# TP n°4 allocateur mémoire

Mem_init : cette fonction nous permet d'initialiser la mémoire et de définir la mémoire disponible en un grand bloc libre ayant les paramètres définis dans notre struct fb.

Mem_fit_first : cette fonction nous permet de trouver le 1 er bloc libre disponible de taille suffisante pour effectuer notre allocation. On défini la variable taille qui correspond à la taille du bloc que l'utilisateur souhaite allouer. On parcourt la liste des blocs jusqu'à ce qu'on trouve un bloc de la taille demandée.

Allign (méthode supplémentaire) : On ajoute cette méthode pour obtenir des adresses qui se suivent et qui nous facilitent le traitement dans mem_alloc ou mem_free.

Mem_alloc : Cette méthode nous permet d'allouer un bloc mémoire de la taille désirée par l'utilisateur. On commence déjà par déterminer la taille nécessaire pour allouer un bloc (celle-ci comprend la taille demandée + la taille des métadonnées). On crée alors une zone de la taille demandée par l'utilisateur. Si on ne trouve pas de zone libre adaptée, on renvoie NULL.
Ensuite on redéfini la zone mémoire contenant les/la zone(s) libres(s) et la zone allouée. On vérifie d'abord qu'il y ait assez d'espace pour une nouvelle zone. On attribue une adresse et une taille à la zone que l'on a alloué et on chaîne les zones entre elles.

Mem_free : On commence par définir la zone mémoire à libérer, c'est-à-dire sa taille et son adresse. On crée une nouvelle structure qui va être la nouvelle zone libre ainsi que les deux zones précédant notre zone à considérer et la zone courante. Celles-ci vont nous permettre de nous repérer dans notre chaînage. On crée aussi 2 booléens indiquant si la zone suivante ou précédente est occupée ou libre. On parcourt donc notre liste chaînée jusqu'à la liste à considérer. On teste si les zones voisines sont libres et si oui on fait une fusion des zones.