Compte Rendu TP1 - CPS
LABBE Nicolas - RUZAFA Rémy

1) Nous n'avons pas besoin de gérer une liste de zones occupées puisque nous pouvons déterminer ces dernières en utilisant la liste de zones libres (En effet une zone est une zone occupée si elle se trouve entre deux zones libres).

2) La structure d'une zone libre ne contient que deux champs. le premier est la taille de la zone (les métadonnées sont comptées dedans) et un pointeur vers la prochaine zone libre.

En ce qui concerne les zones occupées, les métas données se situent au début de la zone (ici la taille des données allouées) puis lesdites données suivent et enfin le padding permet de compléter la zone. La fin d'une zone occupé se situe la ou commence la prochaine zone libre.

3) Les adresses 0,1 et 2 ne sont pas valide puisqu'elle sont inférieur a 8 (adresse minimal accessible par l'utilisateur). Ce dernier n'a accès qu'à des adresses dont la différence par 8 est multiple de l'alignement.

4) Lorsque l'on alloue une zone, l'adresse retournée à l'utilisateur est l'adresse du début de la zone occupé plus 8 octets. Car il n'est pas nécessaire de renvoyer la taille à l'utilisateur.

5) Si la taille de la zone libre créée après allocation est inférieure à la taille de la structure d'une zone libre alors la zone ainsi créée n'est pas utilisable et est donc intégrée au padding de la zone occupée précédemment créée.

Choix d'implémentation :

- Le premier choix que nous avons fait à été de choisir de positionner le pointeur de la liste des zones libres en variable globale.

- Le second choix à été celui de la signification de la taille des zones occupées. Nous avons décidé que cette dernière contiendra la somme de la taille totale des données à allouées, du padding et de 8 octet (le size_t taille). Ce choix est totalement arbitraire et n'a pas beaucoup d'impact sur la manière d'implémenter nos différentes fonctions.

- Nous avons ensuite choisi la position des métadonnées dans les zones occupées. Nous avons choisi de les mettre au début des ces dernière puisque cela simplifierait grandement le fonctionnement de free. En effet, pour free il nous suffit donc de récupérer l'adresse de la zone occupée de retirer 8 afin de connaître la taille de cette dernière et ainsi être capable de supprimer la zone en question.

1 : Les métadonnées (taille)
2 : Les données allouées
3 : Le padding

- Nous avons ensuite décidé de la valeur de l'alignement. Chaque taille de zone doit être multiple de cet alignement. Ainsi nous avons choisi de prendre comme valeur la taille de la structure de zone libre. En effet une zone ne peut être une zone libre si sa taille est inférieure à cette valeur. Ainsi une zone libre trop petite sera ajouté au padding de la zone occupée créée (cf question 5).

Tests réalisés :

- Pour la fonction malloc il a fallu tester tout les cas possible. Il y a d'abord deux gros cas à séparer. Les cas où la zone libre à allouer est la première, et les cas où c'en est une autre. Pour ces deux possibilités il y a trois cas. Tout d'abord, le cas où il n'y a pas de zone libre suivante. Ensuite, le cas où la zone à allouer n'a pas assez d'espace pour créer une nouvelle zone libre, et enfin le cas où elle a assez de place.

- Pour la fonction show, nous avons observé si ce qui était affiché correspondait bien avec ce qui était attendu. Nous avons donc effectué de nombreux tests dans le but d'observer si tout se passait bien.

- Pour la fonction free il nous a fallu tester si il était bien possible de libérer n'importe quelle zone occupée, et si la fusion entre zones libres adjacentes s'effectuait bien.

Limites :

Notre implémentation de l'allocateur possède tout de mêmes des limites. Tout d'abord, si l'utilisateur essaye de libérer une zone qui n'a pas été allouée par notre allocateur, cela va entraîner une perte de toutes les données de la mémoire, puisque celles ci ne seront plus accessibles.

De plus, la méthode de choix de zone libre n'est pas la plus optimale, puisqu'il arrive que de la zone mémoire ne soit pas utilisable, et donc ajoutée au padding d'une zone occupée. L'utilisateur n'a pas conscience de l'existence de ce padding, tout cet espace mémoire est donc complètement perdu, tant que la zone n'est pas libérée.