

TP n°4

Pour avancer dans le compilo

Types, Symbols, Intermediate Representation and Frames

Florent BOUCHEZ, Christoph LAUTER
prenom.nom@ens-lyon.fr

21 mars 2007

Maintenant que vous avez bien terminé les analyses lexicales et grammaticales, il est temps de réfléchir à comment continuer le compilo.

1 Des symboles occultes

Ben oui, on va avoir besoin de tables de symboles. Commençons donc pour s'échauffer par créer le `symbol.ml` correspondant à

```
/home/fbouchez/compilation/TP4_avanti/symbol.mli
```

2 Des étiquettes temporaires

Les temporaires sont de drôles de bêtes qui sont capables de contenir des valeurs. Ils correspondent à des emplacements mémoire et plus tard, ils seront alloués si possible à des registres, sinon à des endroits dans la pile ou le tas. Les labels correspondent à des pointeurs vers des emplacements mémoire, par exemple vers une instruction (utile pour faire un `goto label`), ou vers une chaîne de caractère (pour pouvoir l'afficher). Continuons l'échauffement par la création de `temporary.ml` correspondant à

```
/home/fbouchez/compilation/TP4_avanti/temporary.mli
```

3 De drôles de types

Parce qu'on aime bien les choses un peu propres, on va se faire dans un coin un petit fichier de types qui nous sera bien utile par la suite ; par exemple quand il faudra taper sur cet `@#$$!` de programmeur qui s'est encore planté de type dans ses expressions. De même, faire ce qu'il faut avec

```
/home/fbouchez/compilation/TP4_avanti/types.mli
```

4 Au milieu, on se bat avec qui ?

Dans notre grande bonté, on vous a préparé une petite représentation intermédiaire tout belle pour que vous ne vous cassiez pas la tête.

```
/home/fbouchez/compilation/TP4_avanti/ir.mli  
/home/fbouchez/compilation/TP4_avanti/ir.ml
```

Passez un peu de temps à la découvrir et a en admirer la beauté avant de passer à la suite.

5 Et même des cadres à régler

Là ça rigole moins, voire plus du tout. On rentre dans le cœur de la bête. Ici, on fait déjà un choix, celui de l'architecture finale. Enfin, dans

```
/home/fbouchez/compilation/TP4_avanti/frame.mli
```

peu de choix sont définitivement faits. Il vous faudra les faire vous-même dans `frame.ml` en fonction de vos préférences, sachant qu'historiquement le `frame.ml` associé était prévu pour du `x86`. Néanmoins, si vous vous sentez le courage de cibler du `sparc`, `powerpc` ou même du `VLIW`, à vos risques et périls.