

## Gestion d'un tas

1

## Gestion de la mémoire d'un programme

- Variable globale : allocation statique
- Paramètres et variables locales : gestion en pile
- Allocation dynamique
  - malloc, free
  - new, delete

2

## Besoins

- Besoins: gérer l'occupation d'une plage de mémoire (éventuellement extensible)
  - réserver (allouer) des zones d'une taille spécifiée (malloc)
  - libérer une zone préalablement réservée (free)
- Critères:
  - rapidité des opérations d'allocation et de libération
  - surcoût en mémoire
  - gaspillage (mémoire inutilisée sans raison)

3

## Qui gère le tas ?

- Géré par les bibliothèques standards (malloc/free),
- voire par le langage lui même (new/delete),
- parfois de manière implicite (ramasse-miettes).

4

## Carte de bits

- le tas est subdivisé en blocs de taille fixe
- on établit une carte: un tableau de bits dont chaque bit représente l'état (libre/alloué) d'un bloc

5

## Evaluation

- Avantages
  - surcoût en mémoire raisonnable
- Inconvénients
  - allocation coûteuse en temps (recherche d'espace libre)
  - Surdimensionnement possible des blocs : certains blocs marqués comme occupés ne sont pas totalement utilisés
- Peu utilisé en pratique

6

## Listes chaînées

- On représente l'état de la mémoire par une liste chaînée dont chaque maillon représente une zone (adresse, taille), son état (alloué ou libre) et pointe vers le maillon représentant la zone suivante.

7

## Evaluation

- La durée des opérations et le surcoût en mémoire dépendent du nombre de zones allouées, et non de la taille du tas.
- Pas de fragmentation interne.
- Permet de gérer des zones allouées contigües.
- L'allocation et la libération sont encore relativement coûteuses (recherche dans la liste, y compris pour la libération).

8

## Choix de la zone allouée

- Lors d'une allocation mémoire, si plusieurs zones libres de taille suffisante sont disponibles, laquelle choisir?
  - Premier trouvé (first fit) : méthode la plus simple et la plus rapide
  - Meilleur trouvé (best fit) : cherche la zone libre la plus petite, c.à.d. la plus proche de la taille recherchée.
  - Pire trouvé (worse fit) : cherche la zone libre la plus grande.
- Quel est le meilleur choix ?

9

## Méthode Buddy

- Principe
- Méthode basée sur le principe dichotomique, permettant de diviser la taille du tas par des puissances de 2.
- Inconvénients
  - fragmentation interne (contrainte sur la taille des zones)
- Avantages
  - efficace en temps (logarithmique par rapport à la taille de la mémoire)

10

## Ramasse miettes

- La gestion dynamique est une tâche fastidieuse et une source d'erreur (« fuites mémoires ») pour les programmeurs.
- Ceux-ci utilisent par ailleurs des outils informatiques pour les détecter et les corriger.
- Ne pourrait-on pas laisser l'ordinateur gérer la libération de mémoire tout seul?

11

## Compteur de références

- Chaque objet possède un compteur de références indiquant combien de liens pointent vers lui.
- Chaque qu'on crée ou que l'on brise un lien vers un objet, son compteur de références est mis à jour.
- Lorsque son compteur de références atteint 0, l'objet est libéré, et ses liens sont brisés.
- Le comptage des références n'est pas suffisant.

12