

## ¿ Como personalizar la gestión del desarrollador en un Gran Sistema ? (y II)

o

### Personalización de las gestiones habituales del desarrollador mediante el uso de menús (y II)

o

#### Creación de un Menú para el desarrollador en un gran sistema (y II)

El pasado mes se inicio este tema hablando de la creación de un menu o panel de selección en el entorno de IBM. Este mes se continua el tema, pero para hablar de otro entorno diferente, el entorno DIGITAL. En cualquier caso, solo se presentan distintas formas o funciones básicas que pueden ser utilizadas por el programador para personalizar su propia gestión de desarrollo.

#### DIGITAL : Creación de un menú básico

En el entorno DIGITAL, se trabaja de una forma muy parecida a como se trabaja en un PC. De entrada, cuando se conecta un usuario, siempre el sistema le asigna un directorio por defecto, que suele tener el nombre lógico de **sys\$login**, y además, ejecuta los procedimientos de conexión que cargan todas aquellas variables necesarias para el entorno. Pero, lo mas importante, por lo que respecta al tema que nos ocupa, es que termina ejecutando el fichero de comandos **LOGIN.COM**, en el caso de que exista, del directorio asignado por defecto. Este fichero, que equivale al tan conocido **AUTOEXEC.BAT** del MS-DOS, es el que normalmente usaremos para realizar aquellas asignaciones necesarias para poder personalizar nuestra gestión de desarrollo.

Como ejemplo se muestra en la *figura 1* el listado del archivo **LOGIN.COM**.

```
$ !-----  
$ ! PAL810::DUA0:[000000.DESJMP]LOGIN.COM  
$ !-----  
$ IF F$MODE() .NES. "INTERACTIVE" THEN EXIT  
$ def/key/terminate/noecho e2 -  
    "@sys$login:menu.com"  
$ cd == "DUA0:[000000.DESLFU]cd.COM"  
$ @menu
```

En Digital, los procedimientos de comandos pueden escribirse tanto en minúsculas como en mayúsculas, pues el procesador internamente las trata todas como mayúsculas. No ocurre lo mismo con los valores asignados a las variables, pues diferencia unas de otras, debiendo considerar esta circunstancia para examinar la opción elegida por el usuario en un menú.

Por lo que respecta al fichero citado cabe hacer los siguientes comentarios:

Por ser un archivo de comandos, cada uno de los debe comenzar por el símbolo \$

Los comentarios se identifican por el símbolo '!' por lo que las tres primeras líneas lo son.

La 4ª línea especifica que cuando se invoque este procedimiento en un modo distinto al interactivo, es decir en batch, se salga de él.

La siguiente línea programa la tecla **E2** (Insertar) con el valor que se indica, y que se corresponde con la ejecución del fichero de comandos MENU. El hecho de especificar el direccionamiento completo de dicho archivo es para poder ejecutarle con independencia del directorio en el que se encuentre el desarrollador. Es de resaltar que

como no cabe todo el comando en una línea, se utilizan dos, razón por la cual, la primera termina con un guión '-' y la siguiente empieza **sin** el '\$'.

La siguiente línea asigna al símbolo CD el direccionamiento completo de otro archivo de comandos que permitirá emular el comando CD del entorno MS-DOS.

Por ultimo, se ejecuta el archivo de comandos **MENU.COM** (pues va precedido del símbolo @ y al no especificar extensión, asume que es .COM).

Este ultimo archivo, **MENU.COM** es el objetivo de este artículo, por lo que se expone a continuación un menú muy reducido que permita iniciarse en esta función, si bien en próximos artículos se ira ampliando con nuevas opciones el menú que hoy se comenta. Cabe resaltar, eso sí, antes de empezar, que los números que figuran a la izquierda solo son una referencia para poder comentar cada una de las líneas, por lo que aquellos que quieran ejecutar este ejemplo, deberán hacerlo sin escribir dicho numero.

El listado completo se muestra en la *figura 2*, y los comentarios a cada una de las líneas es el siguiente:

**007** asigna a la variable `esc[0,8]` el valor 27 ascii, es decir el carácter ESCAPE

**008** carga la variable **WS** para utilizarla como abreviatura de `write SYS$output` (es decir escribe sobre donde este definida la salida lógica de mensajes)

**009** asigna a **KEY\_NUM** los caracteres que activa el teclado numérico.

**010** asigna a la variable **CLS** los caracteres para borrar la pantalla

**011** establece el prompt del usuario como **JMP>** (Lo lógico seria incluir esta sentencia en el LOGIN.COM)

**012** hace que el usuario se posicione en el directorio que tiene definido por defecto. De esta forma regresa a él cuando al pulsar la tecla **E2** programada en **LOGIN.COM** se ejecute este procedimiento.

**023** y **024** activa teclado numérico y borra pantalla

**025** asigna a la variable **CABECERA** el valor especificado, donde **0;35H** indica que escriba en la fila 0 columna 35. **#6** indica que escriba en modo de 40 columnas por línea; y **0m** indica que escriba en intensificado lo que sigue hasta encontrar las dobles comillas (JMP).

**027** escribe en pantalla la variable **CABECERA**.

**029** escribe en fila 4 columna 10 y en vídeo inverso (**[7m]**) el texto "directorio"

**031** escribe en 7,10 un 1 en intensificado (**[1m]**) y la palabra usuario como texto normal (**[0m]**)

Así se escriben el resto de opciones, combinando la posición, con el tipo de atributo (intensificado, normal y vídeo inverso)

**053** escribe en la posición 22,5 el contenido de la variable **MENSAJE** en parpadeante (**[5m]**)

**056** lee desde donde este definida la entrada de comandos (**sys\$command**) es decir desde teclado, la opción que se elija, cargando la variable **OPC**.

**057** transforma a mayúsculas el contenido de **OPC**

**059** examina la longitud de **OPC** y si es mayor que 1, carga la variable **mensaje**, cediendo el control al párrafo etiquetado con menu (línea 21) para revisualizar el panel de opciones y listar así el contenido del mensaje en parpadeante.

**066** examina si la opción elegida se encuentra entre los caracteres que contiene la variable **opciones** cargada en la línea anterior.

**071** si la opción elegida es la **1** establece que el directorio por defecto sea **SYS\$LOGIN**.

**075** si la opción elegida es la **2** establece que el directorio por defecto sea el especificado.

**086** si la opción elegida es la **8** establece que el terminal presente la información en **80** columnas.

**087** ídem, pero con **132** columnas.

**089** si la opción elegida es la **B** define que la entrada del sistema sea la entrada de comandos, ya que de no hacer esta asignación, el sistema asume que los datos de entrada del programa que se va a ejecutar a continuación, recibe los datos desde el propio fichero de comandos.

**090** invoca al programa **BASIC**

**093** pide el nombre del fichero que se desea editar. Esta es otra forma de pedir datos al usuario distinta a la vista en la línea 56, pero con la misma funcionalidad.

**099** invoca al programa de utilidad **PHONE** normalmente para ejecutar luego **DIAL id\_usuario** y poder llamar a otro usuario, o **help**, etc.

**103** invoca al programa de utilidad **PHONE** pero con el parámetro **ANSWER** para poder responder a la llamada que se esta recibiendo.

**106** limpia pantalla pues escribe (WS) el contenido de la variable **CLS**

**107** sale del procedimiento

```
001 $ IF F$MODE() .NES. "INTERACTIVE" THEN EXIT
002 $ ! =====
003 $ !                               Menu de UTILIDADES de JMP
004 $ ! =====
005 $ !
006 $ ! ----- asignaciones
007 $   esc[0,8] = 27
008 $   ws       = "write sys$output"
009 $   key_num  = "'esc'>"
010 $   cls     = "'esc'[2J"
011 $   set prompt="Jmp> "
012 $   set def sys$login
020 $ ! -----
021 $ menu:
022 $ !
023 $   ws key_num
```

```

024 $ ws cls
025 $ CABECERA = "'esc'[0;35H'esc'[J'esc'#6'esc'[0m JMP "
027 $ ws "'CABECERA'"
028 $ ! ===== definicion de la pantalla =====
029 $ ws "'esc'[04;10H'esc'[1;7m ** Directorio **'esc'[0m"
031 $ ws "'esc'[06;10H'esc'[1m1.'esc'[0m Usuario      'esc'[0m"
032 $ ws "'esc'[07;10H'esc'[1m2.'esc'[0m Proyecto      'esc'[0m"
039 $ ws "'esc'[19;05H'esc'[1m8.'esc'[0m 80 columnas    'esc'[0m"
040 $ ws "'esc'[20;05H'esc'[1m9.'esc'[0m 132 columnas   'esc'[0m"
041 $ !
042 $ ws "'esc'[04;45H'esc'[1;7m Varios'esc'[0m"
045 $ ws "'esc'[07;45H'esc'[1mB.'esc'[0m Basic          'esc'[0m"
048 $ ws "'esc'[10;45H'esc'[1mE.'esc'[0m Editar         'esc'[0m"
041 $ !
053 $ ws "'esc'[22;05H'esc'[1;5m'mensaje''esc'[0m"
041 $ !
054 $ ! ===== fin de la definicion de la pantalla =====
056 $ read sys$command -
      opc/prompt="'esc'[22;60HTeclee opciñ: "
057 $ opc = f$edit(opc,"TRIM,UPCASE")
058 $ ! ----- examinar entrada
059 $ if f$length(opc) .GT. 1
060 $     then mensaje = "Identifique la opciñ con un solo d;gito"
062 $     goto menu
063 $ endif
064 $ ! ----- examinar opcion elegida
065 $ opciones = "1289BE"
066 $ encontrado = f$locate(opc,opciones)
067 $ if encontrado .eq. f$length(opciones)
068 $     then mensaje = "Seleccione una opciñ existente"
069 $     goto menu
070 $ endif
071 $ if opc .eqs. "1"
072 $     then set def sys$login
073 $     goto fin
074 $ endif
075 $ if opc .eqs. "2"
076 $     then set def desarrollo$disk:[proyecto]
077 $     goto fin
078 $ endif
079 $ if opc .eqs. "3"
080 $     then sh def exit
080 $     exit
081 $ endif
086 $ if opc .eqs. "8" then set terminal/width=80
087 $ if opc .eqs. "9" then set terminal/width=132
088 $ if opc .eqs. "B"
089 $     then define sys$input sys$command/nolog
090 $     basic
091 $ endif
092 $ if opc .eqs. "E"
093 $     then INQUIRE fichero "Archivo a editar ...:"
094 $     define sys$input sys$command/nolog
095 $     edit 'fichero
096 $ endif
097 $ if opc .eqs. "P"
098 $     then define sys$input sys$command/nolog
099 $     phone
100 $ endif
101 $ if opc .eqs. "Q"
102 $     then define sys$input sys$command/nolog
103 $     phone answer
104 $ endif
064 $ ! -----
105 $ fin:
106 $ ws cls
107 $ exit
064 $ ! -----

```