Master S&T - Mention Informatique
$1^{ère}$ année
2015 – 2016
Compilation

TP n° 1

Principe de compilation

Objectif :

- Observer et comprendre l'utilisation de la pile pour réaliser les passages de paramètres et pour représenter les variables locales et les paramètres passés par valeur.
- Observer et comprendre l'implémentation de la règle de portée statique des identificateurs.
- Observer et comprendre la représentation des structures de données de type tableau.

Travail à faire : Réaliser les exercices suivants sur machine et rédiger un compte rendu répondant aux questions des exercices. Ce compte-rendu est à déposer sous moodle pour lundi 28 Septembre.

Exercice 1 : Base

Observer l'exécution pas à pas du programme *inc.c* suivant :

```
void inc(int x, int * y) {
  int l;
    l = x;
  l = l + 1;
  (*y) = l;
}
int main() {
  int x;
  int z;
  x = 255;
  inc(x,&z);
  return 0;
}
```

Exercice 2 : Factorielle

Observer l'exécution pas à pas du programme *fact.c* suivant :

- Dessiner l'état de la pile d'exécution pour les appels de f(4,&x) et f(0,y) après le `(*y) = 1;` commenter ce schéma :
  - en découpant la pile en blocs qui représentent les environnements d'exécution de chaque appel,
  - en expliquant de quoi est composé un bloc
- Expliquer comment s'effectue l'affectation `(*y) = (*y) * x`.

```
void f(int x, int *y) {
  if (x == 0) (*y) = 1;
  else {
       f(x-1,y);
       (*y) = (*y) * x;
  }
}
int main(void) {
   int x;
   x = 4;
   f(x,&x);
   return 0;
}
```

Exercice 3 : Tableau

Observer l'exécution pas à pas du programme *tab.c* suivant :

- Comment sont représentés les tableaux `t1` et `t2` en mémoire.
- De façon générale, comment peut être réalisée l'affectation `t2 = t1` ? recopie case à case ? ou recopie de pointeur ?
- De façon générale, comment sont réalisés les passages de paramètres par valeur et par adresse d'un tableau ?
- Comment sont calculées les cases pour obtenir l'élément y figurant ?

```
void f(int t[5], int * y) {
   int j;
   t[1] = 256;
   for (j = 0; j < 5; j++)
      y[j] = t[j] + 1;
}
int main() {
  int t1[5];
  int t2[5];
  int j;
  t1[1] = 1;
  for (j = 0; j < 5; j++)
     t1[j] = j;
  f(t1,t2);
  return 0;
}
```

Exercice 4 : Portée

Observer l'exécution pas à pas du programme *portee.c* suivant :

- Relativement à l'exemple *Factorielle* ? Quelle information nouvelle figure dans la pile quand f et g s'exécutent.
- Comment est réalisé l'accès à la variable globale y (exemple : accès à y depuis f ou r dans g) ?

```
int y;
void g(int d, int *r) {
   int y;
   y = (*r);
   (*r) = d+1;
}
void f(int d, int * r) {
   int x;
   x = d+1;
   y = 1;
   (*r) = x;
   g((*r),r);
}
int main() {
  int x;
  x = 3;
  f(x,&y);
  return 0;
}
```