

# TP assembleur MIPS

David Delahaye

[David.Delahaye@lirmm.fr](mailto:David.Delahaye@lirmm.fr)

Faculté des Sciences

Master M1 2020-2021



# Programmation en MIPS

## Outils

- Utilisation d'un émulateur ;
- Plusieurs émulateurs : MARS (celui qu'on utilisera), SPIM ;
- Télécharger MARS (lien sous Moodle) ;
- Exécuter l'émulateur : `java -jar Mars4_5.jar`.

## Principe

- Écriture de son programme dans l'éditeur ;
- Assemblage du programme (génération du binaire) ;
- Exécution du programme (possibilité de faire du pas à pas) ;
- Vue sur l'état de la mémoire et des registres.

# À vous de jouer !

## Exercices

- Demander la saisie d'un entier et rendre la valeur absolue de cet entier (afficher le résultat) ;
- Initialiser un tableau de 3 entiers (sans le saisir) et permuter les éléments de ce tableau ;
- Afficher les  $n$  premiers entiers (en partant de 1), où l'entier  $n$  sera demandé à l'utilisateur ;
- Demander la saisie d'un entier et dire si cet entier est pair ou non (afficher le résultat).

# À vous de jouer !

## Exercices

- Écrire une routine qui permute le contenu de deux variables entières de la zone de données avec une variable locale pour effectuer la permutation ;
- Écrire le code assembleur correspondant au code C suivant :

```
int sqr (int x) {  
    return x * x;  
}  
  
int sum (int x, int y) {  
    return sqr(x) + sqr(y);  
}
```

# À vous de jouer !

## Exercices

- Écrire une routine qui effectue récursivement la somme des  $n$  premiers entiers, où  $n$  est un entier passé en argument ;
- Écrire une routine qui implante la suite de Fibonacci :

$$fib(n) = \begin{cases} n, & \text{si } n = 0, 1 \\ fib(n-1) + fib(n-2), & \text{sinon} \end{cases}$$