Práctica nº: 3 Fecha: 30/3/20

Autor: Zdravko Dimitrov Arnaudov

Realizada junto con (si procede):

Introducción

En el desarrollo 3 contamos con una matriz Mi,j de dimensiones 8x9 por enteros de 4B. El objetivo del ejercicio es conseguir acceder al valor de un elemento Ei,j cuyos parámetros serán introducidos por teclado.
Para ello, una vez definida la matriz, se inicializa por medio de un bucle que la recorre por filas y columnas asignando a cada elemento el valor que toma una variable acumulativa a medida que avanzan las iteraciones. Una vez se dispone del elemento buscado, guardamos su valor en una variable “VALOR” en memoria.

Para el desarrollo 4 se solicita realizar una consulta sobre una base de datos que alberga 1106 películas junto con sus correspondientes títulos, año de estreno, número de votos y nota media asociada. El ejercicio solicita imprimir por pantalla el título de aquellas películas que tengan 50 votos o más y superen una nota de corte establecida por el programador. Previamente, es preciso completar el campo nota media de las películas que no fueron valoradas con la nota media de todas aquellas que si lo fueron.
Para ello, precisaremos de instrucciones de punto flotante para trabajar con los datos antes mencionados.

Pseudo-Código

Desarrollo 3:

```
puntero = MATRIZ
@inicializamos matriz
for (int i = 0; i < FILAS ; i++){
        for (int j = 0; j < COLUMNAS; j++){
                MATRIZ[i][j] = acumula
                puntero ++
                acumula++
        }
}

@leemos parametros
while (i != valor_numerico){
i = swi_read()
}
```

```
while (j !=valor_numerico){
j = swi_read()
}

@calculamos desplazamiento
desp = (i*COLUMNAS +j)*T_ELEM
puntero = puntero + desp
Ei,j = puntero*

@salvamos valor
VALOR = Ei,j
swi_termina()
```

Desarrollo 4:

```
puntero = PELICULAS
@comprueba cuántas películas no han sido valoradas y cuales cuentan con una nota media superior a 9
for (int i = 0; i < NUM_PELICULAS; i++){
        if (PELICULAS[i].NOTA_MEDIA == 0){
                películas_no_valoradas ++

        } else {
                if (PELICULAS[i].NOTA_MEDIA > 9){
                        películas_excelentes ++
                }
                nota_media = PELICULAS[i].NOTA_MEDIA
        }
        puntero ++
}

@salvamos los datos obtenidos para responder a las preguntas propuestas al final del ejercicio
películas_no_valoradas en memoria
películas_excelentes en memoria
nota_media en memoria

@calculamos la media de las valoraciones de aquellas películas que han recibido votos
num_peliculas_valoradas = NUM_PELICULAS - peliculas_no_valoradas
div = notas_medias / (num_peliculas_valoradas)

puntero =PELICULAS
for (int j = 0; j< NUM_PELICULAS; j++){

        if (PELICULAS[j].NOTA_MEDIA == 0){
```

```
                        PELICULAS[j].NOTA_MEDIA = div

                } else {
                        if (PELICULAS[j].VOTOS >= 50 &&
PELICULAS[j].NOTA_MEDIA > NOTA_CORTE){
                                PELICULAS[j].TITULO = swi_print()
                        }
                }
                puntero ++
        }
        swi_termina()
```

Código

Desarrollo 3:

```
.data
.equ FILAS, 8
.equ COLUMNAS, 9
.equ T_ELEMENTO, 4
.align 2
VALOR: .space 4
MATRIZ: .space FILAS*COLUMNAS*T_ELEMENTO
.align 2
intro: .asciz "Busqueda de un elemento en matriz en base a los parametros i y j"
lee_i: .asciz "Introduce el parametro i:"
lee_j: .asciz "Introduce el parametro j:"
.text
.globl _start
_start:
    ldr r0, =MATRIZ
    mov r2,#0 @acumula
    mov r3, #0 @i=0
    mov r5, #COLUMNAS
    mov r6, #T_ELEMENTO

INI_FILAS:
    cmp r3, #FILAS
    beq FIN_INI
    mov r4, #0 @j=0

INI_COLUMNAS:
    cmp r4, #COLUMNAS
    beq ACT_FILAS
    @calculamos el desplazamiento
```

```
    mul r1, r3, r5 @i*COLUMNAS
    add r1, r1, r4 @ (I*COLUMNAS +j)
    mul r1, r6 ,r1 @ (i*COLUMNAS +j) * T_ELEM
    add r1, r0, r1 @ dir base + desp
    str r2, [r1] @inicializamos el valor del elemento con la variable incremental acumula
    add r4 ,r4, #1 @j++
    add r2, r2, #1 @acumula++
    B INI_COLUMNAS

ACT_FILAS:
    add r3, r3, #1 @i++
    b INI_FILAS

FIN_INI:

    @leemos la i y la j por teclado
    ldr r0, =intro
    swi 0x02 @imprime mensaje introductorio
    mov r0, #'\n'
    swi 0x00

    ldr r0, =lee_i
    swi 0x02
    mov r0, #'\n'
    swi 0x00

LEE_I:
    swi 0x04
    cmp r0, #0x30
    blt LEE_I
    cmp r0, #0x39
    bhi LEE_I
    sub r0, r0, #0x30 @convertimos a numero
    mov r7, r0

    ldr r0, =lee_j
    swi 0x02
    mov r0, #'\n'
    swi 0x00

LEE_J:
    swi 0x04
    cmp r0, #0x30
    blt LEE_J
    cmp r0, #0x39
    bhi LEE_J
```

```
    sub r0, r0, #0x30 @convertimos a numero
    mov r8, r0

    ldr r0, =MATRIZ
    @ahora podemos acceder al elemento y guardar su valor
    mul r1, r7, r5 @ i*COLUMNAS
    add r1, r1, r8 @ (i*COLUMNAS +J)
    mul r1, r6, r1 @ (i*COLUMNAS +J) * T_ELEM
    add r1, r1, r0 @ dir base + desplazamiento

    ldr r1, [r1] @valor de elemento
    ldr r2, =VALOR
    str r1, [r2] @escribimos sobre variable
    swi 0x11
```

Desarrollo 4:

```
.data
.equ LIM_VALORACIONES, 50
.align 2
NOTA_CORTE: .float 5.0
PELICULAS_NO_VALORADAS: .space 4
PELICULAS_EXCELENTES: .space 4
VALORACIONES_MEDIAS: .space 4
NOTA_MEDIA_VALORACIONES: .space 4
.text
.include "baseDatos.s"
.globl _start
_start:
    ldr r0, =PELICULAS @puntero a peliculas
    mov r1, #0 @i = 0
    mov r2, #0 @acumula valoraciones medias
    vmov s2, r2 @mover a registro de punto flotante

    mov r3, #0 @cuenta peliculas no valoradas
    mov r4, #0 @peliculas con nota media superior a 9

    mov r6, #9 @para comparar la nota media con 9.0
    vmov s6, r6
    vcvt.f32.s32 s6, s6 @convertimos de entero a flotante

    ldr  r7,=N_PELICULAS @salvamos numero de peliculas para los bucles

BUCLE1:
    cmp r1, r7
```

```
        beq FIN_BUCLE1

        vldr s5, [r0, #NOTA_MEDIA]
        vcmp.f32 s5, #0
        FMSTAT
        addeq r3, r3, #1 @anhade pelicula no valorada
        beq ACTUALIZA_1

        vcmp.f32 s5, s6
        FMSTAT
        addhi r4, r4, #1 @anhade pelicula con nota mayor que 9
        vadd.f32 s2, s2, s5

ACTUALIZA_1:
        add r0, r0, #L_PELICULA @actualizamos puntero
        add r1, r1, #1 @i++
        b BUCLE1

FIN_BUCLE1:

        @salvamos los resultados
        ldr r1, =PELICULAS_NO_VALORADAS
        str r3, [r1]
        ldr r1, =PELICULAS_EXCELENTES
        str r4, [r1]
        ldr r1, =VALORACIONES_MEDIAS
        vstr s2, [r1]

        @notas medias y N peliculas valoradas a punto flotante para la division
        mov r5, r7
        sub r5, r5, r3 @peliculas valoradas
        vmov s3, r5
        vcvt.f32.s32 s3, s3 @numero peliculas totales de entero a flotante
        vdiv.f32 s1, s2, s3

        ldr  r1, =NOTA_MEDIA_VALORACIONES
        vstr s1, [r1] @salvamos la media de las valoraciones puntuadas

        ldr r2, =PELICULAS
        mov r3, #0 @i = 0
        ldr r5, =NOTA_CORTE
        ldr r5, [r5]
        vmov s5, r5

BUCLE2:
        cmp r3, r7
```

```
        beq FIN_BUCLE2

        vldr s4, [r2, #NOTA_MEDIA]
        vcmp.f32 s4, #0
        FMSTAT
        vstreq s1, [r2, #NOTA_MEDIA] @completamos si pelicula no valorada
        beq ACTUALIZA_2

        ldr r4, [r2, #NUM_VOTOS]
        cmp r4, #LIM_VALORACIONES @buscamos peliculas con votos >=50
        blt ACTUALIZA_2

        vldr s4, [r2, #NOTA_MEDIA]
        vcmp.f32 s4, s5 @buscamos peliculas que superen la nota de corte
        FMSTAT
        blt ACTUALIZA_2

        @imprime el titulo de las peliculas filtradas
        mov r0, R2
        add r0, r0 , #TITULO
        swi 0x02 @imprime por pantalla el titulo de la pelicula

        mov r0, #'\n' @salto de linea
        swi 0x00

ACTUALIZA_2:
        add r2, r2, #L_PELICULA
        add r3, r3, #1 @i++
        b BUCLE2


FIN_BUCLE2:
        swi 0x11
```

Observaciones

Para el tercer desarrollo es importante destacar que he decidido inicializar la matriz de la forma propuesta, usando un bucle en el código y asignando el valor que toma una variable, que incrementa 1 por iteración, a cada elemento. También, para comprobar que el elemento buscado es el correcto, introducimos los parámetros i = 0 y j = 1 con lo que obtendríamos un valor de 1 y efectivamente funciona.
Me ha parecido importante gestionar los posibles errores que puedan surgir a la hora de introducir los parámetros solicitados, con lo cual el programa volverá a pedirlos en caso de que se introduzca por teclado cualquier valor que no sea un numero del 0 al 9.

Respecto al cuarto y último desarrollo presentado es necesario explicar más cosas.
Para empezar, con motivo de poder usar la base de datos resulta imprescindible dar valor a los campos definidos y que dan cuerpo a la estructura de cada película.
Por lo tanto y teniendo en cuenta las dimensiones de cada campo, definimos la estructura así:

```
.equ TITULO, 0
.equ ESTRENO, 62
.equ NOTA_MEDIA, 64
.equ NUM_VOTOS, 68
.equ N_PELICULAS, 1106
.equ L_PELICULA, 72
```

A continuación con el programa y con intención de poder responder a las preguntas planteadas, creamos un primer bucle en el que contamos el numero de películas no valoradas y el numero de películas con una valoración superior a 9 obteniendo 392 y 45 respectivamente.
También, acumulamos las notas medias de aquellas películas que hayan sido valoradas para poder hacer la media más adelante. Cabe destacar que estos resultados son revisados, en depuración, al finalizar el bucle salvándolos en variables en memoria.

Después, realizamos la media con las valoraciones totales antes obtenidas y el número de películas valoradas, obteniendo un resultado aproximado de 6.72. Este dato será usado posteriormente para completar el campo de valoración de aquellas películas que no fueron puntuadas.

Por último, hacemos un segundo bucle en el que asignamos la media antes calculada a las películas que lo precisen e imprimimos por pantalla el título de aquellas películas que tengan 50 o más votos y superen la nota de corte escogida por el programador.

Podemos observar que este ejercicio se enmarca en las instrucciones de punto flotante, las cuales han sido utilizadas para manejar las notas medias de cada película y así realizar comparaciones y operaciones necesarias con dicho dato.
Me ha parecido conveniente usar, en la medida de lo posible, los mismos números para los registros de punto flotante y los de propósito general con tal de identificarlos mejor.
También, a la hora de realizar comparaciones con los registros de punto flotante, he decidido usar “FMSTAT”, por simplicidad, para copiar los ‘flags’ del registro FPSCR de la unidad de punto flotante al CPSR para poder realizar ejecución condicional posteriormente.

Para comprobar que el código funciona, además de depurarlo y comprobar que las operaciones y resultados son válidos, localizamos los títulos impresos por pantalla y con la base de datos abierta, revisamos si efectivamente la consulta es satisfactoria según los filtros seleccionados.
Para ello, hemos ejecutado el programa desde la ‘TaskWindow’ debido a que por motivos de almacenamiento, el depurador no puede mostrar todos desde la ventana emergente.