MEMORIA ESTRUCTURA DE COMPUTADORES

Jason Felipe Vázquez

23 DE DICIEMBRE DE 2021
UPM

Contenido

Día 1(Comienzo el día 16 de noviembre)

Se empezó con la programación de la subrutina Sqrt1d, además el orden en el que se van a realizar las subrutinas es según el orden del .pdf "Presentación 2020/21".

En cuanto a sqrt1d no hubo mucho problema, ya que se ha seguido a raja tabla el algoritmo proporcionado.

Día 2

Se empezó a comprobar la subrutina programada, primero con las pruebas proporcionadas, y también pruebas realizadas por mí mismo. De la siguiente manera:

```
org             0x5000
Num:    data    0x65541; para realizar más pruebas solo modifico esto
prueba_0:
                or          r30, r0, 0x9000 ; Inicializo la pila en la 2osición 0x9000
                xor         r20, r20, r20   ; Limpio el contenido de r20
                LOAD    (r20, Num)          ; r20 = Num
                PUSH    (r20)               ; Paso el 2osición2 Num por pila
                bsr         Sqrt1d
                addu    r30, r30, 4         ; Limpio la pila
                stop
```

Como los resultados parecían correctos tras comprobarlo con el compilador, se dio por terminado esta subrutina.

Día 3

Se empezó y se terminó de programar la subrutina nFiltrados donde solo tiene una variable de entrada que se le pasa por valor y además del uso de una variable estática Nf que se encuentra en la dirección 0, el resto del programa se realizó siguiendo el algoritmo descrito.

Día 4

Se volvió a seguir la metodología anterior de comprobar la subrutina, donde se programó una sección de pruebas para este caso.

```
Org 0
nF:             res 4
; ***    Pruebas de nFiltrados    ***
prueba_1:
        or              r30, r0, 0x9000 ; Inicializo la pila en la 3osición 0x9000
        xor             r20, r20, r20   ; Limpio el contenido de r20
        or              r20, r0, 14             ; r20 = 14 > 0; modifico eso si quiero realizar
más pruebas
        PUSH    (r20)                   ; Inserto r20 = oper en la pila
        bsr             nFiltrados
        LOAD    (r20, nF)               ; Se muestra en r20 el contenido de la 3osición3
        addu    r30, r30, 4             ; Limpio la pila
        stop
```

Día 5

Comencé con el desarrollo de la subrutina Comp, donde solo tenía 2 parámetros de entrada pasadas por dirección Imagen1 e Imagen 2, donde Imagen1 además contiene 3 campos (nº filas, nº columnas, y elementos almacenados por filas MxN). El resto de la subrutina de desarrollo siguiendo los pasos del algoritmo, puesto en la "descripción". Como esta subrutina tiene que llamar a sqrtd1y ya se comprobó, se espera que no haya ningún problema.

Día 6

Se realizaron las pruebas para Comp:

```
IMAGEN1:
                data    4, 8
                data    0x000000D1, 0x00000000
                data    0x00000000, 0x1D7A0000
                data    0x00000001, 0x00000000
                data    0x00000000, 0xFF000000
IMAGEN2:
                data    4, 8
```

```
                data    0x00000002, 0x00000000
                data    0x00000000, 0x82010000
                data    0x000000FF, 0x00000000
                data    0x00000000, 0x10000000
; ***    Pruebas de Comp        ***
prueba_4:
        or              r30, r0, 0x9000 ; Inicializo el puntero de pila en la 4osición 0x9000
        xor             r25, r25, r25   ; r25 = M (Numero de filas)
        xor             r26, r26, r26   ; r26 = N (numero de pixeles por fila)
        LEA             (r15, IMAGEN1); dir. Imagen1
        LEA             (r16, IMAGEN2); dir. Imagen2
        ;Paso los dos parametros por pila a la subrutina Comp
        PUSH    (r16)                   ; Inserto la dir. De Imagen2
        PUSH    (r15)                   ; Inserto la dir. De Imagen1
        bsr
        Comp                    ; Salto a subrutina
        addu    r30, r30, 8             ; Destruyo la pila
        stop
```

Para realizar más pruebas para esta subrutina solo se debe cambiar las imágenes, que es lo que he estado realizando.

Día 7

Comienzo el desarrollo de la siguiente subrutina SubMAtriz, la cual tiene 4 parámetros de entrada (Imagen, SubImg, i, j), pero se complicó ya que tuve problemas para recorrer matrices.

Día 8

Una vez aprendido el funcionamiento de como recorrer matrices en este lenguaje, empecé con la realización del algoritmo, recorriendo la matriz por filas, y rellenando la submatriz resultado tal como se especifica.

Día 9

Se empezó el desarrollo de las pruebas para submatriz:

IMAGEN1:

```
                data    4, 8
                data    0x000000D1, 0x00000000
                data    0x00000000, 0x1D7A0000
                data    0x00000001, 0x00000000
                data    0x00000000, 0xFF000000

; ***    Pruebas de SubMatriz   ***
prueba_6:
        or              r30, r0, 0x9000 ; Inicializo el puntero de pila en la 5osición 0x9000
        LEA             (r20, IMAGEN)   ; r20 = dir. Imagen
        LEA             (r21, SUBIMAGEN1); r21 = dir. Subimagen
        ; Elemento centrar de una matriz 3x3
        or              r22, r0, 1              ; r22 = 1 = i
        or              r23, r0, 1              ; r23 = j = 1
        PUSH    (r23)                   ; Inserto j en la pila (por valor)
        PUSH    (r22)                   ; Inserto i en la pila (por valor)
        PUSH    (r21)                   ; Inserto la dir. SubImagen en la pila
        PUSH    (r20)                   ; Inserto la dir. SubImagen en la pila
        bsr             SubMatriz
        addu    r30, r30, 16    ; Se eliminan los parámetros de la pila
        stop
```

Igual que las anteriores, solo basta con cambiar la Imagen para varías las pruebas. Cabe destacar que no sabía si las pruebas realizadas eran correctas, aunque según los casos de pruebas proporcionadas, parecía que sí.

Día 10

Por falta de tiempo y por mala organización mía, no desarrollé los dos primeros hitos por lo que el objetivo era realizar el máximo número de subrutinas, aunque siempre comprobándolas de alguna manera de antemano, ya que no tenía las pruebas extras de los hitos, también hay que destacar que como es una asignatura la cuál estoy cursando de nuevo, ya tengo cierta experiencia con este ensamblador.

Día 11

Se empezó y terminó la siguiente subrutina, ValorPixel, con 2 parámetros de entrada (SubImagen, Mfiltro) pasadas por dirección, donde el objetivo es aplicarle un filtro al pixel y devolver en r29 una aproximación al valor de este pixel.

Seguí los pasos descritos para este algoritmo, y como tampoco llama a ninguna subrutina auxiliar, no hubo problemas con su realización.

Día 12

Se realizaron las pruebas correspondientes a esta subrutina, realizando también los casos pruebas proporcionados para comprobar su funcionamiento:

```
SUBIMAGEN:
        data    0x13121110, 0x17161514, 0x18
MFILTRO:
        data    2,      0xFFFFFFFF,     0xFFFFFFFE,     1,      2,      0xFFFFFFFF
        data    0xFFFFFFFE,     1,      0,      1,      0xFFFFFFFE,     1
        data    2,      0xFFFFFFFF,     0xFFFFFFFE,     1,      2,      0xFFFFFFFF
; ***   Pruebas de ValorPixel   ***
prueba_5:
        or              r30, r0, 0x9000 ; Inicializo el puntero de pila en la posicion 0x9000
        LEA             (r21, SUBIMAGEN); r20 = dir. SubImg
        LEA             (r20, MFILTRO) ; r21 = dir. MFiltro
        PUSH    (r20)                   ; Se pasa la dir. de MFiltro por pila
        PUSH    (r21)                   ; Se pasa la dir. de SubImg por pila
        bsr             ValorPixel
        addu    r30, r30, 4             ; Se eliminan los parametros de pila
        stop
```

Día 13

Se realizaron pruebas para todas estas subrutinas realizadas hasta el momento, ya que no quería realizar el resto de subrutinas (FiltPixel, Filtro y FiltRec) sin comprobar las ya hechas hasta el momento, debido a que estas nuevas llamaban a todas las anteriores subrutinas.

Por tanto, se realiza el envío de las pruebas el día 09/12/2020, donde esa misma noche comprobé las pruebas superadas. Y tenía un error en la subrutina sqrtd1 y cmp, según las pruebas.

Este fue el error citado:

Su implementación falla en una prueba en que se hace lo siguiente:

- Llama a 'Sqrt1d', pasándole un parámetro tal que durante la ejecución se tiene que usar el intercambio a<-->b del paso 3.b

Se queda en un bucle infinito

Su implementación falla en una prueba en que se hace lo siguiente:

- Llama a 'Comp', pasándole dos imágenes de 4x5 elementos en las que difieren varios de ellos.

Se queda en un bucle infinito

Al estar en bucle infinito doy por hecho que el error puede estar en sqrtd1 ya que Comp la llama a sqrtd1, y probablemente la hace fallar también, haciéndola quedar en bucle.

Día 13

Empecé a buscar el error que hacía que sqrtd1 se metiera en bucle infinito.

Se encontró con facilidad, ya que era un error leve, y fue el siguiente:

```
bb0     le, r5, BUCL_SQRT     ; Salta a iterar el bucle--→ MAL
bb1     gt, r5, BUCL_SQRT     ; Salta a iterar el bucle → BIEN
```

Como se puede observar fue una pequeña errata.

Ese mismo día se volvió a entregar otra vez el proyecto al gestor de prácticas, y se comprobó que todo iba correcto en cuanto a las subrutinas que se llevaban hasta el momento.

Día 14

Empezamos con FilPixel, donde los parámetros de entrada son los siguientes;

;Imagen -> Parámetro de entrada. Se pasa por dirección (M filas, N columnas, elementos-ij)

; i -> Parámetro de entrada. Numero de fila. Se pasa por valor

; j -> Parámetro de entrada. Numero de columna. Se pasa por valor

; MFiltro -> Parámetro de entrada. Se pasa por dirección

Donde esta subrutina básicamente lo que realiza es la aplicación de la máscara que está ya filtrada al pixel definido por i y j.

Esta rutina llama a Submatriz para construir una matriz cuadrada a partir del pixel que se filtra.

Realicé todo el algoritmo tal cual descrito.

Día 15

Se realizaron las correspondientes pruebas para FilPixel, y se realizo de la siguiente manera:

```
;------Prueba Filpixel---------
IMAGEN_FP:
        data    5, 5
        data    0x44332211, 0x03020155
        data    0x22210504, 0x31252423
        data    0x35343332,     0x44434241
        data    0x00000045
FILTRO_FP:
        data    0, 1, 0, 1, 0, 1
        data    0, 1, -5, -5, 0, 1
        data    0, 1, 0, 1, 0, 1
prueba_8:
        or              r30, r0, 0x9000 ; Inicializo el puntero de pila en la posicion 0x9000
        LEA             (r20, IMAGEN_FP)
        LEA             (r21, FILTRO_FP)
        or              r22, r0, 2              ; i = 2
        or              r23, r0, 3              ; j = 3
        PUSH    (r21)
        PUSH    (r23)
        PUSH    (r22)
        PUSH    (r20)
        bsr             FilPixel
        addu    r30, r30, 16
        stop
```

Día 16

Se comenzó con la penúltima subrutina, Filtro, el objetivo de esta subrutina es aplicar la máscara filtrada Imagen, y dejando el resultado final en IMFiltrada.

Esta subrutina llamará a FilPixel para aplicar la máscara del filtrado al pixel que se desee. El resto del algoritmo es el descrito en el enunciado.

Día 17

Pruebas realizadas para Filtro;

```
; ***     Pruebas de Filtro          ***

prueba_9:
        or          r30, r0, 0x9000 ; Inicializo el puntero de pila en la posicion 0x9000

        LEA         (r20, FILTRO_F)
        LEA         (r21, FILTRADA_F)
        LEA         (r22, IMAGEN_F)
        PUSH    (r20)
        PUSH    (r21)
        PUSH    (r22)
        bsr         Filtro
        addu    r30, r30, 12
        stop
```

Además, compruebo con los casos de pruebas proporcionados en la página web.

Día 18

Se envío otra prueba al gestor de errores y ese mismo día se consulto el resultado.

Subr nFiltrados 0 de 9

Subr Sqrt1d 0 de 9

Subr Comp 0 de 10

Subr SubMatriz 0 de 10

Subr ValorPixel 0 de 10

Subr FilPixel 0 de 10

Subr Filtro 0 de 10

Todas las realizadas hasta el momento, pasaron absolutamente todas las pruebas menos filtRec que no estaba del todo realizada.

Día 19, 20, 21 y 22

Se empezó con filtrec y tuve muchas dificultades para realizar y entender esta subrutina, en un principio encontré una solución rápida para esta subrutina y fue apoyarme en variables globales, pero me percaté que esto estaba penalizado, así que se me complicó aún más la realización de esta.

Mi problema realmente era propagar las variables de salida Ncambio y MaxFiltrado.

Día 23

Conseguí entender la recursividad y como empezar propagar las variables citadas anteriormente.

Día 24

Realicé otra comprobación enviando el proyecto al gestor de errores. El día 18/12/20 comprobé las pruebas que pasaba hasta el momento y fueron las siguientes:

Subr nFiltrados 0 de 9

Subr Sqrt1d 0 de 9

Subr Comp 0 de 10

Subr SubMatriz 0 de 10

Subr ValorPixel 0 de 10

Subr FilPixel 0 de 10

Subr Filtro 0 de 10

Subr FiltRec 1 de 16

La prueba que falla de fitRec es la siguiente:

Su implementación falla en una prueba en que se hace lo siguiente:

- Llama a 'FiltRec' sobre una imagen de 4x4 elementos, con un

filtro que devuelve para cada píxel la media de los que lo

rodean.

El parámetro NCambios tiene valor 400 y MaxFiltrados 4.

El parámetro NCambios tiene valor 400 y MaxFiltrados 4.

Como la entrega del proyecto es el día 21/12/20, no me arriesgue a realizar algún cambio que perjudicará a mi proyecto, por ello decidí no arreglar este error, además de que no sabía dónde se producía.