

Projet N°1 de CPOO : Profiling de mémoire pour Lisaac

Travail demandé

Le travail demandé comporte une modélisation UML et une implantation en Java ou Lisaac. Un rapport court (2 ou 3 pages) devra accompagner les diagrammes UML en format papier. Ce rapport devra

- expliquer vos choix de modélisation et lever les ambiguïtés du sujet
- présenter la répartition du travail

Modélisation UML

Le nombre et le type de diagramme UML à concevoir est laissé à votre appréciation, mais cet ensemble de diagrammes devra d'une part modéliser correctement le système décrit ci-dessous, et d'autre part être aisément compréhensible par le correcteur. Votre modélisation UML devra comporter au minimum :

- Les diagrammes d'héritages
- Les diagrammes des classes
- Un diagramme d'états-transitions

Implantation

- Votre programme devra utiliser le langage Java ou Lisaac (au choix), et devra fonctionner sur la machine *turing*.
- L'interface utilisateur est textuelle (ligne de commande) ou graphique.
- Une petite explication de l'utilisation de votre programme est aussi nécessaire.

Remise du projet

La remise du projet se fera en deux temps :

- Le rapport et la modélisation UML du projet est à rendre **sur papier** lors de la dernière séance de TD (**semaine du 26 Novembre**).
- La programmation qui découle de votre modélisation est à **rendre pour le 04 Janvier**. La remise de cette partie du projet se fera électroniquement par mail à votre responsable de TP, à savoir :
 - L3-PF G1 & G3, L3-RIA : Benoit Sonntag (sonntag@icps.u-strasbg.fr)
 - L3-PF G2 & G4 : Claire Baegert (baegert@dpt-info.u-strasbg.fr)

Réalisation du projet

- La réalisation de ce projet devra se faire impérativement **par groupe de deux ou trois** pour que vous vous répartissiez le travail au sein du groupe.
- Comme tout cahier des charges, celui-ci ne peut être exhaustif. En cas d'ambiguïté, préciser votre interprétation personnelle, et éventuellement les questions à poser à votre interlocuteur (responsable de projet, futurs utilisateurs, etc.). Toute solution cohérente, justifiée et non contradictoire avec le cahier des charges sera acceptée.

Sujet

Dans le cadre du développement d'un nouveau langage de programmation objet, nous vous proposons d'élaborer un outil permettant d'illustrer et de mettre en lumière le système d'allocation dynamique et de libération automatique de la mémoire.

Cet outil a pour objectif d'observer par une représentation graphique et temporelle les allocations, les libérations et les accès mémoire.

L'objectif de cette observation est de tester la véracité d'un nouvel algorithme de ramasse-miette (*Garbage Collector*).

1 Production du fichier de sortie

1.1 Installation du compilateur Lisaac adapté

Vous trouverez à l'adresse http://isaacproject.u-strasbg.fr/li/li_download.html le fichier `lisaac_memory.zip` représentant une version du compilateur Lisaac adapté au besoin du projet.

1.2 Production d'un exécutable adapté

Après installation de celui-ci, à l'instar des systèmes de *profiling* disponible sur le compilateur `gcc`, vous utiliserez l'option `-p` pour activer la production d'un fichier de *profiling* mémoire.

La compilation d'un programme utilisant l'option `-p` a pour conséquence la production d'un exécutable particulier. Cette exécutable, lors de son exécution, produira un fichier contenant les informations concernant la mémoire en plus de son exécution normal.

Ce fichier aura comme nom, le nom de l'exécutable, et comme extension `.mem`.

Par exemple :

```
sonntag@Isaac:~/lisaac/src$ lisaac lisaac -t compiler -p
```

2 Informations utiles sur la gestion mémoire en Lisaac

En interne, nous distinguons 3 formes d'allocation objet :

- 1 Les objets ayant un type dynamique de taille fixe : Ce type d'objet est potentiellement appelé sur des sites polymorphiques. Leur type sera identifié par un numéro unique dans cette catégorie [1]. La taille de ce type d'objet est fixe.
- 2 Les objets ayant un type statique à taille fixe : Ce type d'objet est toujours appelé sur des sites monomorphiques. Leur type sera identifié par un numéro unique de catégorie [2]. La taille de ce type d'objet est fixe.
- 3 Les objets ayant une taille variable : Ce type d'objet est toujours appelé sur des sites monomorphiques. Seule la taille sera une information nécessaire en la description de ce type d'objet.

Trois informations seront pertinentes à notre analyse :

- L'allocation d'un objet de type [1], [2] ou [3].
- Libération d'un objet de type [1], [2] ou [3].
- Accès en lecture ou écriture à un objet.

3 Description du fichier de sortie `.mem`

Le fichier de sortie est composé de deux parties :

1. Description des types.
2. *Profiling* de la mémoire.

3.1 Description des types

Un exemple de type :

```
T1 4 STRING 16
```

En premier lieu, nous avons la catégorie de T1 ou T2 (voir 2). Le type 3 n'est pas défini. Le type est suivi d'un identifiant représenté par un numéro unique pour un type donné (ici, l'identifiant est 4). Ensuite, nous avons le nom complet du prototype (ici, STRING). Puis, la taille de la structure de l'objet en octets (ici, 16 octets).

3.2 Profiling de la mémoire

3 types d'informations sont possibles :

1. type RW : Accès en lecture ou écriture à une adresse donnée.
2. type Ax : Allocation d'un objet de type x.
3. type Fx : Libération d'un objet de type x.

Pour les allocations et libérations de types 1 et 2, nous avons l'identifiant de l'objet suivi de l'adresse d'allocation ou de libération.

Pour les allocations et libérations de types 3, nous avons le nombre d'octets nécessaires suivi de l'adresse d'allocation ou de libération.

Les adresses sont toujours en hexadécimal.

3.3 Exemple de fichier de sortie `example.mem`

```
T1 4 STRING 16
T2 0 ARRAYED_COLLECTION[STRING] 12
T2 1 FAST_ARRAY[STRING] 12
T2 2 MEMORY 16
T2 5 ITM_OBJECT 4
T1 5 LOCAL 44
A1 4 #80140120
RW #80140124
RW #80140120
A1 4 #80140130
A2 1 #80140140
A3 32 #80140240
A2 5 #8014014C
A2 2 #80140150
F1 4 #80140120
A1 4 #80140120
```

4 Cahier des charges du produit

4.1 Détection des incohérences

Votre outil doit détecter un certain nombre de cas d'erreur de mon allocateur de mémoire, à savoir :

- Chevauchement d'allocation mémoire : L'allocation d'un objet sur une plage mémoire déjà occupé ou partiellement occupé.
- Accès en mémoire d'une adresse non alloué.
- Libération d'un bloc de mémoire non alloué.
- Libération d'un bloc de mémoire non aligné sur un objet.

4.2 Représentation graphique de la mémoire

Nous représenterons la mémoire sous forme d'une frise (comme une frise chronologique) ayant pour abscisse les adresses mémoire. Cette frise sera composée de bandes de couleurs.

La couleur ayant la signification suivante :

- **Vert clair** : Zone libre jamais allouée.
- **Vert foncé** : Zone libérée, mais ayant eu déjà une allocation.
- **Rouge vif** : Zone allouée récemment.

- du **rouge vif** au **noir** : Dégradé de nuances donnant une indication sur la date du dernier accès en mémoire de cet objet (**Rouge vif** : Objet récemment créé ou accédé en lecture/écriture, **Noir** : Objet anciennement accédé)

Nous devons pouvoir avoir une vision d'ensemble de la mémoire par une frise globale à l'écran. Mais, nous devons aussi pouvoir zoomer une partie de la frise, nous permettant de visionner avec précision l'adresse d'un objet et le nom de l'objet pointé par le curseur souris sur la frise ainsi zoomé.

D'autres informations sont aussi possibles (optionnel), comme le taux d'accès ou l'écart entre la 'date' actuelle et la 'date' du dernier accès (voir la notion de 'date' en 3.3)

4.3 Représentation temporelle de la mémoire

Il est nécessaire de pouvoir visionner, comme un film, les différentes actions réalisées sur la mémoire. Ici, nous n'avons pas d'information exacte sur la date d'une action. Mais, nous avons une relation de précedence relative (action 1, puis action 2, ...). Nous pouvons donc avoir une date relative par rapport à l'ensemble des actions disponible dans le fichier.

Nous devons avoir le choix en la vitesse d'exécution de ce film. Aussi, nous souhaitons avoir la possibilité de réaliser des pauses, de revenir en arriere, ou d'aller en un point précis.

5 Simplification et précision du sujet

- Le zoom peut être disponible qu'en état de pause.

Good luck !