

Amb l'augment de la sensació d'inseguretat ciutadana (les estadístiques oficials encarregades pels polítics diuen el contrari), les empreses que es dediquen a proveir serveis relacionats amb la seguretat no donen l'abast. Ells disposen d'uns sistemes d'alarma que són capaços d'enviar remotament una sèrie de missatges en cas de produir-se incidències (intrusions, incendis, inundacions...). Aquests missatges provenen de diferents llocs i consisteixen en trames sèrie que poden arribar a la central d'alarmes (el lloc on hi ha permanentment personal de seguretat) per diferents canals (internet, telèfon, ràdio..). Tots aquests missatges es multiplexen (s'ajunten) en una sola línia del tipus RS-232 connectada a un ordinador tipus PC. Cadascun d'aquest missatge està compost per un número d'identificació i un text explicatiu de la incidència.

Per tal d'optimitzar recursos ens han encarregat una feina molt concreta: Dissenyar un programa per aquest PC per tal que atengui la línia sèrie i visualitzi de mode adient les incidències rebudes. L'operador d'aquest servei actuarà en conseqüència (trucant a la policia, als bombers o a qui calgui).

Enunciat del problema:

- Cal escriure en ensamblador TOT el codi necessari (segments, variables, subrutines...) per rebre caràcters d'una línia sèrie, processar els missatges i visualitzar-los a pantalla.
- La línia sèrie serà COM1 i s'hi accedirà per enquesta. Suposarem que ja està inicialitzada (o bé crideu a una funció `ini_serie` dins el vostre codi) en quant a velocitat, nombre de bits, paritat... La UART del port sèrie indica que "ha arribat un caràcter" i es pot llegir amb la combinació "10" als bits 2 i 1 del Interrupt Identification Register (de 8 bits) que és a la adreça 3FAh. El búffer de recepció (d'allà on llegim els caràcters) és a l'adreça 3F8h i té 8 bits. Pel fet de llegir el búffer el flag de "caràcter pendent" s'esborra sol (no cal fer cap tipus de "STROBE").
- Els missatges són seqüències de caràcters ASCII on el primer és un número entre 1 i 10 i la resta text (el codi ASCII de "qualsevol lletra" és més gran que 64). Un exemple de missatges rebuts en un interval de temps pot ser: "...3 Incendi a la planta baixa.6 Detectada aigua als sotans8 Intrusió a la cuina, finestra oberta..." (Cal notar que els números subratllats i en cursiva no són el corresponent a "la lletra número x" sinó al valor).
- Aquests missatges s'han d'escriure a la pantalla a la fila corresponent al "(número*2)+1". Així, un missatge que comença per "3" s'escriurà a la fila 7 mentre que un que comença per "8" s'escriurà a la fila 17. Cal esborrar tota la fila abans d'escriure un nou missatge per tal d'evitar que un missatge curt es "barregi" amb un d'anterior llarg.
- Un missatge amb capçalera número 0 farà que el programa acabi.
- Per evitar "conflictes" amb el sistema operatiu, cal que deshabiliteu la IRQ4, corresponent a COM1.
- Totes les funcions a les que se'ls hi passi més d'un paràmetre aquests s'han de passar per la pila.
- Heu d'escriure, com a mínim, dues subrutines; una per escriure un caràcter a pantalla i una per esborrar una fila de la pantalla.

Suggerències:

- Podeu fer servir el següent esquema:

```
PROGRAMA Alarma
  Inicialitzacions    //Variables, ini_serie, interrupcions...
  REPETIR
    MENTRE NOT NouCaracter    // Bucle que examina si ha arribat
    FMENTRE                // algun caracter.
    LlegirCaracter
    SI (caracter<=10 AND carácter!=0) LLAVORS    //    Examinem    si    és
    número...
        fila:=caracter*2+1
        columna:=1
        esborra(fila)
    ALTRAMENT                // ..o bé lletra
    EscriuCaracter(fila,columna,caracter)
    columna:=columna+1
    FSI-ALTR
  FINS (caracter=0)        // Fi del bucle principal
  RestaurarEestatSistema    // Interrupcions..
FPROGRAMA
```

- Podeu escriure directament a memòria de pantalla o bé fer crides al sistema operatiu.
- Per simplificar els tractaments, podeu tractar la parella fila-columna com a una única variable anomenada posició. Les posicions de pantalla les podeu tindre pre-calculades en una taula.
- Suposarem que els missatges no són més llargs de 80 caracters i no cal comprovar "salts de línea".

Solució:

Per resoldre el problema prenc com a referència l'esquema proposat i la resolució de la pràctica del teclat com a eina bàsica de consulta (Deixo en cursiva l'esquema).

```
;PROGRAMA Alarma
.model large
.386
.stack 100h
Inicialitzacions  //Variables, ini_serie, interrupcions...
.data
taula_files  ; (2*n+1) * 80 *2 = fila * 80 car * caracter+atribut ("salten" de 320 en 320)
              DW  0,480,800,1120,1440  ;la posició "0" de la taula no conté res
              DW  1760,2080,2400,2720
              DW  3040,3360

.code
.startup
inhibir_interr:    CLI
                  IN   AL,21h
                  OR   AL,10h          ;IRQ4 = 0001 0000 = 10h
                  OUT  21h,AL
                  STI

                  CALL adpant          ;no l'implementaré per simplicitat, retorna per AX addr
base
                  MOV  ES,AX           ;deixo a ES l'adreça base

                  CALL ini_serie       ;la suposo implementada

REPETIR

enquesta: MENTRE NOT NouCaracter  // Bucle que examina si ha arribat
          IN   AL,03Fah  ;examino si ha arribat un acharacter
          OR   AL,06h    ;bits 2 i 1 0000 0110 = 06h
          XOR  AL,04h    ;examino si són "10" 0000 0"10"0 = 04h
          FMENTRE  // algun caracter.
          JZ   enquesta  ;a XOR b = 0 si i només si a=b
          LlegirCaracter
          IN   AL,03F8h  ;llegeixo del búffer
          JZ   sortir    ;si és un zero s'acabat
          SI (caracter<=10 AND carácter!=0) LLAVORS // Examinem si és
          número...
          ; arribats a aquí, ja sé q no serà zero => nova condició "SI <11"
          MOV  DI,AL      ;salvo el valor per adreçar amb DS:DI
          SUBB AL,11      ;resto 11
          JC   es_lletra   ;si és negatiu... és més gran que deu
          fila:=caracter*2+1
          MOV  AX,taula_files[DI] ;deixo a AX (paràmetre) posició de la fila a
esborrar
          columna:=1
          MOV  BX,AX      ;faré servir BX com a índex de columna
          esborra(fila)
          CALL esborra_fila
          ALTRAMENT  // ..o bé lletra
```

es_lletra:

EscriuCaracter(fila,columna,caracter)

PUSH BX ;aquí tinc el lloc on s'ha d'escriure

PUSX DI ;aquí tinc el caracter

CALL escriu_caracter

ADD SP,4 ;treure paràmetres

columna:=columna+1

INC BX

INC BX

FSI- ALTR

FINS (caracter=0) // Fi del bucle principal

sortir:

RestaurarEstatSistema // Interrupcions..

CLI

AND AL,0EFh ;IRQ4 = 1110 1111 = EFh

OUT 21h,AL

STI

FPROGRAMA

.end

;Subrutines

esborra_fila: PUSH DI ;salvem els registres q farem servir

ADD AX,158 ;2*79=158, AX ja conté l'adreça base

MOV DI,AX ;faré servir DI com a contador

bucle: MOV ES:[DI],WORD PTR 2007h ;escric un espai

DEC DI

DEC DI ;un caràcter ocupa 2 llocs (car+atribut)

JNZ bucle

POP DI ;restablim DI

RET

escriu_caracter: PUSH BP ;establim el bloc d'activació

MOV BP,SP

;no hi ha variables locals

PUSH BX ;salvem els registres q fem servir

MOV AX,-4[BP] ;recupero el caracter

MOV BX,-6[BP] ;recupero la posició

MOV ES:[BX],AX ;escrivim a pantalla

POP BX ;restaurem el registre

POP BP ;desfem el pivot

RET

.end