**PRÁCTICA 2.4  
Programación en ensamblador MIPS. Procedimientos II**

**Objetivos:**

* Manejar cadenas de caracteres
* Manejar la pila en las llamadas anidadas

**Duración: 2** horas presenciales + trabajo personal

**Desarrollo / Comentario:**

* Escribir el equivalente en ensamblador MIPS32 del siguiente programa en C, donde existen llamadas anidadas.

//Dada una cadena cualquiera de longitud máxima 10, realizar un programa que copie

//en una cadena destino la cadena origen pero eliminando los espacios en blanco que haya.

//También debe obtener el número de espacios en blanco eliminados

#include <stdio.h>

int eliminar\_espacios(char cad1[], char cad2[]);

void imprimircadena(char cad[]);

void main(void){

char cadenao[10] = "123 n f";

char cadenad[10];

int eliminados = 0;

eliminados = eliminar\_espacios(cadenao, cadenad);

printf("la cantidad de espacios eliminados es %d \n", eliminados);

}

int eliminar\_espacios(char cad1[], char cad2[]){

int i=0, j = 0, espacios = 0; //i para cadena origen y j para cadena destino

while (cad1[i] != '\0')

{

if (cad1[i] != ' ') {

cad2[j] = cad1[i];

j++;

}

else espacios++;

i++;

}

cad2[j] = 0;//para acabar la cadena con nulo

imprimircadena(cad2);

return espacios;

}

void imprimircadena(char cad[]){ //Imprime una cadena de caracteres

printf("%s", cad);

}

A continuación se muestra una implementación en ensamblador MIPS32 del programa, en el cual falta incluir la implementación de la subrutina eliminar\_espacios.

#segmento de datos

.data

cadenao: .asciiz "123 n f" #acaba la cadena con 00

cadenad: .space 10

eliminados: .word 0

string: .asciiz "la cantidad de espacios eliminados es"

# segmento de texto

.text

.globl main

main:

la $a0,cadenao

la $a1,cadenad

jal eliminar\_espacios #Llamada a la función

la $t0, eliminados

sw $v0,0($t0) #escribo en memoria resultado

la $a0, string

li $v0, 4

syscall #llamada al sistema para imprimir "la cantidad de espacios eliminados es"

la $t0,eliminados

lw $a0,0($t0)

addi $v0, $0, 1 #llamada al sistema para imprimir eliminados

syscall

addi $v0, $0, 10 #llamada al sistema para salir del programa

syscall

#===============rutina eliminar\_espacios

eliminar\_espacios: .

.

.

.

.

jal imprimircadena

.

.

jr $ra

A continuación se muestra una implementación en ensamblador MIPS32 de la subrutina imprimircadena.

#======== subrutina que imprime la cadena cuya dirección está en la cima de la pila

imprimircadena:

lw $a0,0($sp) #desapilo en $a0

addi $sp,$sp,4

addi $v0, $0, 4 #llamada al sistema para imprimir cadena destino

syscall

jr $ra #retorno al programa llamante

* Escribir el estado de la pila justo antes de ejecutar la primera instrucción de la subrutina imprimircadena.