

Compilation — TD 3

V. Balat, P. Letouzey, G. Manzonetto

2008–2009

1. 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;
}
```

1.1 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` ;
3. dépile les paramètres ;
4. affiche le résultat, qui se trouve dans `$v0`.

1.2 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`.

2. 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) registre(s) supplémentaire(s) faudra-t-il sauvegarder ?

4. Utilisation de la librairie standard du projet

Le langage CTigre que vous compilez dispose d'une petite bibliothèque standard de fonctions qui sont disponibles pour le programmeur et/ou le compilateur. Elle se présente sous la forme d'un fichier `runtime.S` en assembleur, qu'il suffira d'inclure à la fin de vos fichiers assembleur. Les fonctions de cette bibliothèque sont les suivantes :

sortie :

print(s) imprime la chaîne de caractères *s*

printint(i) imprime l'entier *i*

entrée :

getchar() lit un caractère en entrée (opération bloquante)

getString(n) lit une chaîne de caractères de longueur maximale *n* en entrée (opération bloquante)

readint() lit un entier

conversions :

ord(c) le code ASCII du caractère *c*

chr(i) le caractère de code ASCII *i*

mkstring(c) une chaîne de caractères contenant le seul caractère *c*

chaînes :

size(s) renvoie la longueur d'une chaîne

concat(s1,s2) la concaténation des chaînes *s1* et *s2*

mémoire :

malloc(n) alloue *n* octets sur la tas et retourne l'adresse correspondante

exit() sort du programme

1. Modifier le programme suivant pour utiliser les fonctions de la bibliothèque standard pour les entrées sorties

2. Dans le TP précédent, nous avons alloué un tableau statiquement ou dynamiquement sur la pile. Dans votre compilateur CTigre, les tableaux seront alloués dynamiquement sur le tas. Modifier votre programme pour utiliser la fonction `malloc` de la bibliothèque standard CTigre.

```
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);
}
```

5. Exécution sur une machine MIPS

1. Copiez un de vos programmes sur le compte `guest` de la machine `debi anmi ps` (mot de passe `guest`). Comme tout le monde utilise le même compte, faites attention à ne pas écraser les fichiers des autres. Il est recommandé de vous créer un répertoire.
2. Vérifiez que l'extension de votre fichier est `.S` et non `.s` (pour que le préprocesseur `cpp` le traite).
3. Ajoutez la ligne `#include "runtime.s"` au début de votre fichier pour inclure la version appropriée de la bibliothèque standard.
4. Compilez (« assemblez ») à l'aide de la commande :
`./cc -o votreprogramme votrefichierasm.S`
5. Exécutez : `./votreprogramme`

6. Une soirée à Manhattan...

Manhattan est (disons) un quadrillage carré de n rues par n avenues. Ce soir, la municipalité de Manhattan a décidé de mettre les rues en sens unique vers l'Est, et les avenues en sens unique vers le Nord. Vous arrivez donc avec votre voiture du pont de Brooklyn, au coin Sud-Ouest (position $(0, 0)$), et vous voulez vous rendre à votre restaurant favori (position (n, m)). Combien de parcours pouvez-vous faire pour y arriver ?

La fonction récursive suivante calcule la réponse :

$$f(n, m) = \begin{cases} 1, & \text{si } n = 0 \text{ ou } m = 0 \\ f(n - 1, m) + f(n, m - 1) & \text{sinon.} \end{cases}$$

Implémentez en assembleur la fonction f .