TRABAJO PRÁCTICO
S.E.T.I. 2 (Temario antiguo)

Queremos desarrollar un juego de MASTERMIND para el ordenador de mano *Palm Pilot*. El juego consiste en adivinar una clave de N dígitos alfanuméricos previamente almacenada. En cada iteración del juego se introduce una combinación de prueba y el ordenador responde con el número de coincidencias en valor y posición (muertos) y el número de coincidencias en valor, pero no en posición (heridos). El juego termina cuando se supera el número máximo de intentos permitidos M, o cuando se introduce la combinación correcta (nº muertos = N).

```
Ejemplo:
      CLAVE       (5710)
      Intento 1   (6415)      H=1, M=1
      Intento 2   (7105)      H=4, M=0
      Intento 3   (2710)      H=0, M=3
      Intento 4   (5710)      H=0, M=4 Fin del juego
```

Se nos pide:
BÁSICO. Realizar el juego básico utilizando como entrada y salida el puerto serie. Inicialmente se solicita el número de dígitos de la clave a adivinar. Se introduce la clave pero haciendo eco con '*'. Y a continuación se comienza el juego introduciendo los intentos y mostrando el resultado.

```
Ejemplo:
>>    Introduce el número de digitos
<<    4
>>    Introduce la clave
<<    ****
>>    Intento nº 1
<<    1234
>>    1 Muerto y 3 Heridos
...
>>    Correcto, la clave era 1432
>>    ¿Quieres jugar otra vez? (S/N)
<<    N
```

PARA NOTA. Realizar el juego en la misma forma solo que la salida se realice a través de la pantalla gráfica. Para ello tendrás que tener almacenados en una tabla la representación gráfica de cada carácter imprimible.

TRABAJO PRÁCTICO
S.E.T.I. 2 (Temario antiguo)

HARDWARE

La práctica se desarrollará para que funcione en un ordenador de mano *Palm Pilot,* basado en el microcontrolador MC68328 de DragonBall. Este microcontrolador tiene como núcleo básico un MC68000. Más información sobre este microcontrolador en :

ftp://bigseti/pub/DOCUMENTACION/Microcontroladores/Motorola/68000-based/mc68328um.pdf

El ordenador de mano citado, dispone de una conexión serie y de una pantalla gráfica táctil (160x160 puntos en blanco y negro), así como una memoria ROM de 1Mb (comienza en $10c00000) y una memoria ROM de 256 kb (comienza en $10000000, aunque está mapeada parcialmente a partir de la dirección $0).

SOFTWARE

Se proporciona para la realización de la práctica un emulador de *Palm Pilot* que funciona bajo Linux, así como un ensamblador de MC68000:

ftp://bigseti/pub/DOCUMENTACION/Microcontroladores/Motorola/68000-based/emulador_unix.tgz

Para poder trabajar con él se proporcionará una cuenta a cada alumno en la máquina Linux *bellota.ele.uva.es*

Además se proporciona un ensamblador, enlazador y simulador de MC68000 para trabajar en DOS.

ftp://bigseti/pub/DOCUMENTACION/Microcontroladores/Motorola/68000-based/utiles68000_dos.zip

Para poder comenzar a trabajar se proporciona un programa ejemplo comentado.

Para compilar el programa ejemplo, que se encuentra en el paquete **emulador_unix.tgz**, primero desempaquetamos el fichero en el directorio apropiado:

```
>tar xzvf emulador_unix.tgz
```

y a continuación ejecutamos el comando **make:**

```
>make
```

```
* Ejemplo de programa para el Palm Pilot
* El programa se coloca en la memoria ROM del ordenador

       org    $10c00000

* La tabla de vectores de interrupción inicialmente se
*  encuentra en ROM, pero es preciso volcarla en RAM
*  inmediatamente. Tan solo están definidos los dos
*  primeros vectores que almacenan el puntero de pila
*  inicial (que apunta al final de la memoria RAM), y
*  el contador de programa inicial (que señala el comienzo
*  de nuestro programa).
***************** Vector TABLE ************************
vtab:
       dc.l   $10040000     *SP inicial
       dc.l   start         *PC inicial
vtab_end:

***************** CODE SECTION ************************

* Aquí comienza el código ejecutable y lo primero que hago
*  es copiar la tabla de vectores de interrupción a la
*  posición 0 de memoria que forma parte de la RAM.
*---------------------- Entry Point ----------------------
start:
       lea.l  vtab,a0               * Dirección fuente
       lea.l  0,a1                  * Dirección destino
       move.w #1,d0                 * Número de vectores(2)
bvtab  move.l (a0)+,(a1)+
       dbf    d0,bvtab              * Copio hasta acabar

* Aquí comienza la inicialización del microcontrolador
*  Si quieres saber más consulta el datasheet del mismo
*---------------- Dragonball init -----------------------

        moveq   #0,d0
        move.w  d0,$fffff618         * Watchdog off
        move.l  #$00011f07,$fffff114 * CS A1 Mask

        move.w  #$2410,$fffff200     * PLLCR
        move.w  #$123,$fffff202      * PLLFSR

        move.b #0,$fffffA27          * LCKCON
        move.l #face,$fffffA00       * LSSA  Inicio de la pantalla
        move.b #$a,$fffffA05         * LVPW
        move.w #$9f,$FFFFFa08        * LXMAX
        move.w #$9f,$FFFFFa0a        * LYMAX
        move.b #9,$fffffa29          * LBAR
        move.b #0,$fffffa25          * LPXCD
        move.b #$04,$FFFFFa20        * LPICF
        move.b #$58,$fffffA27        * LCKCON
        move.b #$85,$fffff429        * PFDATA
        move.b #$d8,$fffffA27        * LCKCON
        move.b #$c5,$fffff429        * PFDATA
        move.b #$d5,$fffff429        * PFDATA

        move.w  #$e108,$fffff900     * UART Control
        move.w  #0,$fffff908         * UART misc
        move.w  #$0800,$fffff906     * Ignore CTS
        move.w  #$010b,$fffff902     * Baud=9600

* Aquí comienza tu programa. Has de saber que en este momento la memoria
*  que se está visualizando en pantalla (1 bit = 1 pixel) en blanco y
*  negro se encuentra en una zona de memoria ROM, por lo que deberías
*  modificarla para poder escribir en ella.
*-------------- MAIN -----------------------------


main:

       moveq.l #32,d1        * Este programa no tiene ninguna utilidad
bmain1: move.b  d1,d0        *  es tan solo un ejemplo
       bsr     putc          * Lee caracteres del puerto serie
       bsr     getc          *  y reenvía un carácter contador por el mismo puerto
       bsr     putc          *  añadiendo al final los caracteres
```

```
       move.b  #13,d0          *  CR=13 y LF=10 (retorno de carro y avance de
       bsr     putc            *  línea). Comienza enviando el carácter 32.
       move.b  #10,d0          * Y repite indefinidamente
       bsr     putc
       add.b   #1,d1
       bne     bmain1
       bra     main

* Te proporcionamos dos subrutinas básicas para el acceso al puerto serie
* Con  putc  puedes enviar desde el microcontrolador un carácter.
* Con  getc  puedes recibir un carácter desde el puerto serie.
*-------------------- ROUTINES ---------------------

putc:  move.l  d1,-(sp)
       move.w  $fffff906,d1
       btst.l  #13,d1
       beq     putc
       move.b  d0,$fffff907
       move.l  (sp)+,d1
       rts

getc:  move.l  $fffff310,d0
       btst.l  #2,d0
       beq     getc
       move.w  $fffff904,d0
       rts

* Para que puedas comprobar como se visualizan los datos en la
*  pantalla gráfica del Palm Pilot, te mostramos un ejemplo.

*------------------- Too much face -------------------

face:
       dc.b    $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$03,$FA,$80,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$05,$56,$80,$00,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$2B,$FA
       dc.b    $A0,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$5E,$DF,$D0,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00
       dc.b    $04,$95,$F7,$F6,$B4,$00,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$00,$03,$DB,$7F,$BD,$BF
       dc.b    $DF,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$00,$4E,$B6,$D6,$EF,$DA,$ED,$D0,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$3B
       dc.b    $FF,$FF,$FB,$7F,$FF,$7C,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$02,$FE,$D6,$ED,$5F,$EF
       dc.b    $5B,$ED,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$01,$D7,$FF,$FF,$F6,$FB,$FE,$BF,$80,$00
       dc.b    $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$07,$7F
       dc.b    $7B,$B6,$FF,$BF,$D7,$F5,$40,$00,$00,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$05,$DF,$FE,$FF,$DF,$FE
       dc.b    $FD,$BF,$E0,$00,$00,$00,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$0F,$FD,$DF,$BD,$F6,$F7,$BF,$EE,$D0,$00
       dc.b    $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$15,$77
       dc.b    $FF,$EF,$7F,$DD,$FB,$FF,$F8,$00,$00,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$2F,$FF,$5B,$FF,$F6,$FF
       dc.b    $EE,$BB,$74,$00,$00,$00,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$0B,$AD,$FE,$FB,$5F,$B6,$BF,$FF,$FC,$00
       dc.b    $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$1E,$FB
       dc.b    $57,$AF,$F5,$FF,$F7,$EF,$EE,$00,$00,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$37,$B7,$55,$7D,$5F,$6D
       dc.b    $BF,$7B,$7B,$00,$00,$00,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$1D,$EC,$AA,$13,$6A,$DB,$ED,$FF,$FF,$00
       dc.b    $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$37,$55
       dc.b    $00,$44,$AA,$AB,$7F,$B7,$BD,$80,$00,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$2D,$E8,$40,$01,$09,$01
       dc.b    $5A,$FD,$EF,$40,$00,$00,$00,$00,$00,$00,$00,$00
       dc.b    $00,$00,$1E,$92,$00,$00,$20,$48,$AF,$EF,$FF,$A0
       dc.b    $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$2B,$40
       dc.b    $08,$20,$00,$00,$15,$7E,$FF,$C0,$00,$00,$00,$00
       dc.b    $00,$00,$00,$00,$00,$00,$5E,$A0,$00,$00,$00,$00
       dc.b    $4B,$DB,$ED,$D0,$00,$00,$00,$00,$00,$00,$00,$00
*..,  etc.
```

Resultado gráfico del ejemplo.