

# ARQUITECTURA DE COMPUTADORES

## 543.426

Tarea No. 1  
27 de marzo de 2017

Resuelva los siguientes problemas utilizando la convención estándar de uso de registros MIPS (temporales, argumentos, valores de retorno, etc.). La eficiencia del código (velocidad, tamaño, uso de registros) también se considerará en la evaluación de la tarea.

Entregar un informe escrito (en computador) con el código debidamente comentado y las explicaciones correspondientes. Plazo máximo de entrega: Martes 4 de Abril hasta las 17 horas en Secretaría de Electrónica. No se corregirán tareas atrasadas.

Se debe trabajar en grupos de 2 personas. Grupos de 1 persona son permitidos pero no recomendados. No se permitirán grupos de más de 2 personas. Se les recuerda que cualquier copia (entre tareas o de fuentes externas) resultarán en calificación 1 para todas las tareas involucradas.

### Problema 1

Escriba una función en assembly MIPS que busque la primera aparición de un string “s2” dentro de otro string “s1”. La posición se deberá retornar como un entero que apunte al carácter de “s2” encontrado. En caso de encontrar el string deberá retornar el entero 0. El siguiente código en C es una referencia de la implementación de esta función.

```
1  int strfind (char *s1, char *s2)
2  {
3      char *begin_s2 = s2;
4      int s1_cnt = 0;
5      int cnt = 0;
6
7      while(*s2 != 0){
8          if(*s1 == 0)
9              return 0;
10
11         if(*s1 == *s2)
12             s2++;
13
14         else{
15             s2 = begin_s2;
16             cnt = s1_cnt + 1;
17         }
18
19         s1++;
20         s1_cnt++
21     }
22     return cnt;
23 }
```

Esta función en C recibe como argumentos los punteros hacia los primeros caracteres de cada string. Asuma que las direcciones de dichos caracteres se reciben en su función en los registros \$a0 y \$a1.

## Problema 2

La siguiente función en C es una implementación que permite calcular el promedio de datos transferidos por segundo en kb/s desde “n” dispositivos de almacenamiento flash en un equipo. Esta función recibe como argumento la dirección de un vector de elementos de una estructura llamada “datos”. Además recibe como argumento la cantidad “n” de elementos de dicha estructura que posee el vector. La estructura “datos”, también definida en el código, posee la información de la cantidad de datos transferidos (en kb) y el tiempo que tardó en transferirlos para cada dispositivo.

```
1 struct datos {
2     int cantidad;           // cantidad de datos en kb
3     int tiempo;            // tiempo en s
4 } datos;
5
6 int prom_transf (struct datos *pd, int n)
7 {
8     int sum = 0;
9     int cnt;
10    int flujo;
11    for(cnt=0; cnt < n; cnt++){
12        flujo = (pd+cnt).cantidad / (pd+cnt).tiempo;
13        sum = sum + flujo;
14    }
15
16    return sum/n;
17 }
```

Escriba una función en assembly MIPS que sea capaz de ejecutar esta función. Asuma que el programa puede detectar como mínimo una velocidad de 1 kb/s.

## Problema 3

La siguiente línea de código es el protipo de una función en C:

```
1 int spc_sum (int *v1, int n);
```

Esta función realiza operaciones entre los elementos de un vector. Se recibe como argumento un puntero al primer elemento de dicho vector. Además se recibe como un segundo argumento un entero que indica la cantidad de elementos que posee el vector. La función calcula la suma de todos los elementos múltiplos de 3 y múltiplos de 5 (exceptuando aquellos valores repetidos que ya hayan sido considerados como múltiplos de 3) que posee el vector de largo “n”.

Escriba una función en assembly MIPS que cumpla con la funcionalidad descrita y que retorne como entero la suma final total. Al escribir su función busque minimizar la cantidad de instrucciones necesarias para su ejecución. Estime cuantas instrucciones requiere un vector de “n” elementos para ejecutar su función.