

PRINCIPIO DE COMPUTADORES.

PRÁCTICA 5. BÚSQUEDA LINEAL EN UN VECTOR.

Descripción.

Te proponemos realizar una subrutina en ensamblador para resolver el problema de la búsqueda lineal de un elemento en un vector dado. La búsqueda lineal consiste en ir inspeccionando cada elemento del vector de izquierda a derecha hasta que, o bien se encuentra el elemento buscado (en cuyo caso se devolverá un uno), o bien se ha recorrido todo el vector (en cuyo caso se devolverá un cero).

Esta práctica deberá resolverse en pareja, y se deberán entregar dos soluciones del mismo problema (una solución iterativa y una solución recursiva). En ambos casos, la subrutina devolverá 1 cuando se encuentre el elemento y 0 en caso contrario. Proporcionaremos el cuerpo principal del programa y el segmento de datos para facilitar la resolución de esta práctica. En ambos casos deberás introducir el pseudocódigo (o código en un lenguaje de alto nivel) como comentario. Implementa posteriormente tu programa en ensamblador acorde al diseño que propongamos.

- 1) Realiza una primera versión de la subrutina de manera que sea NO recursiva.
- 2) Realiza una segunda versión de la subrutina, ahora utilizando la recursividad. Recordemos que la búsqueda lineal consiste en ir inspeccionando cada elemento del vector de izquierda a derecha hasta que, o bien se encuentra el elemento buscado, o bien se ha recorrido todo el vector. Así puedes plantear la recursividad de la siguiente manera:
 - Si buscamos un elemento en un vector de tamaño cero podemos concluir que el elemento no está en éste.
 - Si el elemento buscado coincide con el primer elemento del vector entonces lo hemos encontrado.
 - En otro caso deberemos buscar (llamada recursiva) a partir de la siguiente posición del vector.
 -

En el programa `busquedaLineal.s` tienes el cuerpo principal y la llamada

```
# PROGRAMA QUE REALIZA UNA BÚSQUEDA LINEAL DE UN ELEMENTO
# EN UN VECTOR.

size = 4          # tamaño de los elementos del vector
N = 10            # número de elementos del vector

.data
vector:           # 2,3,5,6,8,1,3,2,5,9
.abuscar:         # 6
.encontrado:      # 0

.text
main:
    la    $a0,vector    # primer parametro la direccion base del vector
    li    $a1,N          # segundo parametro el numero de elementos del vector
    lw    $a2,abuscar    # tercer parametro el elemento a buscar
    jal   blineal        # quedara en $v0 1 si lo encontro 0 en caso contrario
    sw    $v0,encontrado # almacenamos en resultado de busqueda en una posicion de memoria

    # salida limpia del sistema
    li    $v0,10
    syscall

# FIN DEL PROGRAMA PRINCIPAL

blineal:          # funcion blinealR:
#                 $a0 direccion base del vector
#                 $a1 numero de elementos del vector a buscar
#                 $a2 elemento buscado
#                 devuelve: $v0 = 1 si encuentra el elemento $v0 = 0 en caso contrario
#                 INTRODUCE COMO COMENTARIO EL PSEUDOCÓDIGO O EL PROGRAMA EN LENG. ALTO NIVEL
#
#
#
#
#
#
# INTRODUCE AQUI EL CÓDIGO DE TU SUBROUTINA
```