# Projet avancé : étude d'*Interim OS*

Marc Ducret  Florentin Guth

9 janvier 2017

# Différents niveaux

## 2 couches

*Interim OS* est scindé en deux parties : un partie permettant de gérer précisément la mémoire, écrite en *C*, et une partie contenant la gestion des différents programmes (comme un éditeur, une console, . . . ).

# Fonctionnement général

La partie gérant la mémoire fonctionne de la manière suivante :

- On lit l'expression *Minilisp* donnée,
- On formate cette expression en une représentation plus structurée pour faciliter la compilation,
- On produit du code assembleur correspondant à l'exécution de l'expression donnée, que l'on stocke dans un fichier temporaire,
- Le code produit utilise des fonctions spéciales d'allocations qui permettent de faire fonctionner le ramasse-miettes pour libérer la mémoire lorsque c'est nécessaire,
- On exécute ce code assembleur,
- On affiche le résultat (qui est une valeur *Minilisp*).

# TODO

Si tu pouvais me rédiger (au moins un brouillon) cette partie, ce serait pas mal parce que je pense que tu connais mieux que moi.

# Structure de données

minilisp.h[81-91]

```
81 typedef struct Cell {
82   union ar {
83     jit_word_t value;
84     void* addr;
85   } ar;
86   union dr {
87     jit_word_t size;
88     void* next;
89   } dr;
90   jit_word_t tag;
91 } Cell;
```

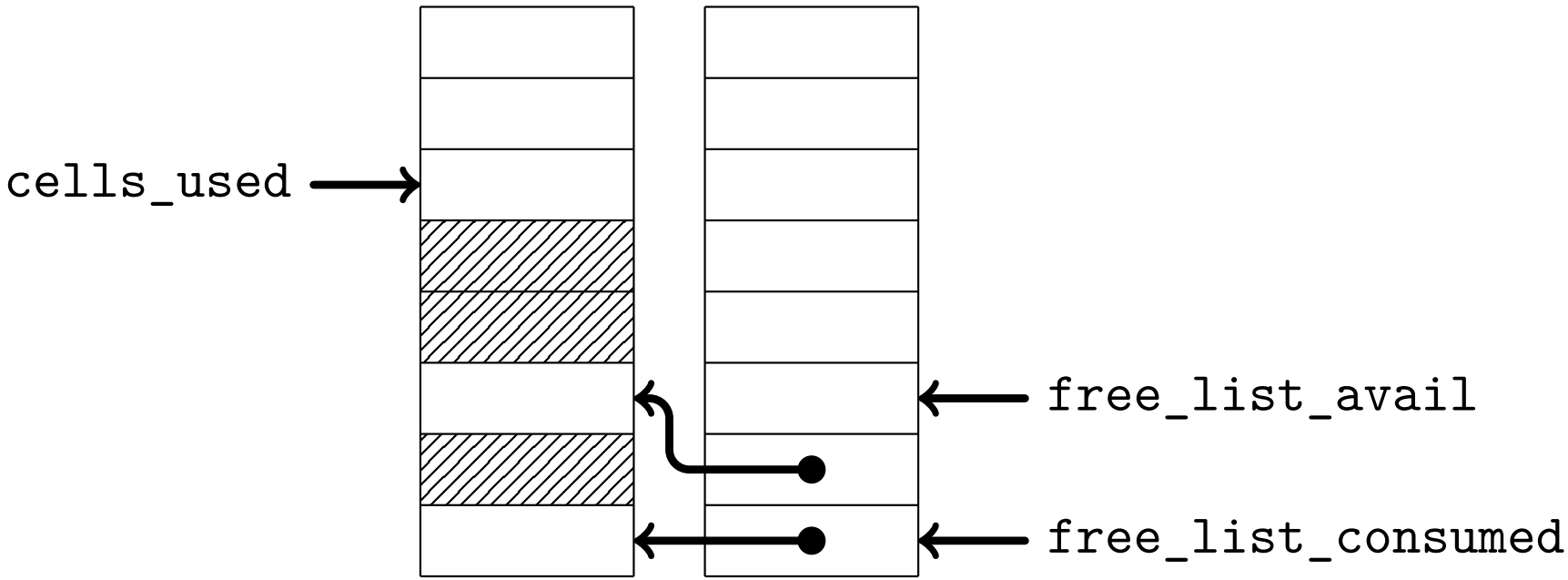
Listing 1 – La `struct Cell`

# Avantages

Cette représentation a plusieurs avantages :

- Elle permet d'effectuer facilement (en théorie . . . ) la *garbage collection*, car en fonction du tag on sait si on a affaire à des entiers ou des pointeurs,
- Toute les valeurs ont la même taille (sans compter les cellules pointées par l'un des membres), et il n'y a pas de tableau, ce qui simplifie la gestion des blocs en mémoire.

# Schéma de la mémoire



# GC

*Interim OS* utilise un *GC mark-and-sweep* classique.

**GC mark-and-sweep**

Un *GC mark-and-sweep* est un *GC* qui parcourt en profondeur le graphe des `Cells` en marquant les nœuds rencontrés. Dans une deuxième phase, on libère (*sweep*) les nœuds non visités en parcourant l'intégralité du tas.

Les racines sont la pile, les fichiers ouverts et l'environnement global.

# TODO

# TODO

# Truc