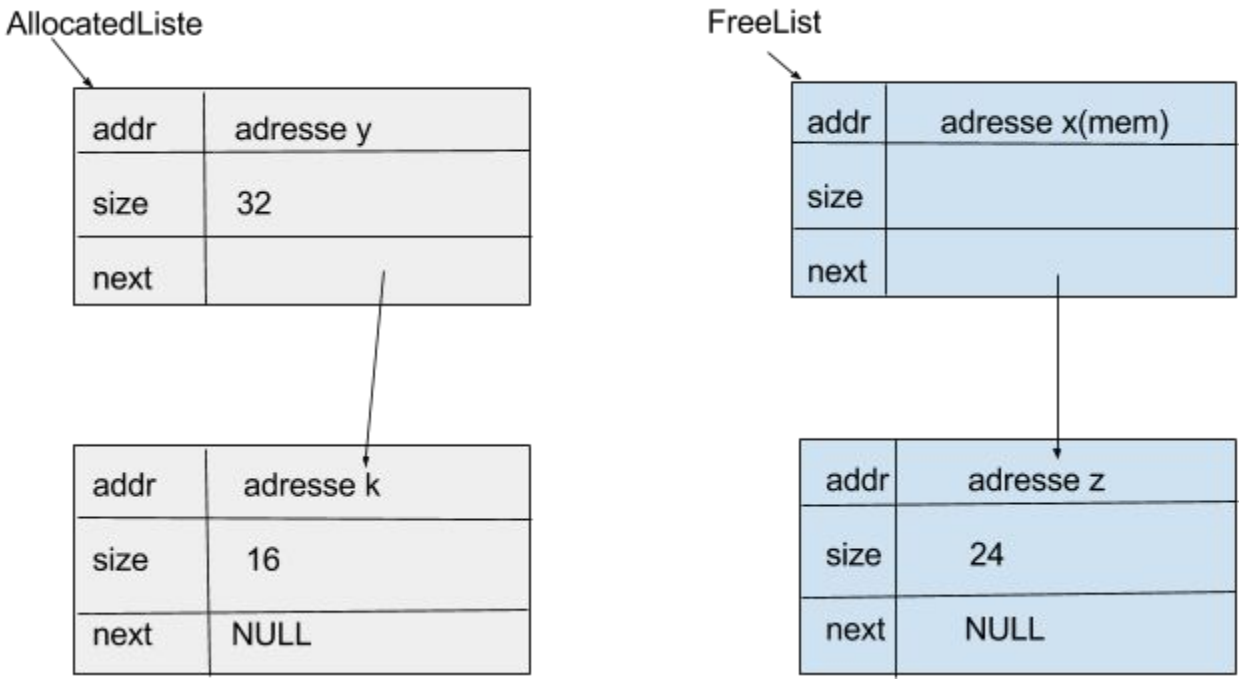


En premier lieu il y'a seule node dans la liste free qui contient le head de la liste allouée.

Quand appelle Mem_Alloc(32); Mem_Alloc(24); Mem_Alloc(16) :



Si apres On appelle par exemple Mem_Free(24):



III) Difficultés rencontrées dans la mise en oeuvre de ce projet:

-Trouver une méthode pour stocker les cases allouées et vu notre familarité avec les listes chainées en prepas au Maroc on a adopte la méthode des listes chainées.

-Comprendre le fonctionnement de mmap et sbrk.

-Comprendre le fonctionnement interieure de malloc et free.

-Ecrire toutes les fonctions, et manipuler les pointeurs.

IV)Repartition du temps de travail et de conception:

On a bien répartis les taches entre nous, nous avons travaillés ensemble depuis le début du projet.

La totalité de notre projet a été réalisée en un mois et trois semaines dont une semaine pour la recherche.

V)Remerciement:

Nous avons realisés ce projet en s'aidant des sites suivants:

-**http://www.cdf.toronto.edu/~csc209h/winter/assignments/a2/a2.html**

-**https://openclassrooms.com/courses/**

-Des projets publics sur **https://github.com/**