

# Interprétation des programmes – TP 7 : Découverte de MIPS

Université Paris Diderot – Master 1

(2015-2016)

Cette feuille de TP vous donne les étapes à suivre pour découvrir l'architecture MIPS à l'aide d'exemples tout en parcourant la spécification de l'assembleur :

[http://spimsimulator.sourceforge.net/HP\\_AppA.pdf](http://spimsimulator.sourceforge.net/HP_AppA.pdf)

Pour le moment, on mettra de côté :

- la convention d'appel des fonctions ;
- la compilation séparée ;
- les exceptions et les interruptions ;
- l'encodage des instructions ;
- les calculs flottants.

## Exercice 1 (Hello MIPS et variantes)

1. Avant de commencer à programmer, partez à la recherche des informations suivantes dans la spécification :
  - (a) Combien l'architecture MIPS a-t-elle de registres ?
  - (b) Quelles sont les instructions disponibles ?
  - (c) Quelles sont les différents types d'adressage sur cette architecture ?
  - (d) Y-a-t-il des zones "réservées" dans la mémoire ?
  - (e) Que signifie le terme "aligned data" ? Qu'est-ce que cela implique sur l'usage de la mémoire ?
  - (f) Est-ce qu'il s'agit d'une architecture little endian ou big endian ?
  - (g) Quels sont les appels systèmes fournis par SPIM ?
2. À quoi sert pMars ?
3. Quelle est la structure d'un programme MIPS ?
4. Utilisez pMars et la spécification MIPS pour comprendre ce que fait chaque instruction et directive du programme suivant :

```
.data
pstart: .asciiz "*** Program start ***\n"
pend:
.asciiz "\n*** Program end ***\n"
.text
main:
la $a0, pstart
li $v0, 4
syscall
li $a0, 28
li $v0, 1
syscall
```

```
la $a0, pend
li $v0, 4
syscall
jr $ra
```

5. Écrivez un programme MIPS qui affiche "Hello MIPS!"
6. Qu'est-ce qu'une macro ? Quelle est la syntaxe pour définir une macro ? Quelles différences faites-vous entre les macros et les fonctions/procédures ?
7. Définissez une macro `print_int` qui affiche un entier.
8. Écrivez un programme MIPS qui affiche les entiers de 0 à 1023.

□

## Exercice 2 (Calculs itératifs en MIPS)

1. Écrivez un programme MIPS qui demande deux entiers positifs à l'utilisateur et affiche la somme des entiers compris entre ces deux entiers.
2. Écrivez un programme MIPS qui affiche les nombres premiers compris entre 2 et 1023. Vous utiliserez l'algorithme d'Eratosthène en représentant le crible en mémoire.

□

## Exercice 3 (Calculs à l'aide de la pile en MIPS)

1. Comment est représentée la pile du système dans la mémoire de MIPS ?
2. Comment est initialisée la mémoire et les registres ?
3. Comment pousser une valeur sur la pile de MIPS ?
4. Comment retirer une valeur du sommet de la pile de MIPS ?
5. Comment tester si la pile est vide ?
6. Écrire une macro `print_stack` qui affiche le contenu de la pile.
7. Écrire un programme MIPS qui attend que l'utilisateur saisisse un entier  $N$  et qui affiche de  $N$ -ième entier de la suite de Fibonacci. Vous devrez utiliser la pile de MIPS pour stocker les termes intermédiaires de la suite.

□