

Objectif : Appels aux sous programmes et gestion de la pile

Exercice n°1 :

Écrire un programme MIPS qui demande deux entiers et un opérateur arithmétique (+, -, x, /) à l'utilisateur (en entrée), puis affiche le résultat de l'opération en utilisant les fonctions suivantes : Addition, Soustraction, Multiplication et Division.

Exercice n°2 :

Écrire un programme MIPS qui saisit 10 entiers aux claviers, les empiler en mémoire ; puis les afficher à l'écran dans l'ordre inverse.

Exercice n°3 : Trouver le maximum d'un tableau (avec appel de sous-programme itératif)

Ecrire un programme principal qui initialise les valeurs du tableau. Puis Implémenter un sous-programme **find_max** qui prend en entrée l'adresse du tableau et sa taille, puis retourne la valeur maximale. Et enfin afficher la valeur maximale à l'aide du programme principal.

Indications :

- Utiliser \$a0 pour l'adresse du tableau, \$a1 pour sa taille, et \$v0 pour retourner le maximum.

Exercice n°4 : Calcul de la factorielle (avec appel de sous-programme récursif)

Écrire un programme en assembleur MIPS qui lit un entier positif n et calcule la factorielle de n de façon récursive.

Indications:

- Si $n = 0$ ou $n = 1$, la fonction retourne 1.
- Sinon, elle retourne $n * \text{fact}(n-1)$.
- Utiliser les registres \$a0 pour passer le paramètre et \$v0 pour retourner le résultat.

Lab Series No. 4

Objective: Subprogram calls and stack management.

Exercise 1:

Write a MIPS program that prompts the user for two integers and an arithmetic operator (+, -, *, /) as input, then displays the result of the operation using the following functions: Addition, Subtraction, Multiplication, and Division.

Exercise 2:

Write a MIPS program that reads 10 integers from the keyboard, pushes them onto the stack in memory, and then displays them on the screen in reverse order.

Exercise 3: Finding the maximum value in an array (with iterative subprogram call)

Write a main program that initializes the values of an array. Then implement a subprogram **find_max** that takes as input the address of the array and its size, and returns the maximum value. Finally, display the maximum value using the main program.

Hints:

- Use \$a0 for the array address, \$a1 for its size, and \$v0 to return the maximum value.

Exercise 3: Factorial Calculation (with recursive subprogram call)

Write a MIPS assembly program that reads a positive integer n and calculates the factorial of n recursively.

Hints:

- If n = 0 or n = 1, the function returns 1.
- Otherwise, it returns n * fact(n-1).
- Use register \$a0 to pass the parameter and \$v0 to return the result.