

Laboratorio #3

Generación de Código Ensamblador

Traducción de Expresiones Aritméticas

A. Generación de Código Ensamblador

Con el simulador listo, escribir un programa en MIPS que calcule el de forma recursiva el Máximo Común Divisor de dos números enteros cualesquiera.

Inicio

Algoritmo de Euclides

Enviado por fmoreno el Dom, 22/09/2019 - 19:51

Leer este manual en este lenguaje de programación: Todos ▾

El algoritmo de Euclides es un conocido algoritmo para calcular el máximo común divisor de dos números. Se basa en la siguiente propiedad: $\text{mcd}(a, b) = \text{mcd}(a - b, b)$. Así, si se va sustrayendo el número menor de el número mayor, cada vez los pares de números que quedan se van haciendo más pequeños hasta que uno de los números es 0, y $\text{mcd}(a, 0) = a$.

Por ejemplo, para calcular el máximo común divisor de 105 y 70:

$$\text{mcd}(105, 70) = \text{mcd}(105 - 70, 70) = \text{mcd}(35, 70) = \text{mcd}(35, 70 - 35) = \text{mcd}(35, 35) = \text{mcd}(35, 35 - 35) = \text{mcd}(35, 0) = 35$$

Se puede programar de forma

C++

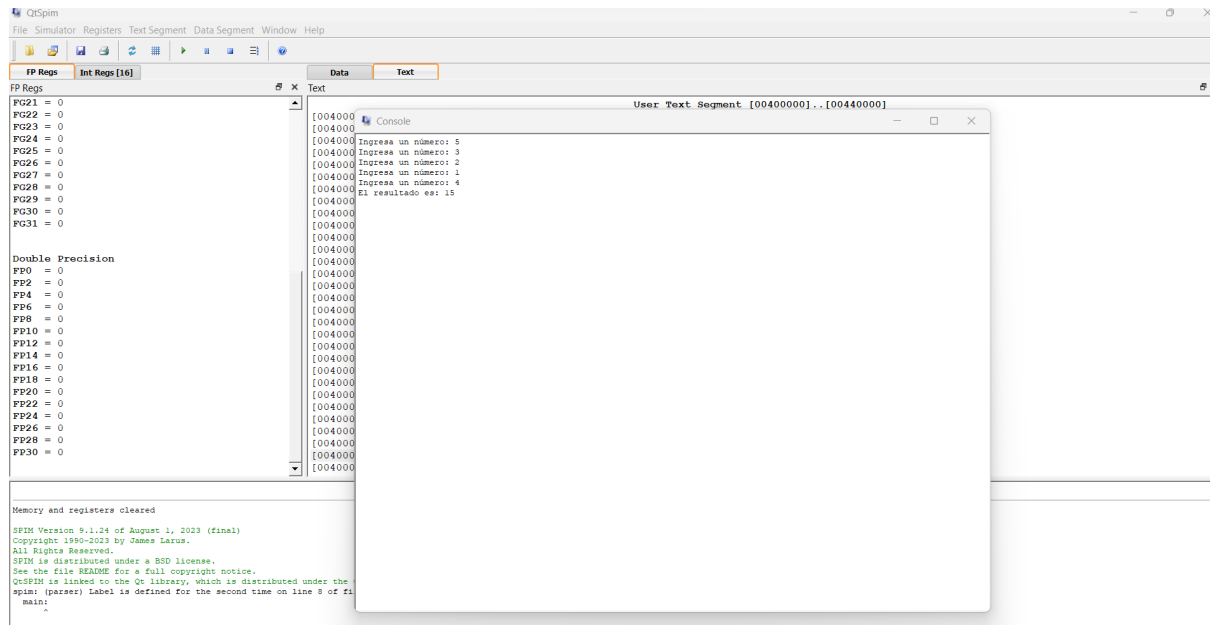
```
int mcd(int a, int b) {
    if(b == 0) return a;
    return mcd(b, a%b);
}
```

Console

```
Enter the first number: 105
Enter the second number: 70
Your final result is: 35
```

Python

ejemplo ingresando 5 valores y sumandolos



se adjuntan archivos en entrega

B. Traducción de Expresiones Aritméticas

Para las operaciones de suma y división, dentro de sus expresiones aritméticas, realizar la traducción a instrucciones MIPS equivalentes (que cumplan la estructura y semántica de MIPS)

Operaciones aritmeticas a MIPS

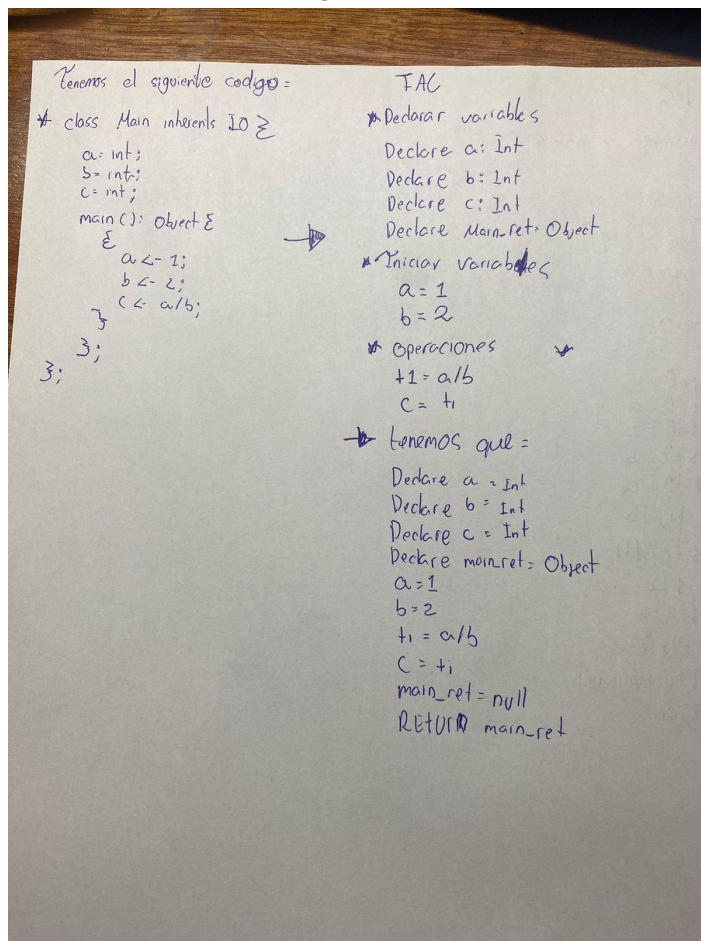
1. **Resta ($x = y - z$):**
`sub $t0, $s1, $s2 # $t0 = $s1 - $s2`
2. **Multipliación ($x = y * z$):**
`mul $t0, $s1, $s2 # $t0 = $s1 * $s2`
3. **Suma constante ($x = y + 5$):**
`li $t1, 5 # Cargar el valor 5 en $t1`
`add $t0, $s1, $t1 # $t0 = $s1 + 5`
4. **División constante ($x = y / 3$):**
`li $t1, 3 # Cargar el valor 3 en $t1`
`div $s1, $t1 # $lo = $s1 / 3`
`mflo $t0 # $t0 = $lo`
5. **Operación con tres operandos ($x = y + z - w$):**
`add $t0, $s1, $s2 # $t0 = $s1 + $s2`
`sub $t0, $t0, $s3 # $t0 = $t0 - $s3`
6. **Operación mixta ($x = y * 2 + z / 3$):**

```

li $t1, 2      # Cargar el valor 2 en $t1
mul $t0, $s1, $t1 # $t0 = $s1 * 2
li $t2, 3      # Cargar el valor 3 en $t2
div $s2, $t2    # $s2 = $t2 / 3
mflo $t2        # $t2 = $s2
add $t0, $t0, $t2 # $t0 = $t0 + $t2

```

Ejemplo de traducción de un programa escrito en YAPL, que contenga expresiones aritméticas hacia el código MIPS respectivo.



→ MIPS

Declaraciones de variables en MIPS.

- data

a: .word 0

b: .word 0

c: .word 0

main_ret: .word 0

- text

- globl main

main:

li \$t0, 1

sw \$t0, a

li \$t0, 2

sw \$t0, b

lw \$t0, a

lw \$t1, b

div \$t0, \$t1

mflo \$t2

sw \$t2, c

li \$t0, 0

sw \$t0, main_ret

jr \$ra