

PROJET DE LA PREMIÈRE PARTIE DU COURS
COMPILATION AVANCÉE
MASTER STL - M1
2012-2013

Il vous est demandé de réaliser un GC pour remplacer celui utilisé en ILP (Implantation des Langages de Programmation).

Garbage Collecteur Mark&Sweep pour ILP

Ce projet a pour but d'écrire un nouveau gestionnaire mémoire pour les programmes exécutés en ILP que vous avez suivi au 1er semestre. Ce GC est à écrire en C. Il remplacera celui utilisé au 1er semestre dans la bibliothèque d'exécution d'ILP.

Représentation des données Selon les épisodes d'ILP les données sont représentées soit par des valeurs allouées (épisode ILP1) soit par des objets (ILP6). On cherchera à implanter le GC pour ces 2 cas.

1. épisode ILP1 : Toute valeur de la bibliothèque d'exécution est une structure allouée comprenant un type et un contenu (voir transparent 108 du fichier ilp1-4.pdf) comme ci dessous : .

```
typedef struct ILP_Object {
    enum ILP_Kind      _kind;
    union {
        unsigned char asBoolean;
        int           asInteger;
        double        asFloat;
        struct asString {
            int      _size;
            char     asCharacter[1];
        } asString;
        struct asPrimitive {
            ILP_general_function _code;
        } asPrimitive;
        ILP_OTHER_STRUCTS
    }
    _content;
} *ILP_Object;
```

2. épisode ILP6 où tout est objet :

```
typedef struct ILP_Object {
    struct ILP_Class* _class;
    union {
        unsigned char asBoolean;
        int           asInteger;
        double        asFloat;
        struct asString {
            int      _size;
            char     asCharacter[1];
        } asString;
        struct asException {
            char      message[ILP_EXCEPTION_BUFFER_LENGTH];
            struct ILP_Object* culprit[ILP_EXCEPTION_CULPRIT_LENGTH];
        } asException;
        struct asClass {
            struct ILP_Class* super;
            char*             name;
            int               fields_count;
        }
    }
};
```

```

        struct ILP_Field*   last_field;
        int                 methods_count;
        ILP_general_function method[1];
    } asClass;
    struct asMethod {
        struct ILP_Class*   class_defining;
        char*               name;
        short               arity;
        short               index;
    } asMethod;
    struct asField {
        struct ILP_Class*   defining_class;
        struct ILP_Field*   previous_field;
        char*               name;
        short               offset;
    } asField;
    struct asInstance {
        struct ILP_Object*  field[1];
    } asInstance;
}
        }
        _content;
} *ILP_Object;

```

Ainsi dans les deux cas toutes les valeurs, y compris les entiers où les nombres à virgule flottante, sont allouées. Il n'y aura pas besoin de distinguer les pointeurs des valeurs immédiates. Ici tout est pointeur, et le type de la valeur pointée est le premier champ de la structure.

algorithme de GC : Comme la bibliothèque d'exécution est en C, il est difficile d'utiliser des GC déplaçant les valeurs. Pour cela il est demandé d'implanter un algorithme Mark& Sweep comme vu au cours 4 et aux TD/TP correspondants.

Rendu

Il est demandé un rapport clair et concis (moins de 10 pages sans les annexes) présentant la problématique du sujet et en guidant le lecteur par un exemple pertinent. On peut ajouter des annexes à ce document (en particulier les listings ou des exemples plus complets).

Evaluation

L'évaluation du projet comprend :

- le rapport du projet
- une archive (tgz) de l'implantation et des jeux d'essai

La note du projet sera commune au binôme.

Les rapports sont à rendre au plus tard le jeudi **21/03/2013** à 16h par courrier électronique à Benjamin.Canou@lip6.fr, Emmanuel.Chailloux@lip6.fr, Philippe.Trebuchet@lip6.fr en précisant l'objet suivant ca-2013fev.