

# Architecture des Ordinateurs, MIPS, TP 1

## Recommandations

- Etudier préalablement, et utiliser le MémomIPS proposé en ligne.
- Commenter soigneusement *chaque* ligne écrite en assembleur...

**Exercice 1.** Ecrire en assembleur MIPS un programme qui affiche à l'écran le message

`Ceci est une chaine de caracteres en ascii`

suivi d'un saut à la ligne.

**Exercice 2.** Etant donné le programme MIPS suivant :

```
.text
ori $t0, $zero, 2
e1: ori $v0, $zero, 5
    syscall
    or $a0, $zero, $v0
    mul $a0, $a0, $t0
    ori $v0, $zero, 1
    syscall
    bne $a0, 100, e1
    ori $v0, $zero, 10
    syscall
```

1. Que fait ce programme ?
2. Quel est le type de boucle mis en œuvre ?
3. Comment le compléter pour qu'il affiche un message invitant l'utilisateur à saisir une valeur à chaque itération.
4. Comment, en remplaçant au moins une instruction par une autre, améliorer significativement sa rapidité d'exécution ?

**Exercice 3.** A partir des possibilités d'appels système sur les entrées-sorties, écrire un programme qui lit répétitivement un entier au clavier et l'ajoute aux entiers précédemment lus. Le programme s'arrête quand l'entier lu est 0, et affiche alors le message "Somme = ", suivi du résultat.

**Exercice 4.** Le programme C++ suivant permet d'afficher le contenu d'un tableau T :

```
#include <iostream>
#include <iomanip>

using namespace std;

int main (void)
{
    const unsigned N=5;
    unsigned T[]={1, 2, 3, 4, 5};

    cout << "T :\n";
    for (unsigned i = 0 ; (i < N) ; ++i)
        cout << setw(3) << T[i];

    return 0;
} // main()
```

En respectant la structure et la sémantique de ce programme, le traduire en assembleur MIPS en optimisant au mieux les échanges entre mémoire et registres, et en commentant chaque instruction en assembleur.

**Exercice 5.** On considère les registres \$a0, \$a1 ; \$a2 associés respectivement à l'adresse de début d'un tableau **save**, au paramètre **k**, et au paramètre **n** passés à la fonction **Recherche** du programme C/C++ suivant :

```
#include <iostream>

using namespace std;

int Recherche (int save[], int k, int rang)
{    unsigned i;

    for(i=0; i <= rang && save[i] != k; ++i) {};
    return i;
} // Recherche()

int main (void)
{
    int rangmax=9;
    int save [] = {0, 0, 0, 0, 0, 0, 0, 1, 2, 3};
    cout << Recherche (save, 1, rangmax);
    return 0;
} // main()
```

En associant par ailleurs le registre \$s0 à la variable locale **i** de la fonction, traduire ce programme aussi fidèlement que possible en assembleur MIPS (en commentant chaque instruction).