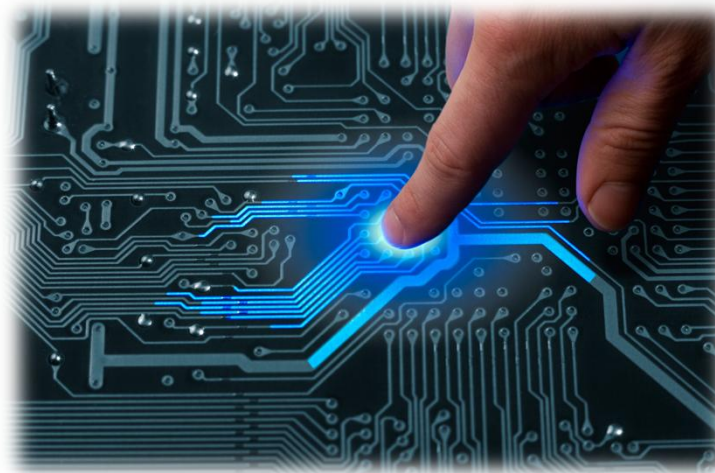


Arquitectura de Computadores
Práctica 1. Diseño de un Benchmark Sintético



Programa “Cadena”

Cadena: Bloc de notas

```
Archivo Edición Formato Ver Ayuda
.MODEL small

.DATA
mensaje db ' -0123456789ABCDEFGHIJKMNÑOPQRSTUVWXYZ-0123456789ABCDEFGHIJKMNÑOPQRSTUVWXYZ-0123456789ABCDEFGHIJKMNÑOPQRSTUVWXYZ **'
; Cadena formada por [1 comienzo_de_cadena + 36caracteres x 3 + 3 guiones + 2 espacios // el asterisco no se imprime]

.CODE

    MOV AX,seg mensaje
    MOV DS,AX

    MOV AL, 03h
    MOV AH, 0
    INT 10h

    MOV AX, 0B800h ; Elegimos la dirección de memoria en modo texto
    MOV ES, AX

pinta_cadena:
    MOV AL, mensaje[SI]
    MOV AH, 0Eh
    MOV ES:[DI], AX
    ADD DI, 2

    INC SI
    CMP mensaje[SI], ""
    JNE pinta_cadena

    MOV SI, 0
    INC mensaje[0]
    CMP DI, 4158 ; el valor máximo que puede alcanzar DI es 4158 = [80 x fila + columna]x2 = [80 x 25 + 79]x2
    JBE pinta_cadena
    MOV DI, 0 ; si ha llegado al final de la pantalla, vuelve a principio de pantalla
    JMP pinta_cadena

END
```

7 instrucciones fuera del bucle

13 instrucciones dentro del bucle

Línea 1, columna 1

19:58
10/03/2019

Programa "Reside"

Reside: Bloc de notas

```

Archivo Edición Formato Ver Ayuda

.MODEL SMALL
.CODE
    org 100h
Programa_Int:
    JMP Reside
    contador_int db 0

Rutina_Servicio PROC
    CLI
    INC contador_int
    CMP contador_int, 2
    JNE fin

    MOV AH, 0
    INT 16h

    MOV AL, 03h
    MOV AH, 0
    INT 10h

    MOV AH, 4ch
    INT 21h
fin:
    STI
    IRET
ENDP
Reside: ;etiqueta para determinar la dirección siguiente a la última de la rutina que debe quedar residente
    MOV DX, offset Rutina_Servicio
    MOV AX, 0
    MOV ES, AX
    MOV SI, 1ch*4
    CLI
    MOV ES:[SI], DX
    MOV ES:[SI+2], CS
    STI
    MOV DX, offset Reside
    INT 27h
    
```

Contador interrupciones: 2

7 instrucciones (contador = 1)

11 instrucciones (contador = 2)

10 instrucciones a partir de "Reside"

Cálculo MIPS

1. Número impresiones de cadena = $107 - 31 = 76$ (75 completas + 1 no completa)
2. Código ascii carácter 'k' = 107d y del espacio = 32d
3. Contador interrupciones = 2
4. Tiempo entre interrupciones = 55 ms = $1/18,2$
5. Tiempo de ejecución del programa principal = $2 \times 55 \text{ ms} = 110 \text{ ms}$
6. Programa Principal - Instrucciones ejecutadas:

Repetido
75 veces

- 7 instrucciones fuera del bucle
- 13 instrucciones dentro del bucle para imprimir la cadena completa 75 veces::

3 carácter_comienzo cadena (a, b, c, ...) + 36 caracteres x 3 +

3 guiones (-) + 2 espacios // el asterisco no se imprime

- 13 instrucciones para imprimir última cadena:

1 carácter_comienzo cadena (k) + 36 caracteres x 2 +

+ 3 guiones (-) + 9 números (0-8 últimos caracteres)

7. Rutina de Servicio a la Interrupción- Instrucciones ejecutadas:

- 28 instrucciones (7 + 11 + 10)

Cálculo MIPS

Instrucciones ejecutadas:

$$7 + 75 \times [13 \times (3 + 36 \times 3 + 3 + 2)] + 13 \times (1 + 36 \times 2 + 3 + 9) + 28 = 114240$$



fuera del bucle



75 veces Cadena completa



Última
impresión



Rutina de
Servicio a la
Interrupción

Tiempo de ejecución: 110 ms

$$\text{MIPS} = 113264 \text{ instrucciones} / (110 \text{ ms} \times 10^6) = 1,03$$

- ✓ Dato sin incluir las instrucciones que se ejecutan cada vez que se provoca una interrupción software del tipo INT xxh
- ✓ Cada interrupción INT xxh implica ejecutar una Rutina de Servicio a la Interrupción formada por un número de instrucciones concreto