



Sujet : simulation de la gestion de la mémoire centrale.

1. Présentation de la problématique :

Il s'agit de simuler la gestion de la mémoire centrale sollicitée par tout processus utilisateur qui a besoin d'exécuter son programme.

- Initialement, cette mémoire forme un seul bloc et est supposée vide (ne pas tenir compte de la partie réservée au système d'exploitation).
- Elle admet une **taille** maximale (physique non extensible) et est désignée **libre** par un **flag**.
- Au fur et à mesure des demandes d'allocation de la mémoire cette taille diminue jusqu'à l'impossibilité de répondre favorable à d'autres demandes d'allocation s'il n'y a pas eu de libération (exemple de malloc en langage C ou de création new d'objets en Java ou en C++...).
- Les demandes de libération peuvent être faites à tout moment et de façon aléatoire.
- A un instant donné, la mémoire peut être vue comme un tableau de segments **alloués/libérés** alternativement. Plusieurs cas peuvent se présenter. Il s'agit de la vision **statique** du tableau !!!!

2. Travail demandé :

- Une analyse rapide du problème avec possibilité de schémas illustratifs.
- Une présentation succincte de la solution au problème (structures algorithmiques, découpage modulaire, les classes...)
- Codage en Java (lisibilité et commentaires)
 - ⇒ **Contrainte1 : implémentation dynamique à l'aide de collections.**
 - ⇒ **Contrainte2 : Gestion des erreurs possibles.**
- Une conclusion sur les problèmes rencontrés et les améliorations possibles...

Précisions très importantes :

- ✓ Ce TP est à rendre pour le **vendredi 31 mai** au soir (**avant minuit**).
- ✓ Organisez vous en binômes.
- ✓ Le code est à envoyer à l'adresse mail suivante : coulmi.ges@gmail.com

Critères d'évaluation du TP GRAM

→ Le TP à rendre sera noté sur 20 :

- une note sur 6 affectée à l'**analyse** faite du problème et les solutions proposées (découpage fonctionnel).
- une note sur 14 affectée au **codage** en Java.

→ La note est généralement commune au binôme mais on se garde le droit de donner des notes individuelles si un membre du binôme fait preuve d'une aptitude supérieure ou inférieure par rapport à l'autre membre.

→ Les critères d'évaluation de l'analyse :

- La synthèse, la précision, la rigueur, illustration schématique,...
- Les erreurs (français et autres...), similarité des textes....

→ Les critères d'évaluation du code :

1. Qualité du code :

- lisibilité.
- commentaires +++
- modularité.

2. Implémentation

- minimum de concepts,
- concepts du cours,
- dépasse le cadre du cours.

3. Démontrabilité

- minimaliste,
- correcte répond au cahier des charges (besoins fonctionnels),
- au delà du cahier des charges.

→ Concepts objets mis en œuvre :

- **notions de base** : classe, constructeurs, méthodes, visibilité, héritage.
- **notions autres** : surcharge de méthodes, composition de classes, les exceptions...