eman ta zabal zazu
universidad del país vasco
euskal herriko unibertsitatea

departamento de electricidad y electrónica
elektrika eta elektronika saila

Examen de S.E.T.I.

1er curso de Ingeniería Electrónica
8 de febrero de 2010

Ejercicio de C.	**PARTE 1**	1 punto

1 Programar en C la rutina "StupidSort" que sirve para ordenar un array de enteros mediante el siguiente mecanismo extremadamente sencillo (e ineficaz):

- Localizar el primer par de números adyacentes desordenados.
- Si no se encuentra, el vector ya está ordenado. FINALIZAR.
- De lo contrario, intercambiarlos y VOLVER A EMPEZAR.

Escribir también un programa principal que sirva para comprobar su funcionamiento con un array inicializado en el mismo código.

(Adóptese la solución que se quiera en todo aquello que el enunciado no obligue a nada).

Cuestiones.	**PARTE 2**	1,25 + 1,25 puntos

2 El problema de la codificación de textos (caracteres) en la actualidad es más importante de lo que puede imaginarse alguien que no se haya tenido que enfrentar a él. Existiendo una solución como es UNICODE, muchos sistemas no le dan soporte. Póngase en el lugar de un desarrollador de software que quiere convencer a su cliente para que el encargo de un determinado programa incluya dicho soporte. Argumente sobre las capacidades de UNICODE y los problemas de no soportarlo.

3 Si utilizo como fuentes dos modelos de Markov de orden N y N+1 "aprendidos" de un mismo lenguaje (P.ej. del inglés) , ¿Qué relación habrá entre sus entropías?. Razónelo.

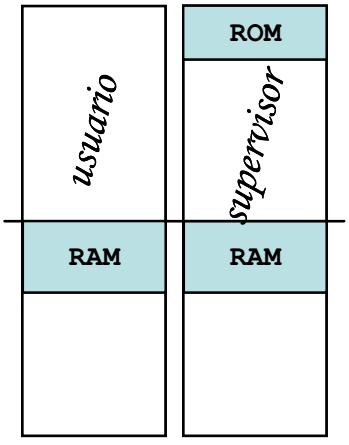
Estructura de un microprocesador.	1,25 punto

4 Nuestro computador virtual es extremadamente simple. Comparándolo con el 68000 carece de muchas capacidades y características. Haga un listado de estas diferencias. Se valorará la completitud, la concisión, la claridad…. (lo que no impide que algunas apreciaciones puedan ser puntos concretos de otras).

p.ej.: - Dispone de muchas menos instrucciones (16 frente a 56. Carece de producto, división, rotaciones, etc)
- No incluye nada semejante a la instrucción TAS que permita la ejecución multitarea.

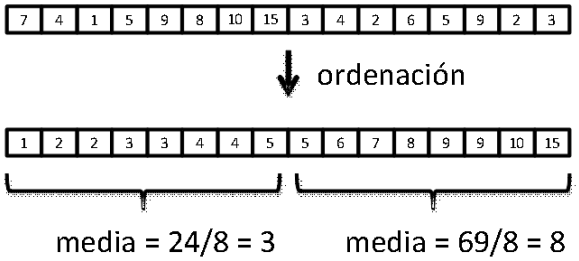
Mapeo de memoria.	1,25 puntos

5 Se pretende diseñar un circuito de computador con µP68000 utilizando como memorias 2 ROM estructuradas en bytes de 1 MByte cada una y 4 RAM de 1MByte cada una estructuradas en niblas (4 bits). Se le pide que diseñe los circuitos necesarios para que la ROM quede emplazada en las posiciones iniciales del espacio direccionable y sólo sea accesible desde el estado supervisor, y que la RAM se situe alineada con el comienzo de la segunda mitad del espacio direccionable y sea accesible en todo caso. Indique también qué rangos de direcciones quedan ocupadas.



Programación del 68000.	**PARTE 3**	4 puntos (1 + 1 + 1 + 1)

6 Supongamos que por alguna extraña razón nos interesa tomar un vector de enteros y calcular dos medias aritméticas: por un lado la media de la mitad del vector compuesta por los números más pequeños, y por otro lado la media de la otra mitad del vector (compuesta por los números más grandes). Una forma de atacar este problema pudiera ser ordenar primero el vector, para posteriormente calcular la media de cada mitad del vector ordenado. La figura muestra los pasos.



Téngase en cuenta, que la media es obtenida a partir de la división entera, y que de forma general, dado un vector de tamaño n, se entiende que la mitad inferior es de longitud n/2 (división entera), y la superior es de longitud n-n/2 (no n/2) (es decir, si hay un número impar de enteros, la segunda "mitad" tiene un número más).

Se pide que se escriban las siguientes subrutinas y programa principal (las subrutinas han de cumplir las descripciones de entrada/salida indicadas. De no ser así, justifíquese el cambio):

```
; Subrutina:  STUPIDSORT
; Ordena un vector de enteros sin signo de 16 bits. (ver enunciado ejercicio 1)
; Entrada:  A0.L  Dirección del vector
;           D0.W  Longitud del vector
; Modifica: …
```

```
; Subrutina:  MEDIA
; Calcula la media aritmética entera de un vector de enteros sin signo de 16 bits.
; Entrada:  A0.L  Dirección del vector
;           D0.W  Longitud del vector;
; Salida:   D1.W  Media
; Modifica: …
```

```
; Subrutina:  RAREZA
; Calcula la media inferior y superior de un vector de enteros sin signo de 16 bits.
; Entrada:  A0.L  Dirección del vector
;           D0.W  Longitud del vector;
; Salida:   D1.L  Word inferior=media inferior, Word superior=media superior
; Modifica: …
```

Programa principal

El programa principal deberá crear un vector de enteros, obtener su media inferior y superior y mostrar el resultado por pantalla.

eman ta zabal zazu
universidad euskal herriko
del país vasco unibertsitatea

departamento de electricidad y electrónica
elektrika eta elektronika saila

Examen de S.E.T.I.

1er curso de Ingeniería Electrónica
8 de febrero de 2010

SOLUCIONES Y COMENTARIOS

1 pendiente

2 pendiente

3 pendiente

4 el computador virtual…

0a- Dispone de muchas menos **instrucciones** (16 frente a 56. Carece de producto, división, rotaciones, etc)
0b- No dispone de nada semejante a la instrucción TAS que permita la ejecución **multitarea**.
1- Dispone de muchos menos **modos de direccionamiento**.
2- No dispone de mecanismo de **excepciones** (necesario p.ej en el caso de instrucción ilegal)
3- No dispone en particular de mecanismo de **interrupciones.**
4- No dispone de capacidad de ejecutar **subrutinas** (sin instrucciones ad-hoc ni manejo de pila)
5- Dispone de menos **flags** (no hay extensiones, control de rebose –add/sub sólo sin signo-, menos condiciones de salto)
6- No soporta dos modelos de programación **usuario/supervisor**.
7- Tiene arquitectura de acumulador (frente a la de 2 operandos y modelo de registros más complejo)

5 pendiente

6

```
        org     $400

        lea     lista,a0                ; Apuntamos a la lista
        move.w  #(finlista-lista)/2,d0  ; El número de elementos es el_espacio_ocupado_en_bytes/2
        bsr     rareza                  ; cálculo de la rareza
        trap    #15                     ; Terminación de la ejecución
        dc.w    0


lista:  dc.w    7,4,1,5,9,8,10,15,3,4,2,6,5,9,2,3
finlista:

;=============================================================================================
; Subrutina: STUPIDSORT
; Ordena un vector de enteros sin signo de 16 bits.
; Entrada: A0.L Dirección del vector
; D0.W Longitud del vector
; Modifica:


Stupidsort:
        movem.l a0-a1/d0,-(sp)
        lea     2(a0),a1        ;Compararemos los pares con A0 y A1
        subq.w  #2,d0           ;Una comparación menos que elementos y una menos por contar el cero
busca:  cmpm.w  (a0)+,(a1)+     ;       Se desplaza por todos los pares
        dblo    d0,busca        ; mientras estén OK y no llegue al final
        tst.w   d0              ;¿Se ha salido porque todo está OK?
        bmi.s   finSS           ;Si-- terminar
        move.w  (a0),d0         ;No-- intercambiar y repetir todo
        move.w  -2(a0),(a0)     ;       como han avanzado se usa A0 que iba por detrás
        move.w  d0,-2(a0)       ;       y el elemento anterior -2(A0)
        movem.l (sp)+,a0-a1/d0  ;       para repetir todo se recupera la pila
        bra.s   Stupidsort      ;
finSS:  movem.l (sp)+,a0-a1/d0  ;
        rts


;=============================================================================================
; Subrutina: MEDIA
; Calcula la media aritmética entera de un vector de enteros sin signo de 16 bits.
; Entrada: A0.L Dirección del vector
; D0.W Longitud del vector;
```

```
; Salida: D1.W Media
; Modifica: La parte alta de D1 sale con el resto


Media:   tst.w    d0              ;Si la longitud es cero
         beq.s    finM            ; nada que hacer
         movem.l  a0/d2,-(sp)     ;
         move.w   d0,-(sp)        ;El contador se salva para recuperarlo y calcular la media
         clr.l    d1              ;Sumaremos en largo para estar menos limitados
         clr.l    d2              ;limpiamos d2 para recoger de memoria en largo
         bra.s    fcicloM         ;Entramos en el ciclo por el final por contar el cero
cicloM:  move.w   (a0)+,d2        ; recogemos cada número en word
         add.l    d2,d1           ; y lo sumamos en largo
fcicloM: dbra     d0,cicloM
         move.w   (sp)+,d0        ;recuperamos el contador para...
         divu     d0,d1           ;dividir la suma y obtener la media
         movem.l  (sp)+,a0/d2
finM:    rts

;=================================================================================================
; Subrutina: RAREZA
; Calcula la media inferior y superior de un vector de enteros sin signo de 16 bits.
; Entrada: A0.L Dirección del vector
; D0.W Longitud del vector;
; Salida: D1.L Word inferior=media inferior, Word superior=media superior
; Modifica:

Rareza:  move.w   d2,-(sp)
         bsr      Stupidsort      ;se aplica el sort al vector

         move.w   d0,d2           ;conservamos la longitud en D2
         lsr.w    #1,d0           ;y ponemos la mitad
         bsr      Media           ;para obtener la primera media

         sub.w    d0,d2           ;a la long. original le quitamos la de la primera mitad para tener la segunda
         lsl.w    #1,d0           ;el inicio de la segunda mitad está a 2xd0 del inicio
         lea      0(a0,d0.w),a0   ;de manera que adelantamos el apuntador
         move.w   d2,d0           ;y colocamos la longitud de la segunda mitad
         move.w   d1,d2           ;en d2 guardamos el resultado de la primera media
         bsr      Media           ;calculamos la segunda
         swap     d2              ;el resultado de la primera media va a la parte alta
         and.l    #$ffff,d1       ;del de la segunda media nos quedamos solo con la parte baja
         or.l     d2,d1           ; y los fundimos
         move.w   (sp)+,d2
         rts
```