



Universidad
Francisco de Vitoria
UFV Madrid

Fundamentos de la ingeniería informática

Ingeniería de sistemas industriales

Curso 2019-2020

Ejemplos 1

1. Tablero

tablero es una secuencia 64 de **casillas** donde se pueden ubicar **fichas** de distintos colores (rojo amarillo, verde y azul), 4 de cada color, en cada casilla, al mismo tiempo no puede haber más de una ficha, así que la estrategia para saber dónde se encuentra cada ficha es anotar en la casilla en la que se encuentra una ficha su color R (ascii \$52) para el rojo, Y(ascii \$59) para el amarillo, G (ascii \$47) para el verde y (ascii \$42) para el azul. En caso de no contener ninguna ficha, el contenido de la casilla es cero. De este modo en un momento dado el tablero puede tener el siguiente aspecto:

\$00	
\$01	
\$02	Y
\$03	R
\$04	G
\$05	
\$06	
\$07	Y
\$08	R
\$09	
\$0A	B
\$0B	B
\$0C	
\$0D	
...	
\$3B	
\$3C	B
\$3D	G
\$3E	Y
\$3F	

En las soluciones que vienen a continuación se declaran recursos (p. ej: Variables) que cuando se construya el programa completo habrá que reunir

1.1. Colocar

Se pide redactar una subrutina en Ensamblador CH-2020 que tenga:

Entradas

- D0 un color de ficha (R, G, B o Y)
- A0 la dirección del tablero
- D1 un valor de posición ($0 \leq \text{posición} < \40)

Salidas

- D0 resultado de la operación

Y que coloque en la posición indicada por D1 una ficha del color indicado por D0 y devuelva en D0 un \$00, con la salvedad de qué si en la casilla indicada ya se encuentra previamente una ficha, la rutina no modificará la casilla, pero devolverá \$FF.

Solución:

La dirección de la casilla será la dirección del tablero + la posición de la casilla.

Una buena opción para salvar temporalmente datos es utilizar la pila para ello. La instrucción push guarda un dato en la pila, la operación pop saca el último dato. Hay que tener en cuenta que, debido al funcionamiento de la pila, los datos que se guardan en ella hay que recuperarlos en orden inverso al de introducción y de que se sacan todos los datos que se han metido en ella y no más. También hay que tener cuidado de que el programa principal inicie el valor del sp a una dirección que nos asegure que no colisiona con otros espacios de datos. Se suele ubicar al final de la memoria de datos, en este caso \$FFFF.

```

;*****
; COLOCAR
;   Entradas:
;       D0: Color
;       A0: Tablero
;       D1: Posición
;   Salidas:
;       D0: Resultado OK ($00), Error ($FF)
;*****
OK    = 0
ERROR= 0FF

Colocar
    push d0                ; Salvo D0 para posterior uso.
    mov  d1 d0             ; Obtener la dirección de la casilla.
    call Incrementar
    mov  (a0) d0           ; Verificar que está vacía.
    mov  #0 D1
    cmp
    pop  d0                ; Recupero el valor del color de la
ficha.
    jr   NZ Colocar_error
    mov  d0 (a0)           ; Dejo la nueva ficha en la casilla
vacía.

    mov  #OK d0            ; Devolver OK
    jmp  Colocar_fin

Colocar_error
    mov  #ERROR d0        ; Cuando la celda está ocupada
;    devolver error

Colocar_fin
    ret

```

1.2. Buscar

Se pide redactar una subrutina en Ensamblador CH-2020 que tenga:

Entradas

- En D0 un color de ficha (R, G, B o Y)
- En A0 la dirección donde se encuentra el tablero
- En D1 un valor de inicio (< \$40)

Salidas

- D0 con la posición buscada.

Operación

Busque la primera casilla ocupada por una ficha del color indicado por D0 a partir de la casilla indicada por D1 (inicio \leq posición < \$40) y devuelva su posición en D0

En caso de que no se encuentre ninguna ficha de dicho color devolverá \$FF

Solución:

La posición de la casilla es la dirección de la casilla – dirección del tablero

Con A0 recorreremos una a una las casillas hasta encontrar la ficha buscada, empezando por A0+D1;

En este caso necesitamos guardar la dirección del tablero para al final hacer la resta

El bucle lo repetiremos hasta llegar al final del tablero si no se encuentra la ficha antes; el número de veces que habrá que repetirlo será \$40-D1

```
;*****  
; Buscar  
; Busca una ficha en el tablero a partir de cierta posición  
; Entradas:  
; D0: Color  
; A0: Tablero  
; D1: Posición inicial  
; Salidas:  
; D0: Posición, No encontrado($FF)  
;*****  
ERROR= 0FF  
TAMANO_TABLERO = 40  
  
Buscar_variables = 2000  
buscar_tablah space 1  
buscar_tablal space 1  
buscar_color space 1  
buscar_inicio space 1  
  
Buscar  
mov h0 buscar_tablah  
mov l0 buscar_tablal
```

```

mov    d0 buscar_color
mov    d1 buscar_inicio

mov    d1 d0      ; Calculo la dirección donde empezar a buscar
call  Incrementar

mov    buscar_inicio d1; Calculo el número repeticiones del bucle
mov    #TAMANO_TABLERO d0
sub
mov    d0 l1      ; utilizo l1 como contador

Buscar_seguir
mov    #0 d0      ; ¿fin de la tabla?
mov    l1 d1
cmp
jr     NS Buscar_fin_tabla

mov    buscar_color d1 ; Color buscado
mov    (a0) d0      ; Leo la casilla
cmp
jr     Z Buscar_encontrado

dec    a1
inc    a0
jmp    Buscar_seguir

Buscar_encontrado
; en A0 está la dirección de la casilla
; Hay que calcular la posición en la tabla.
mov    buscar_tablah h1
mov    buscar_tablal l1
call  Restar_direcciones ; una subrutina que A0 = A0-A1
mov    l0 d0
jmp    Buscar_volver

Buscar_fin_tabla
mov    #ERROR d0

Buscar_volver
ret

```

1.3. Contar

Se pide redactar una subrutina en Ensamblador CH-2020 que tenga:

Entradas

- En D0 un color de ficha (R, G, B o Y)
- En A0 la dirección donde se encuentra el tablero

- En D1 un valor de inicio (< \$40)

Salidas

- D0 la cuenta buscada

Operación

Cuenta, a partir de la casilla indicada en D1 (inclusive), el número de fichas del color indicado en D0 y devuelva el resultado en D0.

Solución

En este ejercicio voy a utilizar otra forma de controlar el fin de las repeticiones utilizando, no un contador sino la dirección del final de la tabla

```
;*****
; Contar
;   Entradas:
;       D0: Color
;       A0: Tablero
;       D1: Posición inicial
;   Salidas:
;       D0: Cuenta
;*****
TAMANO_TABLERO = 40

Contar_variables = 2010
contar_tablah   space 1
contar_tablal   space 1
contar_color    space 1
contar_inicio   space 1
contar_contador space 1

Contar
    mov  h0 contar_tablah
    mov  l0 contar_tablal
    mov  d0 contar_color
    mov  d1 contar_inicio

    mov  #TAMANO_TABLERO D0      ; Calcular el final del tablero,
    call Incrementar            ; la dirección siguiente a la
                                ; última casilla, y guardarlo en A1

    mov  a0 a1

    mov  contar_tablah h0      ; Calcular la dirección de inicio
    mov  contar_tablal l0
    mov  contar_inicio d0
    call Incrementar

Contar_repetir
    mov  h0 d0                  ;comparar a0 y a1
    mov  h1 d1                  ; saltar a contar terminar
```

```

        cmp                    ; cuando son iguales
        jr nz Contar_seguir
        mov l0 d0
        mov l1 d1
        cmp
        jr z Contar_terminar

Contar_seguir
        mov (a0) d0           ; Incrementar contador si la casilla
        mov contar_color d1  ; tiene el color buscado.
        cmp
        jr nz Contar_avanzar
        mov contar_contador d0
        mov #1 d1
        add
        mov d0 contar_contador
Contar_avanzar
        inc a0
        jmp Contar_repetir

Contar_terminar
        mov contar_contador d0
        ret

```

1.4. Programa

Redactar un programa en Ensamblador del CH-2020 que utilizando las subrutinas construidas previamente

- Vacíe el tablero
- Ponga las fichas en las posiciones indicadas en el array de cada color (ver más adelante)
- Cuente el número de fichas rojas ubicadas más allá de la posición \$12 y deje el resultado en **cuenta**
- Busque la primera ficha de color azul que se encuentre después de la primera ficha de color amarillo y guarde su posición en **casilla_azul**

Restricciones:

El tablero se ubicará en la dirección \$12FA

Los arrays de colores serán:

```

fichas_rojas      BYTE 0 23 30 3F
fichas_azules     BYTE 10 20 22 30
fichas_verdes     BYTE 1A 1F 2C 33
fichas_amarillas  BYTE 22 24 31 3C

```

ubicados a partir de la dirección \$1100

Las variables: cuenta, casilla_azul y otras que el programador necesite, se ubicarán a partir de la dirección \$1000

Se utilizarán etiquetas de forma masiva.

Solución

```
;*****
;* Tablero
;*****

;*****
;Constantes
;*****
TAMANO_TABLERO = 40
NUMERO_FICHAS = 4

OK = 0
ERROR = 0FF

ROJO = 52
VERDE = 47
AZUL = 42
AMARILLO = 59
VACIO = 0

Constantes = 1000
;*****
;Valores
;*****

fichas_rojas      BYTE 0 23 30 3F
fichas_azules    BYTE 10 20 22 30
fichas_verdes    BYTE 1A 1F 2C 33
fichas_amarillas BYTE 22 24 31 3C

;*****
;Variables
;*****
Variables = 1100

tablero  space TAMANO_TABLERO
cuenta   space 1
casilla_azul space 1

;Buscar_variables
buscar_tablah space 1
buscar_tablal space 1
buscar_color  space 1
buscar_inicio space 1

;Contar_variables
contar_tablah space 1
contar_tablal space 1
```



```

contar_color    space 1
contar_inicio   space 1
contar_contador space 1
contar_condador space 1

;Cargar_tablero
cargar_color    space 1
cargar_contador space 1
cargar_ficha_h  space 1
cargar_ficha_l  space 1

;*****
;Variables
;*****
pila = 4000

;*****
;Variables
;*****
Reset = 0
      JMP  Inicio

Inicio = 100
      mov #MSB(pila) h0 ;Obligatorio siempre que se utilice la pila
      mov #LSB(pila) l0 ; como es el caso de la llamada a subrutinas
      mov  a0 sp

      call Vaciar_tablero
      call Cargar_fichas_rojas
      call Cargar_fichas_verdes
      call Cargar_fichas_azules
      call Cargar_fichas_amarillas

;Contar fichas rojas
mov  #ROJO d0
mov  #MSB(tablero) h0
mov  #LSB(tablero) l0
mov  #12 d1
call Contar
mov  d0 cuenta

;Buscar primera ficha amarilla
mov  #AMARILLO D0
mov  #MSB(tablero) h0
mov  #LSB(tablero) l0
mov  #0 d1
call Buscar

;Buscar siguiente azules
mov  d0 d1
mov  #AZUL d0

```

```
mov #MSB(tablero) h0
mov #LSB(tablero) l0
call Buscar
mov d0 casilla_azul
```

```
STOP
```

```
;*****
;Cargar fichas
;*****
```

```
; Cargar tablero con fichas rojas
```

```
Cargar_fichas_rojas
```

```
mov #MSB(fichas_rojas) h0
mov #LSB(fichas_rojas) l0
mov #ROJO d0
mov #NUMERO_FICHAS d1
call Cargar_tablero
ret
```

```
; Cargar tablero con fichas verdes
```

```
Cargar_fichas_verdes
```

```
mov #MSB(fichas_verdes) h0
mov #LSB(fichas_verdes) l0
mov #VERDE d0
mov #NUMERO_FICHAS d1
call Cargar_tablero
ret
```

```
; Cargar tablero con fichas azules
```

```
Cargar_fichas_azules
```

```
mov #MSB(fichas_azules) h0
mov #LSB(fichas_azules) l0
mov #AZUL d0
mov #NUMERO_FICHAS d1
call Cargar_tablero
ret
```

```
; Cargar tablero con fichas amarillas
```

```
Cargar_fichas_amarillas
```

```
mov #MSB(fichas_amarillas) h0
mov #LSB(fichas_amarillas) l0
mov #AMARILLO d0
mov #NUMERO_FICHAS d1
call Cargar_tablero
ret
```

```
;*****
```

```
;Vaciar_tablero
```

```
; Entradas: ninguna
```

```
; Salidas: ninguna
```

```
; Proceso:
```

```

;           Rellenar el tablero con VACIO
;*****
Vaciar_tablero
; Vaciar el tablero
mov  #TAMANO_TABLERO l1 ;voy a utilizar l1 como (des)contador
mov  #MSB(Tablero) h0   ; A0 como puntero a las casillas
mov  #LSB(Tablero) l0

Vaciar_seguir
mov  #VACIO d0 ;Vacío la casilla actual: (A0)
mov  d0 (a0)
inc  a0          ; Avanzo a la siguiente casilla
dec  a1          ; descuento el descontador
clc
mov  h1 d0       ; ¿Hemos llegado al final?
mov  #0 d1
cmp
jr   nz Vaciar_seguir
ret

;*****
;Cargar_tablero
;   Entradas:
;       D0: Color
;       D1: Número de fichas
;       A0: lista de casillas
;   Salidas: ninguna
;   Proceso:
;       La lista de casillas contiene las posiciones del tablero
;       donde colocar las Numero de fichas del Color indicado
;*****
Cargar_tablero
mov  d0 cargar_color      ; Guardar los argumentos
mov  d1 cargar_contador
mov  h0 cargar_ficha_h
mov  l0 cargar_ficha_l

Cargar_repetir
mov  cargar_ficha_h h1 ; A1 = Dirección de la ficha actual
mov  cargar_ficha_l l1
mov  (a1) d1
inc  a1          ; Calcular dirección siguiente ficha
mov  h1 cargar_ficha_h ;
mov  l1 cargar_ficha_l

mov  cargar_color d0
mov  #MSB(tablero) h0   ; A0 = Dirección de la casilla
mov  #LSB(tablero) l0
call Colocar

mov  cargar_contador d0 ; Control de numero de repeticiones
mov  #1 d1

```

```

    sub
    mov  d0 cargar_contador
    jr   nz Cargar_repetir

    ret

;*****
; COLOCAR
;   Entradas:
;       D0: Color
;       A0: Tablero
;       D1: Posición
;   Salidas:
;       D0: Resultado OK ($00), Error ($FF)
;*****
Colocar
    push d0                ; Salvo D0 para posterior uso.
    mov  d1 d0             ; Obtener la dirección de la casilla.
    call Incrementar
    mov  (a0) d0           ; Verificar que está vacía.
    mov  #0 D1
    cmp
    pop  d0                ; Recupero el valor del color de la
ficha.
    jr   NZ Colocar_error
    mov  d0 (a0)           ; Dejo la nueva ficha en la casilla
vacía.

    mov  #OK d0            ; Devolver OK
    jmp  Colocar_fin

Colocar_error
    mov  #ERROR d0        ; Cuando la celda está ocupada
;       devolver error

Colocar_fin
    ret

;*****
; Buscar
;   Busca una ficha en el tablero a partir de cierta posición
;   Entradas:
;       D0: Color
;       A0: Tablero
;       D1: Posición inicial
;   Salidas:
;       D0: Posición, No encontrado($FF)
;*****
Buscar
    mov  h0 buscar_tablah
    mov  l0 buscar_tablal

```

```

mov    d0 buscar_color
mov    d1 buscar_inicio

mov    d1 d0      ; Calculo la dirección donde empezar a buscar
call  Incrementar

mov    buscar_inicio d1; Calculo el número repeticiones del bucle
mov    #TAMANO_TABLERO d0
sub
mov    d0 l1      ; utilizo l1 como contador

```

Buscar_seguir

```

mov    #0 d0      ; ¿fin de la tabla?
mov    l1 d1
cmp
jrc   NS Buscar_fin_tabla

mov    buscar_color d1 ; Color buscado
mov    (a0) d0      ; Leo la casilla
cmp
jrc   Z Buscar_encontrado

dec    a1
inc    a0
jmp    Buscar_seguir

```

Buscar_encontrado

```

; en A0 está la dirección de la casilla
; Hay que calcular la posición en la tabla.
mov    buscar_tablah h1
mov    buscar_tablal l1
call  Restar_direcciones ; una subrutina que A0 = A0-A1
mov    l0 d0
jmp    Buscar_volver

```

Buscar_fin_tabla

```

mov    #ERROR d0

```

Buscar_volver

```

ret

```

```

;*****
; Contar
; Entradas:
;     D0: Color
;     A0: Tablero
;     D1: Posición inicial
; Salidas:
;     D0: Cuenta
;*****

```

Contar

```

mov    h0 contar_tablah

```

```

    mov    l0 contar_tabla1
    mov    d0 contar_color
    mov    d1 contar_inicio

    mov    #TAMANO_TABLERO D0      ; Calcular el final del tablero,
    call Incrementar                ; la dirección siguiente a la
                                    ; última casilla, y guardarlo en A1

    mov    a0 a1

    mov    contar_tabla_h h0        ; Calcular la dirección de inicio
    mov    contar_tabla_l l0
    mov    contar_inicio d0
    call Incrementar

Contar_repetir
    mov    h0 d0                    ; comparar a0 y a1
    mov    h1 d1                    ; saltar a contar_terminar
    cmp
    jr    nz Contar_seguir
    mov    l0 d0
    mov    l1 d1
    cmp
    jr    z Contar_terminar

Contar_seguir
    mov    (a0) d0                  ; Incrementar contador si la casilla
    mov    contar_color d1         ; tiene el color buscado.
    cmp
    jr    nz Contar_avanzar
    mov    contar_contador d0
    mov    #1 d1
    add
    mov    d0 contar_contador

Contar_avanzar
    inc    a0
    jmp    Contar_repetir

Contar_terminar
    mov    contar_contador d0
    ret

;*****
; Incrementar
; Entradas:
;     A0: Dirección
;     D0: Incremento
; Salidas:
;     A0: Dirección incrementada A0 = A0 + D0
;*****
Incrementar
    clc
    mov    L0 D1

```

```
add
mov  D0 L0
mov  #0 D0
mov  H0 D1
add
mov  D0 H0
ret
```

```
;*****
; Restar_direcciones
; Entradas:
;     A0: Direccion
;     A1: Incremento
; Salidas:
;     A0: Dirección incrementada  $A0 = A0 - A1$ 
;*****
```

```
Restar_direcciones
  clc
  mov  10 d0
  mov  11 d1
  sub
  mov  d0 10
  mov  10 d0
  mov  11 d1
  sub
  mov  d0 h0
  ret
```