Compilation — TD 2

J. Chroboczek et P. Letouzey

11 Octobre 2006

1. On considère le programme « C » suivant :

```
int main()
{
    int i, j;
    for(i = 0; i < 10; i++) {
        j = 3 * i + 2;
        printf("%d\n", j * j);          /* (1) */
    }
    exit(0);
}
```

On appelle *branchement* une conditionnelle de la forme

```
if(e) goto l;
```

c'est à dire une conditionnelle dont la conséquence est une instruction de saut et dont l'alternative est vide.

1. Éliminez les boucles de ce programme en convertissant les `for` en `while`, puis les `while` en branchements.
2. Convertissez le programme ainsi obtenu en code à trois valeurs, puis affectez des registres temporaires à toutes les variables.
3. Traduisez le programme résultant en assembleur MIPS et exécutez-le avec SPIM.
4. Même exercice après avoir remplacé la ligne (1) par
   `if(i % 2 != 0) printf("%d\n", j * j);`

2. Allocation de données

On considère le programme « C » suivant :

```
int main()
{
    int a[10];
    int i, j;
    a[0] = 1;
    a[1] = 1;
    for(i = 2; i < 10; i++)
        a[i] = a[i - 2] + a[i - 1];
    for(j = 9; j >= 0; j--)
        printf("%d\n", a[j]);
```

```
    exit(0);
}
```

2.1 Allocation statique

On va d'abord allouer le tableau `a` statiquement[1], comme ce serait le cas si on le définissait à l'extérieur de la fonction `main` ou si on utilisait le mot clef `static` en « C ».

1. Écrivez le code assembleur qui alloue 40 octets de mémoire dans la section `.data` et lui donne l'étiquette `a`. (On pourra utiliser soit la pseudo-instruction `.word` suivie de 10 zéros, soit la pseudo-instruction `.space 40`. Il faudra en outre penser à utiliser `.align`.)
2. On rappelle que la donnée `a[i]` réside à l'addresse $\texttt{a} + 4\texttt{i}$. Convertissez le programme précédent en code à trois valeurs et traduisez-le en assembleur MIPS.

2.2 Allocation sur la pile

1. Modifiez le programme précédent pour qu'il alloue 40 octets en décrémentant `$sp` de 40, et qu'il accède à cette mémoire à travers `sp` au lieu de la zone statique `a`. Vous n'omettrez pas de restaurer la valeur de `$sp` à la fin.
2. Il est tentant d'optimiser le programme précédent en laissant `sp` inchangé et en accédant aux données en utilisant des déplacements négatifs à partir de `$sp`. Pourquoi une telle approche ne peut-elle pas marcher sur un vrai processeur (indication : pensez aux interruptions asynchrones) ?

3. Appel de fonctions feuilles

Le but de cet exercice est d'implémenter en assembleur le programme suivant :

```
int main()
{
    printf("%d\n", max(3, 5));
}

int max(int x, int y)
{
    int m;
    if(x >= y)
        m = x;
    else
        m = y;
    return m;
}
```

3.1 Écrivez en assembleur le code de `main` qui effectue la suite d'actions suivante :

1. empile les paramètres effectifs ;
2. appelle la fonction `max` à l'aide de l'instruction `jal` ;

[1] *Statique* caractérise tout ce qui se fait lors de l'écriture du programme ou lors de sa compilation, contrairement à *dynamique*, qui se réfère à ce qui se fait lors de l'exécution.

3. dépile les paramètres ;
4. affiche le résultat, qui se trouve dans `$v0`.

3.2 Quelle est la taille en octets du cadre de pile de `max` (en incluant l'espace nécessaire pour la sauvegarde de l'ancienne valeur de `$fp` ? On appellera cette valeur `framesize` = `frameoffset`.

Écrivez la fonction `max` en assembleur. Cette fonction devra effectuer la suite d'actions suivante :

1. ajuster `$sp` pour pointer au sommet (qui se trouve en bas...) du nouveau cadre de pile ;
2. sauvegarder l'ancienne valeur de `$fp`, puis faire pointer `$fp` à l'adresse la plus basse de l'ancien cadre de pile ;
3. exécuter le corps de `max` ;
4. restaurer la valeur de `$fp` puis `$sp` ;
5. retourner à l'appelant, à l'adresse stockée dans `$ra`.

4. Appel de fonctions générales

Implémentez en assembleur les fonctions suivantes :

```
int pair(int n)
{
    if(n == 0)
        return 1;
    return !impair(n - 1);
}

int impair(int n)
{
    if(n == 0)
        return 0;
    return !pair(n - 1);
}
```

Quel(s) registe(s) supplémentaire(s) faudra-t-il sauvegarder ?