

SUPERCHARGE

Program and documentation © 1985 Simon N. Goodwin

Published by:
Freddy Vachha BSc
Digital Precision Ltd
Glossary Copyright Freddy Vachha 1985

SINCLAIR, Q.U. Super BASIC
Are Trademarks of Sinclair Research Ltd.



CONTENIDO

- 1 Una introducción rápida
- 2 Una introducción más pausada
- 3 Utilizando Supercharge
- 4 Multitarea
- 5 Extensiones al SuperBasic
- 6 Supercharge y el SuperBasic
- 7 Glosario

1 UNA INTRODUCCION RAPIDA

POR FAVOR LEA ESTO PRIMERO

Sabemos que está impaciente por comenzar a utilizar el Supercharge y que quizás no está dispuesto a leerse la totalidad de este manual.

Si tiene la paciencia de leerse las tres primeras hojas, será capaz de comenzar a usarlo casi sin problemas.

CODIGO MAQUINA (casi al instante)

- 1.- Haga un reset a su QL
- 2.- Introduzca el cartucho con el programa en el microdrive 1
- 3.- Arranque Supercharge con este comando :
LRUN mdvl_BOOT
- 4.- Teclée LOAD mdv2_PROGRAMA, en donde PROGRAMA será el programa que Vd. desea compilar.
- 5.- Teclée EXEC mdvl_parser_task
- 6.- Teclée EXEC mdvl_despro
- 7.- Teclée PAUSE y pulse ENTER
- 8.- Pulse SPACE y espera a que en la pantalla se dibuje deformado el código OK y aparezca el mensaje Check OK through lens
- 9.- Pulse SPACE. Supercharge le solicita introducir los caracteres señalados en la parte superior izquierda tras la "X". El mensaje es : Please identify these characters. Después de haberlo hecho aparecerá en la parte superior de la pantalla y recuadrado

DIGITAL PRECISION SUPERCHARGE

Ahora Supercharge nos solicita el nombre del programa, que este tendrá cuando sea compilado. Después le preguntará si deseamos un listado de los errores generados durante la compilación, si existen. Si nuestra respuesta es afirmativa en la siguiente pregunta introduciremos el nombre del fichero en el que dichos errores serán almacenados. Si elegimos la pantalla como fichero, los errores se irán visualizando en la ventana superior.

En este momento Supercharge analiza su programa básico, comprobando si existen errores y traduciéndolo a un código intermedio. Aparecerán líneas coloreadas en la pantalla mientras el compilador está trabajando.

Cualquier error que sea detectado por el compilador hará que este se detenga. Los mensajes de error serán estudiados en el capítulo 3. Si no se encuentran errores, deberá teclear :

EXEC mdvl_codegen_task

y el generador de código producirá el programa compilado con el nombre que Vd. especificó.

EJECUTANDO UN PROGRAMA COMPILADO

Si Vd. llamó a su programa, por ejemplo mdv2_juego, puede ejecutarlo mediante el comando :

```
EXEC_W mdv2_juego.
```

Vd puede correr varios programas al mismo tiempo si utiliza el comando EXEC en lugar del EXEC_W. Cuando un trabajo se carga con EXEC se ejecuta independientemente del SuperBasic, es decir Vd. puede teclear otros comandos mientras el trabajo se está ejecutando.

BUSCANDO MEMORIA EXTRA

Cuando se carga un programa compilado, se reservan automáticamente 2K de memoria para almacenar los datos generados por el programa tales como valores de variables, pointers y detalles acerca de los canales. Si un programa compilado se detiene con el mensaje : OUT OF MEMORY, es señal de que se necesita incrementar este espacio. Esto puede hacerse con el programa DATASPACE. Para ello cargue este programa con :

```
EXEC_W mdv1_DATASPACE_TASK
```

Introduzca el nombre del programa que desea modificar. Se le mostrará el tamaño y la situación de los datos. Teclée el nuevo espacio en kilobytes. Pulse ENTER. Para más información lea el capítulo 4.

SUPERCHARGER CONTRA SUPERBASIC

Esto es simplemente un resumen de las diferencias existentes entre el intérprete y el compilador. Puede encontrar más información y ejemplos en el capítulo 6. La principal diferencia es que el programa compilado trabaja mucho más rápido y permite la multitarea.

Supercharger es casi totalmente compatible con SuperBasic, pero existen algunas discrepancias.

- 1.- Comandos tales como LIST, MERGE Y CONTINUE no son admitidos.
- 2.- Los valores en coma flotante son calculados y visualizados con 9 cifras decimales.
- 3.- Las sentencias DIM deben usarse para declarar cadenas de más de 256 caracteres. El rango debe ser entero y hasta un valor de 32767. Solamente cadenas y subcadenas pueden ser troceadas
- 4.- Todos los bucles, procedimientos y funciones deben tener su correspondiente END.
- 5.- Sentencias tales como GOTO A*10 no son permitidos.
- 6.- Las líneas DATA deben ser fijas.

2 UNA INTRODUCCION MAS PAUSADA

Este manual describe el programa Supercharge, una poderosa herramienta que traduce programas SuperBasic a código máquina.

Es de gran ayuda a programadores con conocimientos de SuperBasic, ya que les proporciona programas que se ejecutan a una elevada velocidad. Los programas compilados pueden ser cargados y ejecutados sin necesidad de tener cargado el Supercharge.

QUE ES UN COMPILADOR

Un compilador convierte programas escritos en lenguaje de alto nivel en programas escritos en código máquina.

El corazón de un ordenador es el procesador. En esencia un procesador solo puede hacer tres cosas :

- mover datos de un sitio a otro dentro de la memoria
- efectuar simples operaciones aritméticas
- seleccionar la siguiente instrucción a ejecutar

Quando se enciende el QL miles de instrucciones, almacenadas en código máquina dentro de la ROM, comienzan a ejecutarse. Existe un programa en la ROM que se conoce con el nombre de "intérprete".

Este programa es exactamente igual que un diccionario, es decir convierte las intrucciones SuperBasic en instrucciones en código máquina.

El intérprete es lento porque la función de traducción debe hacerla una a una, y gasta una gran parte de tiempo localizando cada instrucción SuperBasic convirtiéndola después a código máquina y ejecutándola más tarde.

DIEZ RAZONES PARA SER OPTIMISTAS

1.- Los programas compilados por Supercharger se ejecutan de 10 a 20 veces más rápidos que sus equivalentes SuperBasic

2.- Pueden ejecutarse varios programas al tiempo

3.- Los programas compilados pueden ser comprobados utilizando el intérprete de Basic. Las variables pueden ser alteradas e impresas en cualquier momento. Esto no es posible en otro tipo de programas compilados en lenguajes tales como Pascal, C o Fortran.

4.- Pueden añadirse comandos y funciones a un programa compilado.

5.- Las funciones en coma flotante se realizan a gran velocidad dando hasta 9 decimales.

6.- Las operaciones con números enteros se realizan a una velocidad de treinta veces superior a su equivalente en SuperBasic.

7.- Los programas compilados se cargan más rápidamente que los no compilados.

8.- Los programas compilados no pueden listarse, dando así protección adicional

9.- Un gran número de restricciones impuestas por el intérprete no son aplicables a los programas compilados.

10.-Cualquier tipo de error durante la compilación queda claramente definido, evitando los mensajes criptográficos.

3 UTILIZANDO EL SUPERCHARGE

EL PROCESO DE COMPILACION

Supercharge trabaja en dos etapas, con objeto de dejar libre la mayor cantidad posible de memoria. Ambas etapas se enlazan automáticamente.

En la primera etapa, un programa llamado "parser" convierte el programa Basic en un código intermedio, que se almacena en memoria si el espacio lo permite o, en el microdrive en caso contrario. El programa "parser" comprueba además que su programa Basic no contiene errores, enviándole, si los hubiera los correspondientes mensajes.

En la segunda etapa el código intermedio es convertido a código máquina para que sea entendido por el microprocesador 68008. Este trabajo es llevado a cabo por el generador de código, que se carga en la parte superior del "parser".

EMPEZANDO

Antes de compilar un programa deberá tenerlo cargado en memoria.

Si recibe el mensaje OUT OF MEMORY, deberá reducir el tamaño de su programa Basic o asignar más memoria. Si esto último no puede hacerlo deberá dividir su programa en dos o más partes independientes, compilándolas por separado.

La primera pregunta que se le hará será :

Name for compiled program ?

Nombre del programa compilado ?

Deberá introducir un nombre precedido del dispositivo, es decir mdv1,mdv2,etc.

Si el nombre dado por Vd. no es válido, el programa le repetirá la pregunta.

Si el nombre dado ya existiera recibirá el siguiente mensaje :

Do you want delete it (Y/N) ?

· Desea anularlo (Y/N) ?

Si pulsa Y el programa antiguo será anulado y si pulsa N le será solicitado un nuevo nombre.

Una vez que haya introducido un nombre aceptable se le preguntará :

Compilation listing (Y/N) ?

Listado de compilación (Y/N) ?

Si pulsa ENTER se asume Y.

Por último se le solicita si desea el informe generado por el compilador. Este listado contiene todos los mensajes de error y de aviso.

Report File (ENTER - SCR) ?

Informe (ENTER - SCR) ?

Introduzca el nombre del fichero o dispositivo al que desea sea enviado dicho informe. Si tiene una impresora conectada al ser1, pulse "ser". Si desea enviarlo a un microdrive teclée por ejemplo mdvl_juego_rep, si el programa se llamaba juego, etc. Si solamente pulsa ENTER el informe se enviará a la pantalla del QL. Puede parar la imagen pulsando CTRL y F5. Cualquier otra tecla reiniciará el report.

EL ANALIZADOR DEL PROGRAMA

El programa "parser" se desplaza a través de su programa desde el principio hasta el fin dos veces. Cada desplazamiento se denomina "pasada". Durante estas pasadas Supercharge visualiza el número de línea cada vez que es analizada.

Durante la segunda pasada el programa es analizado en detalle. Cualquier error que se detecte es visualizado y se envía el correspondiente mensaje de error junto con el número de línea que lo ha producido. Los mensajes producidos por Supercharge están precedidos de 4 asteriscos, para distinguirlos de cualquier otro.

Si solicita un listado, los errores se encuentran justo después del punto que lo ha producido. En el caso de algunos errores tales como omisiones de END's o "sentencias ambiguas" la auténtica causa del error puede hallarse en cualquier parte del programa.

Al contrario que el interprete de Basic, Supercharge examina todas las instrucciones del programa, pudiendo detectar fallos que no hubieran sido encontrados en el programa Basic.

Una vez que el error ha sido detectado el compilador cesa su ejecución, sin intentar efectuar la conversión a código máquina.

Corrija los errores y vuelva a ejecutar el Supercharge.

GENERACION DEL CODIGO

Si no han existido errores al final de la segunda pasada, el código intermedio generado por "parser" es convertido automáticamente a código máquina.

EJECUTANDO UN PROGRAMA COMPILADO

Vd. puede ejecutar un programa compilado tecleando :

```
exec_w mdvl_nombre del programa
```

ERRORES EN LA COMPILACION

El compilador puede detenerse por cualquier mensaje de error, por ser un programa excesivamente largo o por otras circunstancias.

Si posee expansión de memoria puede incrementar el area destinada al compilador. Esto puede hacerse utilizando el programa DATASPACE, tal como se explica en el capítulo 6, en el apartado "Tareas y Memoria".

Los ficheros que deberá modificar se llaman PARSER y CODEGEN. Estos corresponden a las dos etapas de la compilación.

Si ocurriese un error distinto de OUT OF MEMORY, deberá efectuar un reset y comenzar de nuevo. Si el error persiste examine la última línea visualizada antes de producirse el error. Compruebe que su programa no contraviene ninguna de las restricciones que se explican en el capítulo 6 de este manual.

Recuerde que Supercharge está diseñado para compilar programas que antes han sido ejecutados. Si su programa no trabaja en "modo normal", no espere que tampoco lo haga compilado.

RESUMEN DE ERRORES

La mayoría de los mensajes generados por Supercharge son autoexplicativos. Algunos de estos mensajes son solamente "avisos", es decir, pequeños errores que no impiden la compilación.

AMBIGUOUS NAME USED
NOMBRE AMBIGUO

La línea correspondiente tiene un nombre de tipo indeterminado, quizás se ha declarado como una matriz y luego se ha utilizado como una función. Este mensaje puede generarse más de una vez con la intención de que la detección pueda realizarse con más facilidad.

AMBIGUOUS DECLARATION OF NAME
DECLARACION AMBIGUA

Se ha efectuado una declaración incompatible con una matriz, función o procedimiento

ARRAY NAME REQUIRED
SE PRECISA NOMBRE DE MATRIZ

El primer parámetro de una sentencia DIM debe de ser el nombre de la matriz. Este mensaje aparece si dicho nombre falta o si dos matrices distintas aparecen con el mismo nombre.

ARRAY OPERATION NOT IMPLEMENTED
OPERACION CON MATRICES NO IMPLANTADA

El compilador ha detectado una operación de troceado con valores numéricos.

ASSIGNMENT TO FUNCTION ATTEMPTED
ASIGNACION INCORRECTA

Se ha encontrado una sentencia con el nombre de una función. Quizás se ha intentado dar el valor de una variable al nombre de una función. es equivalente al mensaje Basic "Bad name"

CHANNEL SPECIFICATION NEEDED
SE PRECISA ESPECIFICAR EL CANAL

Se necesita dar el número de canal

COMMAND MEANINGLESS IF COMPILED
COMANDO SIN SIGNIFICADO SI SE COMPILA

El comando especificado es incompatible en el programa compilado.

COMPILATION ABORTED
COMPILACION ABORTADA

Los errores detectados hacen imposible la compilación. Cualquier error posterior no será detectado.

END OF STATEMENT EXPECTED
FIN DE SENTENCIA ESPERADO

El compilador entiende que falta un END. Este mensaje aparece si Vd. trata de pasar un número erróneo de parámetros a un procedimiento o si cierra más paréntesis de los que ha abierto.

END IF EXPECTED
SE ESPERA UN ENDIF

Se ha encontrado un IF sin su correspondiente ENDIF

END REPEAT EXPECTED
SE ESPERA UN END REPEAT

Las reglas de anidamiento no han sido respetadas.

END SELECT EXPECTED
SE ESPERA UN END SELECT

Igual al anterior

ERROR DIAGNOSIS FAILED
ERROR DE DIAGNOSTICO

A ocurrido un error interno. Examine la última línea por si fuera un error obvio. También puede deberse a un fallo de su QL o de la copia del programa Supercharge.

EXPRESSION NOT ALLOWED IN DATA
EXPRESSION NO PERMITIDA EN DATA

Las sentencias DATA no deben contener expresiones. Pueden contener cadenas encerradas entre comillas o números

EXPRESSION SYNTAX INCORRECT
EXPRESSION SINTACTICA INCORRECTA

Puede deberse a un número no aparejado de paréntesis o caracteres ilegales en la expresión

EXPRESSION TOO COMPLEX
EXPRESSION MUY COMPLEJA

Ocurre cuando introduce una expresión realmente compleja, con 20 paréntesis o más, o una larga serie de operadores unitarios o muchas funciones que se llaman unas a otras. Es Vd. demasiado listo para el compilador. La solución consiste en dividir la expresión en otras más sencillas.

FAULTY LINE NUMBER
FALLO EN EL NUMERO DE LINEA

El número de línea especificado en un GOTO, RESTORE, GOSUB o ON GOTO deben de ser valores fijos, no el resultado de un cálculo.

FUNCTIONS MUST RETURN A VALUE
LAS FUNCIONES DEBEN DEVOLVER UN VALOR

Se ha encontrado una sentencia RETURN en la definición de una función sin tener un valor asociado que devolver. Si el valor es inmaterial utilice RETURN 0 para evitar el error.

INCORRECT NUMBER OF PARAMETERS
NUMERO INCORRECTO DE PARAMETROS

Auto-explicativo

INCORRECT SUPERBASIC SYNTAX
SINTAXIS SUPERBASIC INCORRECTA

Se ha hallado un error sintáctico SuperBasic

LOCALS MUST FOLLOW DEFINITIONS
LAS VARIABLES LOCALES DEBEN PRECEDER A LA FUNCION

Las variables locales deben definirse al principio del procedimiento o función.

LOOP DOES NOT EXIT HERE
EL BUCLE NO ACABA AQUI

Una sentencia NEXT o EXIT se ha encontrado antes del END

MISSING ARRAY SUBSCRIPT
MATRIZ NO HALLADA

El programa contiene referencia a una matriz no definida

NAME NOT FOUND
NOMBRE NO ENCONTRADO

A ocurrido un error interno. Puede deberse a interferencias con otros programas dentro de la memoria o si el programa a compilar se ha corrompido.

NON-EXISTENT LOOP OR SELECTION
BUCLE O SELECCION NO EXISTENTE

Una estructura tiene más de un END. También puede deberse a duplicidad en END IF, END FOR, END SELECT o END REPEAT.

ONLY FUNCTIONS MAY RETURN VALUES
SOLO LAS FUNCIONES DEVUELVEN VALORES

No deben utilizarse sentencias tales como RETURN var dentro de las definiciones de procedimientos

PREVIOUS DEFINITION INCOMPLETE
DEFINICION ANTERIOR INCOMPLETA

El comienzo en la definición de un procedimiento o función se ha encontrado antes que el END DEFINE del anterior.

PROCEDURES DO NOT HAVE VALUES
LOS PROCEDIMIENTOS NO TIENEN VALORES

El nombre de un procedimiento, que no tiene un valor asociado, se ha encontrado en una expresión aritmética.

STATEMENT IS NOT YET SUPPORTED
SENTENCIA AUN NO IMPLANTADA

La sentencia se encuentra en la ROM pero aún no forma parte del SuperBasic. Este mensaje aparece si se utiliza INPUT o EOF en un contexto inválido.

TOO MANY STRUCTURES
DEMASIADAS ESTRUCTURAS

Auto-explicativo

UNEXPECTED SYMBOL IN SUPERBASIC
SIMBOLOS NO RECONOCIBLES EN SUPERBASIC

Este error es debido a fallos en el QL o en el programa

VARIABLE ASSIGNMENT EXPECTED
SE ESPERA ASIGNAR VARIABLE

Se ha encontrado una referencia a una variable al principio de la instrucción y no ha sido precedida del signo igual

MENSAJES DE AVISO

Los próximos mensajes no son errores y no impiden por tanto la compilación de un programa.

WARNING; DIM STRINGNAME \$(256) ASSUMED
ATENCIÓN DIM NOMBRE DE CADENA (256)

La variable no está dimensionada en nuestro programa y le supone una DIM de 256.

WARNING: LOCAL STRINGNAME \$(256) ASSUMED
ATENCION: LOCAL NOMBRE DE CADENA (256)

Igual al anterior

WARNING: END DEFINE sub ASSUMED
ATENCION SE SUPONE END DEFINE

Auto-explicativo
WARNING: PARAMETERS ARE NOT RETURNED
ATENCION: NO SE DEVUELVEN PARAMETROS

Auto-explicativo

WARNING: VARIABLE NAME ASSUMED
ATENCION: SE ASUME NOMBRE DE VARIABLE

Supercharge no puede asumir si el nombre que ha encontrado es el de una variable o el de un fichero.

ERRORES DURANTE LA EJECUCION

Los errores que se produzcan durante la ejecución de un programa compilado son los proporcionados por el sistema operativo del QL.

4 MULTITAREA

ESPACIO, TIEMPO Y MULTITAREA

Una "tarea" es cualquier programa cargado mediante el comando EXEC o EXEC_

Tales programas se diferencian de los SuperBasic o de los programas en código máquina cargados mediante LBYTES en que pueden ejecutarse más de uno al mismo tiempo. A esto se le llama "multitarea"

Todos los programas compilados con Supercharge admiten la multitarea, en otras palabras, pueden ejecutar un montón de programas al mismo tiempo. Todos los programas comienzan a ejecutarse en círculo, es decir, el procesador va dándoles consecutivamente atención. Al ser el proceso muy rápido Vd. cree que todos se están ejecutando a la vez.

La principal limitación de esta modalidad radica en el hardware y particularmente en la cantidad de memoria.

Cada programa compilado requiere al menos 4 Kilobytes de memoria. Este espacio contiene 2K de espacio para almacenar datos y siempre se asigna. Más adelante explicaremos como se puede alterar este espacio.

Cuando se efectúa la multitarea el ordenador debe distribuir los dispositivos de forma que puedan ser utilizados por todos los programas que se están ejecutando. Algunos como la pantalla puede ser fácilmente compartido, sin embargo otros como el teclado resultan más problemáticos.

La utilización de dispositivos sería también ocasiona problemas. Imagine el caos que podría organizarse si por la misma impresora salen mezclados distintos informes de distintos programas.

Algunos trabajos pueden utilizar un solo microdrive o disco.

La visualización sobre la pantalla puede organizarse adecuadamente con el uso apropiado de "ventanas"

TAREAS Y MEMORIA

Cuando un programa se carga con EXEC o EXEC_W se reserva automáticamente espacio para los datos.

Supercharge utiliza este espacio para almacenar la información generada cuando el programa se ejecuta. Cuando este espacio se agota el programa se detiene con el mensaje OUT OF MEMORY.

El programa llamado DATASPACE le permite modificar este espacio. Para ello teclee :

```
EXEC mdvl_DATASPACE_TASK
```

Una vez que el programa esté cargado aparecerá una ventana en la parte superior de la pantalla, y la pregunta "Task file name ?", Nombre de la tarea ?, que indica que el programa está esperando el nombre del programa que ha de ser modificado.

Pulse Control C, hasta que el cursor esté parpadeando en la ventana. Teclee el dispositivo y nombre del programa a modificar. Si no está seguro del nombre salgase de la ventana y teclee el comando DIR.

En cualquier momento puede detener el programa DATASPACE pulsando simplemente ENTER.

Despues de haber introducido el nombre del dispositivo y del programa se visualizará el tamaño del programa el número de bytes del código y el número de bytes asignados a los datos. Así por ejemplo :

3836 code bytes, 2048 data bytes
New data size (in Kilobytes) ?
Nuevo tamaño (en Kbytes) ?

Nunca se admitirá un valor inferior a 2.

5 EXTENSIONES AL SUPERBASIC

EXTENSIONES AL SUPERBASIC

Con el Supercharge se facilitan 5 nuevos comandos, que pueden ser incluidos en SuperBasic. Estos comandos le posibilitan el control sobre sus trabajos y le permiten encontrar errores. Estos nuevos comandos pueden ser utilizados bien en programas normales de basic o en programas compilados.

Estos nuevos comandos se cargan mediante la instrucción :

```
LRUN mdvl_BOOT
```

Esto hace que las ampliaciones al SuperBasic sean instaladas en la memoria. Estos comandos permaneceran en la memoria de su QL hasta que efectue un RESET o apage el ordenador. Solamente utilizan 512 bytes de memoria.

Para utilizar estos nuevos comandos antes debe como es logico cargarlos

De estos comandos tres de ellos le facultan para ejercer un control sobre los trabajos o tareas. Estos comandos trabajan con cualquier programa que se cargue mediante la instrucción EXEC o cualquier otro lenguaje que admita ser lanzado mediante la instrucción EXEC. Los otros comandos utilizados para hallar la cantidad de memoria libre y errores seran discutidos más adelante.

VIENDO QUE TAREAS SE ESTAN EJECUTANDO

El comando LIST_TASKS produce un listado de todos los trabajos que se estan ejecutando en el momento. Dicho listado consiste en cuatro columnas separadas por comas :

Nombre, Número, Etiqueta, Prioridad

Puede dirigir este listado a cualquier dispositivo del QL seguido del número del canal correspondiente en la misma forma que hace con las instrucciones PRINT o DIR. el caracter # es opcional. Así para enviar el listado a la ventana #0, deberá pulsar :

```
LIST_TASKS #0
```

Si Vd. pulsa LIST_TASKS antes de que un trabajo haya sido lanzado, obtendrá la respuesta :

```
BASIC, 0 ,0, 32.
```

Esta línea indica que solamente está ejecutandose un trabajo Basic. Si existieran más trabajos aparecerían un líneas posteriores.

La primera información es el nombre del trabajo, en nuestro ejemplo, BASIC. Los programas compilados poseen el nombre de Supercharge.

Después del nombre vienen dos números que identifican el trabajo al sistema operativo del QL. Estos números se llaman "números de trabajo". Estos números son necesarios cuando Vd. utiliza los comandos del TASK_CONTROL. Desgraciadamente se utilizan dos números en lugar de uno solo, pero esta es una característica impuesta por el propio sistema operativo del QL. El intérprete de Basic siempre asigna los números 0,0 a los programas Basic.

El último número es la prioridad de la tarea. Cuando solamente existe un trabajo ejecutándose, este número no es importante. Cuando existen varios trabajos determina la proporción de tiempo que asigna el QL a cada uno de ellos.

Los números de prioridad van desde el 0 hasta el 127. Si un trabajo tiene prioridad cero no se ejecutará. Más adelante explicaremos como una prioridad puede ser modificada. Pero primero veremos con más detenimiento lo que significa esta palabra.

TRATAMIENTO DE LAS PRIORIDADES

Si hay tres trabajos ejecutándose, y todos tienen prioridad 32 (el valor standard dado por el comando EXEC), reciban exactamente la misma atención por parte del QL y corran a la misma velocidad. Si la prioridad de uno de ellos se reduce, este trabajo recibirá mucha menos atención que los otros dos restantes y parecerá que se ejecuta mucho más lentamente. Es una buena norma el evitar dar prioridades excesivamente altas a un trabajo, excepto en casos excepcionales.

La exacta relación de tiempo asignado depende del tipo de trabajo que se esté realizando. En general los trabajos con mayor prioridad reciben la mayor proporción de tiempo, aunque esto no es siempre cierto. Si dos trabajos están esperando recibir información (por ejemplo desde el teclado o desde un canal serie), el QL no malgasta su tiempo en ellos, hasta que tenga datos que procesar. En este caso el trabajo con prioridad inferior recibe más atención simplemente porque es el único trabajo que no precisa de una entrada exterior.

El QL no se "olvida" de los trabajos pendientes, a menos que su prioridad sea cero. Incluso aunque su prioridad sea 1, recibirá atención ocasionalmente.

A veces puede observar el proceso trabajando. Suele ser común asignar prioridades de "reloj" o de "calendario", tales que visualizan la hora actual, y generalmente requieren poca atención por parte del QL. Si tiene algún programa de este tipo podrá observar que el reloj marca la hora correcta, excepto cuando introduce algún dato por el teclado, en donde generalmente se observa como aparentemente se pierden algunos segundos en la visualización.

MODIFICANDO LA PRIORIDAD DE LOS TRABAJOS

Siempre puede modificarse la prioridad de un trabajo una vez cargado. El comando para realizarlo es :

```
SET_PRIORITY
```

El QL necesita conocer dos cosas para poder efectuar el cambio : "el número de identificación" (el número y la etiqueta) y la nueva prioridad. Utilice el comando LIST_TASKS para hallar el número y la etiqueta. El formato del comando es :

SET_PRIORITY 0,0,16.

Este comando muestra como el trabajo 0,0 se le asigna la prioridad 16. No se permite poner la prioridad de un trabajo 0,0 (Basic) a cero, ya que la entrada de más comandos sería imposible. De todas formas si lo intenta recibirá el mensaje "Bad parameter", (parámetro erróneo). Si el trabajo especificado no existe recibirá el mensaje "Invalid job" (trabajo inválido).

DETENIENDO TRABAJOS

También pueden quitarse trabajos de la memoria del QL con el comando REMOVE_TASK.

El formato es REMOVE_TASK 1,1 en donde 1,1 representa el "número de identificación".

Al igual que anteriormente es preciso que el trabajo especificado exista. También es imposible eliminar el trabajo 0,0.

Cuando un trabajo es eliminado, todos los canales por el utilizados son cerrados, los dispositivos son liberados para otros usos y la memoria ocupada es limpiada.

MEMORIA

La función FREE_MEMORY devuelve el valor del espacio disponible para SuperBasic, o la cantidad de espacio no utilizado dentro de un área de trabajo. El formato es :

PRINT FREE_MEMORY

ATRAPANDO ERRORES

Existen ciertos comandos que pueden hacer que un programa aborte, sin que Supercharge pueda reconocerlos. Esto ocurre por dos razones :

1.- No han sido formalmente especificados ni documentados, de forma que su operación puede cambiar de una versión a otra del QL. Supercharge está diseñado para funcionar con todas las versiones del QL.

2.- Existen grandes errores en los nuevos comandos (al menos en la ROM "MG")

Por ello no hemos implantado en Supercharge los comandos WHEN.

Sin embargo hemos provisto a Supercharge de una función DEVICE_STATUS, que comprueba los posibles errores en la mayoría de las áreas comunes.

DEVICE_STATUS es una función que espera un simple parámetro. Este parámetro deberá tener el nombre de un dispositivo del QL o el nombre de un fichero.

Esta función analiza si el dispositivo se encuentra abierto o no. Así por ejemplo la expresión y sus parámetros CON_448x180A32x16 serán analizados exhaustivamente.

Finalmente la función intentará abrir el canal para comunicarse con el dispositivo elegido.

RETURN_STATUS devuelve un número que indica el grado de éxito que ha acompañado a la operación. Estos números se encuentran tabulados más adelante.

La función se adapta automáticamente al hardware existente, de forma que Ud. puede confiar plenamente en que el sistema se adaptará a cualquier configuración de floppy disks, modems, impresoras paralelas, etc.

Así por ejemplo el comando :

```
PRINT DEVICE_STATUS (mdvl_Supercharge_spec)
```

3

Indica que el fichero Supercharge_spec, existe en el microdrive 1

Valor devuelto	Significado
0 ó más =	El dispositivo existe y no está ocupado.
3 ó 6 =	El dispositivo y los parámetros son válidos pero el QL no tiene suficiente espacio para abrir un nuevo canal.
7 =	No existe dispositivo con el nombre especificado
8 =	Existe un fichero con el nombre especificado sobre el dispositivo indicado
9 =	El dispositivo existe pero ya está utilizado y no puede utilizarse hasta que el trabajo en curso finalice, o bien el fichero especificado está en proceso de escritura.
12 =	El nombre del dispositivo es válido, pero el fichero o los parámetros no.

VALIDACION DE ENTRADA (un truco útil)

Si utiliza normalmente WHEN ERROR para detectar una condición de error, cuando los usuarios introducen letras en lugar de números, considere este sencillo "truco". Lea la entrada dentro de una cadena variable, mejor que desde un valor numérico. Luego si A2 es la cadena que desea evaluar, el comando :

```
X = "0" & A2
```

ponida en X el valor de A2 o cero si A2 no es un número. En ambos casos nunca dará error.

COPIANDO LOS COMANDOS DE EXTENSION

En vista del hecho de que `DEVICE_STATUS` es un comando útil, puede copiarlo para poder utilizarlo en sus programas Basic.

Los programas que deberá copiar son :

`extensions_code` y `extensions_bas`.

6 SUPERCHARGE Y EL SUPERBASIC

Los programas que escritos en SuperBasic funcionan adecuadamente, también lo hacen compilados. Sin embargo existen unas pequeñas diferencias que es preciso tener en cuenta. En los siguientes párrafos explicamos estas diferencias y las acciones que deben tomarse si influyen negativamente en sus programas.

- a) Identificadores SuperBasic
- b) Margen y exactitud en operaciones matemáticas
- c) Manejo de ordenaciones y cadenas
- d) Estructuras anidadas
- e) Líneas de cálculo
- f) Edición de programas
- g) Cálculos en DATA.s
- h) Números de canal
- i) Situación de los parámetros

a) Identificadores SuperBasic

Los identificadores se definen en la sección de "Conceptos" de la Guía del Usuario del QL como una secuencia de hasta 255 caracteres, que se usan para identificar variables, procedimientos, funciones y estructuras de programas. A menudo también se les denomina por "nombres".

En SuperBasic Vd. puede utilizar el mismo identificador para diferentes propósitos en distintos puntos a lo largo de sus programas. Por ejemplo Vd. puede utilizar el identificador VECTOR para describir una ordenación unidimensional en una parte del programa y más adelante para definir una matriz tridimensional. No obstante no es recomendable hacerlo ya que puede ser causa de confusión y errores, pero es admitido por el intérprete, que siempre utiliza la última definición del identificador. Si un programa va a ser compilado necesita tener un único nombre y no se permiten varios. Los valores de las variables pueden ser de tres tipos en SuperBasic :

enteros, de coma flotante o cadenas.

El compilador deduce el tipo, examinando el último carácter del identificador. Así por ejemplo, el identificador VECTOR será de coma flotante, mientras que el identificador VECTOR% será entero y VECTOR\$ será una cadena.

Esta regla difiere un poco del SuperBasic en donde el tipo de un parámetro pasado a un procedimiento o función es determinado por el tipo del correspondiente valor. En programas compilados, el tipo de cada parámetro ha de ser conocido en el momento de la compilación.

Existen tres usos distintos para un identificador :

1) Variables simples, enteros, coma flotante y cadenas. Estos nombres pueden usarse para almacenar valores o para identificar bucles FOR y REPEAT. Solamente pueden utilizarse variables simples en la primera línea de una sentencia SELECT. Esto se corresponde con las reglas del SuperBasic, con la ventaja de poder utilizar identificadores enteros y de cadena en sentencias FOR, REPEAT y SELECT, si su intérprete de SuperBasic se lo permite.

2) **Variables de Ordenación.** Los nombres utilizados en las ordenaciones y declarados mediante sentencias DIM y LOCAL. Un identificador de ordenación puede ser declarado más de una vez en un programa, pero el número de dimensiones (y no necesariamente su tamaño), debe de ser el mismo en cada declaración.

3) **Nombres de definición.** Son los nombres utilizados en los procedimientos y funciones. Cada procedimiento o función debe tener un único nombre.

Si un identificador es utilizado para cualquiera de uno de estos propósitos, no puede ser utilizado para ningún otro. Si se hace, el compilador enviará un mensaje de "ambiguo" y marcará claramente el nombre. La única solución si Vd. desea compilar el programa es el de cambiar todos los nombres "ambiguos" asignando otros nombres nuevos.

b) Margen y exactitud en operaciones matemáticas

El compilador soporta el mismo margen de exactitud que el intérprete de SuperBasic. Sin embargo la exactitud visualizada es mayor en los cálculos de coma flotante, 9 decimales frente a los 7 u 8 del intérprete.

TRABAJE EN UNIDADES ENTERAS cuando necesite una precisión exacta. Esto significa que cuando trabaje con programas de gestión en los que deba trabajar con valor de moneda, deberá introducir el valor de los céntimos o sus equivalentes.

SuperBasic, ya sea compilado o interpretado, almacena los números en notación binaria. Esto significa que las fracciones son almacenadas en la misma forma, es decir, en potencias recíprocas de 2. Así, por ejemplo, el valor $3/8$ es almacenado como 0.11 en binario ($1/2 + 1/4$), pero muchas otras fracciones, tales como $1/10$ ó $1/100$ no pueden expresarse como una fracción exacta. Sin embargo si Vd. introduce valores exactos, no existe el problema, ya que únicamente las fracciones superiores a $1E9$ serán inexactas.

La exactitud en un programa compilado nunca es menor que su homólogo en intérprete.

Las variables enteras no son procesadas en el intérprete de SuperBasic a mayor velocidad, ya que son convertidas a coma flotante para efectuar los cálculos. Sin embargo utilizan una menor cantidad de memoria (2 bits frente a 6), que las variables en coma flotante.

Sin embargo Supercharge efectúa los cálculos con enteros a mayor velocidad que en coma flotante. Es aconsejable utilizar la aritmética de números enteros siempre que sea posible, especialmente en bucles y en cálculos con ordenaciones.

c) Manejo de ordenaciones y cadenas

Con las típicas vaguedades de la Guía del Usuario del QL, se dice que "bajo ciertas circunstancias" un nombre puede ser utilizado para referenciar más de un elemento de una ordenación. Estas circunstancias son más restrictivas con el compilador que con el intérprete.

Los programas compilados ocupan una cantidad "estática" de memoria si van a ser ejecutados en modalidad multitarea. Por ello a todas las ordenaciones unidimensionales se les asigna una longitud fija de 256 caracteres.

Si encuentra este límite restrictivo puede evitarlo dimensionando explícitamente dicha ordenación, así por ejemplo, si Vd. necesita almacenar 4000 caracteres en la cadena PARAS utilice el comando :

```
DIM PARAS(4000)
```

al comienzo de su programa.

Es importante no asignar más espacio del necesario, ya que cada sentencia DIM incrementa el tamaño del programa compilado.

NOTA 1 El comando CLEAR hace que todas las ordenaciones y cadenas queden sin dimensionar. Por lo tanto deberá dimensionarlas de nuevo.

NOTA 2 En algunas circunstancias el intérprete devuelve la longitud de una cadena cuando se pregunta por su elemento cero. Supercharge lo ignora.

El número total de elementos en una ordenación no puede superar el número 65355. Este límite permite al compilador acceder a gran velocidad a los elementos de una ordenación.

d) Estructuras anidadas

Los procedimientos y funciones no pueden estar anidados. En otras palabras, el final de una definición debe preceder al comienzo de la siguiente. Así por ejemplo :

```
10 REMark Bucle legal
20 FOR I=1 TO 10
30 REPEAT X
40 END REPEAT X
50 END FOR I
```

```
10 REMark Bucle ilegal
20 FOR I=1 TO 10
30 REPEAT X
40 END FOR I
50 END REPEAT X
```

e) Líneas de cálculo

El intérprete de SuperBasic permite que un comando envíe el control a cualquier número de línea mediante un cálculo,

En programas compilados solo se permite especificar un número constante cuando se requiere un número de línea. Esta regla afecta a los comandos GOTO, GOSUB y RESTORE. Si Vd. necesita calcular una línea deberá utilizar los comandos ON GOTO y ON GOSUB TO. Alternativamente puede utilizar la construcción SELECT que es más flexible y le permite elegir entre varias alternativas. En programas compilados SELECT e IF deberán ser usados en lugar de RESTORE, ya que SuperBasic no soporta sentencias del tipo ON RESTORE.

f) Edición de programas

El editor de SuperBasic está incluido dentro del intérprete. Los comandos AUTO, DLINE, LIST, LOAD, MERGE, MRUN, RENUM y SAVE no están soportados por el compilador.

Los comandos CONTINUE y RETRY tampoco lo están ya que están diseñados para la corrección de errores y esto no es posible en un programa compilado.

g) Cálculos en los DATA's

Supercharge no permite expresiones de cálculo en las sentencias DATA, ya que ello complicaría la ejecución y generación del código compilado. Solo se permiten cadenas y valores numéricos constantes, aunque existe una excepción en lo que se refiere a los operadores unitarios, +, - y NOT.

h) Números de canal

La Guía del Usuario del QL no establece un número máximo de canales a utilizar en sentencias tales como OPEN y CLOSE. En efecto, cuando se utiliza el intérprete de SuperBasic el máximo número de canales solo depende de la cantidad de memoria disponible. Los programas compilados deberán ejecutarse dentro de un área limitada de memoria, de forma que Supercharge pueda establecer sitio para 16 canales, numerados del 0 al 15. Esto debería ser suficiente para la mayoría de los programas. Recuerde que puede utilizarse un canal una vez que los ficheros asociados a él hayan sido previamente cerrados.

i) Situación de los parámetros

Cuando se utiliza Supercharge, los parámetros (variables declaradas en sentencias DEF), son locales con respecto a la definición en la que aparecen. Los cambios en sus valores no afectan a las variables externas. En otras palabras los cambios a un parámetro nunca afectan a las variables de las cuales el parámetro se deriva.

Si Vd. desea compilar un programa SuperBasic que pasa valores desde un procedimiento o función mediante una lista de parámetros, deberá usar variables "globales", es decir una que exista fuera de la rutina desde la que esta siendo llamada para pasar y almacenar el valor, mejor que un parámetro. Esto, ya lo sabemos, no es nada elegante, pero le garantizará que su programa funcionará adecuadamente cuando sea compilado.

ERRORES CORREGIDOS POR EL COMPILADOR

Aunque el SuperBasic es un sofisticado y flexible lenguaje de programación, existen un elevado número de fallos en su intérprete. Nos hemos tomado la libertad de corregir muchos de estos fallos con Supercharge.

Estos fallos son :

a.- El intérprete de Basic puede quedarse "colgado" si más de nueve parámetros o variables locales se utilizan en un procedimiento o función. Supercharge admite cualquier número de variables o parámetros sin ningún problema.

b.- Los resultados matemáticos se visualizan con un máximo de siete decimales en el intérprete, aunque internamente lo haga con mayor precisión. El compilador visualiza nueve decimales.

c.- SuperBasic se detiene con un mensaje de "buffer full" si se introducen más de 128 caracteres en respuesta a un INPUT, en las versiones AH y 3M del QL. El compilador permite la introducción de hasta 32767 caracteres, aunque realmente esta cantidad depende de la memoria disponible.

d.- El intérprete no permite la multitarea en programas Basic. Con programas compilados pueden ejecutarse gran cantidad de programas al mismo tiempo, siempre que de nuevo, tenga la suficiente memoria disponible.

e.- El intérprete no permite manejar sentencias SELECT con enteros o con cadenas. Los programas compilados trabajan perfectamente con enteros y cadenas en sentencias SELECT.

f.- Las variables pasadas a los procedimientos no pueden ser utilizadas en sentencias SELECT, en la versión JS del QL. El compilador no impone esta restricción.

g.- La función Basic RESPR, utilizada para reservar memoria, no trabaja bajo el intérprete, si se está ejecutando una multitarea. El compilador siempre lo permite.

h.- El intérprete no ejecuta una línea simple (forma corta), en los bucles FOR si estos contienen sentencias del tipo GOSUB. Esta restricción no es aplicable a los programas compilados con Supercharge.

i.- Las sentencias CALL, utilizadas para llamar a rutinas en código máquina, hacen que el intérprete se quede "colgado", si la longitud de tales programas es superior a 32K en las versiones hasta JM. Esto no lo hace el compilador.

j.- Los bucles FOR no permiten el uso de los valores enteros, por ejemplo :
FOR I%=1 TO 10. Supercharge puede generar el código máquina sin problemas. No obstante las actuales versiones del editor del QL no permiten la introducción de tales líneas.

k.- El valor de la variable identificadora de un bucle FOR o REPEAT es puesto a cero cuando dicha sentencia es ejecutada por el intérprete. Por ejemplo las sentencias :

```
10 LET T=3  
20 FOR T=T TO 6: PRINT T
```

imprimen valores desde 0 hasta 6, en lugar de hacerlo desde 3 hasta 6. Este fallo es corregido por Supercharge.

LO QUE PUEDE HACER Y LO QUE NO PUEDE HACER UN COMPILADOR

Supercharge incrementa en un factor muy elevado la velocidad de ejecución de los programas, simplemente racionalizando los pasos a seguir. En general este factor se encuentra entre 5 y 20. Los programas largos sufren un mayor incremento de su velocidad.

Supercharge y el hardware

Supercharge no es más que un programa y no puede modificar su hardware. Si el tiempo invertido por un programa está dominado por la transferencia de datos desde los microdrives, el proceso de compilación no podrá hacer maravillas.

Sin embargo Supercharge aumentará espectacularmente la velocidad de los programas que copian datos, especialmente en dispositivos pseudo-aleatorios, tales como discos o microdrives, ya que reduce el tiempo entre lectura y escritura y de esta forma los datos pueden ser empaquetados más densamente. La única forma de saber si su programa está dentro de esta categoría es compilarlo y medir el tiempo antes y después de la compilación.

Los programas compilados se cargan por lo general a mayor velocidad que los no compilados. De igual forma los programas compilados son más seguros, ya que no pueden ser listados. También los valores numéricos son más precisos. Generalmente unas pocas líneas consumen la mayor parte del tiempo de ejecución de un programa. Una buena regla es que el 10 por ciento de un programa se ejecuta en el 90 por ciento del tiempo y viceversa.

En lo referente a bucles anidados, la optimización del bucle interior hará que el programa se acelere considerablemente. Veamos un ejemplo :

```
10 FOR I=0 TO PI STEP PI/180
20 A(J)=A(J) + SIN (I) * COS (TETA)
30 END FOR I
```

Este programa se ejecutará más rápidamente si el cálculo del COS(TETA) se saca fuera del bucle, ya que calculamos dicho COS 180 veces más de lo necesario. Así el programa quedaría :

```
10 TEMP = COS (TETA)
20 TOTAL = A(J)
30 FOR I=0 TO PI STEP PI/180
40 TOTAL = TOTAL + SIN (I) * TEMP
50 END FOR I
60 A(J) = TOTAL
```

Manejo de cadenas

Supercharge y SuperBasic difieren en la forma de tratar las cadenas. El compilador almacena las cadenas en dos partes, una de longitud fija que contiene la dirección de la cadena y su longitud y otra que contiene el texto. El intérprete almacena la longitud y el texto juntos. Esto dificulta el acceso a cadenas anteriores a la última introducida.

El resultado de todo esto es que Supercharge es extremadamente rápido en el troceado y búsqueda de cadenas, siempre que evitemos en estas instrucciones las comparaciones y las llamadas a funciones, ya que en este caso el compilador a de reorganizar todo su almacenamiento.

Aritmética de enteros

Todas las operaciones matemáticas son realizadas por el intérprete en coma flotante. Este procedimiento es complicado y lento. No obstante podría pensarse que si dichas operaciones fueran realizadas sobre números enteros

la velocidad se incrementaría. Desafortunadamente no es así, ya que el intérprete reduce todos los números a coma flotante y podría darse la paradoja que el programa :

```
SUM% = CUENTA% + 1
```

Se ejecutará más lentamente que :

```
SUM = CUENTA + 1
```

Esto es debido a que el intérprete no conoce la forma de sumar dos enteros, sino solo la forma de sumar dos números en coma flotante, por lo que primero convierte los enteros a coma flotante, luego los suma y después los vuelve a convertir a enteros.

Supercharge por el contrario si conoce la forma de sumar números enteros y por ello siempre que sea posible deberemos utilizar en nuestros programas la aritmética entera, ya que la velocidad se verá aumentada entre 5 y 10 veces.

Generación de código en línea

Supercharger cuando compila un programa lo hace en el modo llamado "encajado". este código contiene una mezcla de datos y rutinas con sus correspondientes direcciones. El código se ejecuta saltando de un sitio a otro por medio de las direcciones almacenadas. SuperBasic por el contrario genera un código llamado "en línea", en donde las instrucciones se van ejecutando unas después de otras.

La ventaja del código "encajado" es que es muy reducido y si una operación ha de realizarse varias veces en un mismo programa, solamente aparecerá una vez en el código compilado, ahorrando una considerable cantidad de memoria. No obstante Vd. puede obligar al compilador a generar su código "en línea", simplemente poniendo REMARK + antes de la línea deseada y REMARK - al final de ella. Generalmente el código "en línea" se ejecuta más rápidamente aunque incrementa en forma considerable el gasto de memoria. Así, si intenta compilar programas largos por este método, es posible que no pueda, ya que el compilador impone una restricción de 64K a los programas compilados.

7 GLOSARIO

ABORTED
ABORTADO

Detenido por resultar imposible su continuación.

ADD-COMMANDS
COMANDOS AÑADIDOS

Procedimientos y funciones residentes en código máquina fundidos con el lenguaje SuperBasic, de forma que pueden ser utilizados en la misma forma que las funciones standard del QL.

ARRAY
ORDENACION

Una colección ordenada de cierto número de elementos del mismo tipo. La posición de cada elemento se define de forma unívoca mediante el uso de un conjunto de números llamados índices. El número de índices que se precisa para especificar la posición de un elemento es igual al número de dimensiones de la ordenación. Una ordenación de una sola dimensión suele denominarse vector. Una ordenación de dos dimensiones recibe el nombre de matriz o tabla.

ASSEMBLY LANGUAGE
LENGUAJE ENSAMBLADOR

Una notación para escribir programas en código máquina, utilizando una notación mnemotécnica alfanumérica, de forma comprensible para las personas.

ASSIGNMENT
ASIGNACION

Una sentencia dentro de la mayoría de los lenguajes, incluyendo el SuperBasic, en donde un nuevo valor se asigna a una variable.

BASIC
BASIC

Iniciales de la frase : Beginner's All purpose Symbolic Instruction Code. Es un lenguaje desarrollado en los años 60, partiendo del FORTRAN. Es el lenguaje más utilizado en el mundo de los microordenadores. A pesar de los esfuerzos realizados con objeto de standardizarlo, existen gran cantidad de "dialectos". De todos ellos el SuperBasic es el más potente.

BENCHMARK
BANCO DE PRUEBAS

Una prueba o un conjunto de pruebas diseñadas para medir la características de un sistema de software o de hardware. Fundamentalmente se utiliza para medir la velocidad de ejecución.

BINARY
BINARIO

Que está expresado en el sistema de numeración de base dos. (Solamente se permiten los dígitos 0 y 1).

BITWISE OPERATOR
OPERADOR MODO BIT

Operador lógico que trabaja en estructura de dígito binario en lugar de en estructura Booleana.

BUGS
ERRORES

Errores localizados en un programa o en un sistema. El intérprete de SuperBasic los poseía en elevado número.

CALL
LLAMADA

Instrucción para transferir el control a una subrutina, función o procedimiento con la condición de que una vez ejecutada regrese a la instrucción siguiente a la de la llamada.

CALL BY REFERENCE
LLAMADA POR REFERENCIA

Instrucción para llamar a un procedimiento o función con uno o más parámetros de forma que estos cambien su valor dentro del propio procedimiento o función, causando los correspondientes cambios en los valores que se encuentran fuera.

CALL BY VALUE
LLAMADA POR VALOR

Instrucción para llamar a un procedimiento o función con uno o más parámetros de forma que estos cambien su valor dentro del propio procedimiento o función no causando cambios en los valores que se encuentran fuera. Este se efectúa haciendo una copia exacta de los parámetros en el momento de la llamada y desechando la copia en el momento de regresar.

CASE
MODO

Concepto aplicable únicamente al alfabeto. Puede ser de dos tipos :

UPPER (Mayúsculas, caracteres ASCII del 65 al 90)
LOWER (Minúsculas, caracteres ASCII del 97 al 122)

CENTRAL PROCESSING UNIT
UNIDAD CENTRAL DE PROCESO

La parte operativa más importante del ordenador. Comprende la unidad aritmética y lógica, y la unidad de tiempo para controlar las comunicaciones. El QL posee el microprocesador Motorola M68008.

CHANNEL
CANAL

Un camino para la transmisión de datos hacia un dispositivo.

CODE GENERATOR
GENERADOR DE CODIGO

Parte del compilador, que es la última en actuar, y que se encarga de generar el código de salida hacia el dispositivo.

COMPATIBLE
COMPATIBLE

Hablando de software, es la habilidad de reproducir con fidelidad el comportamiento de sus precededores, (especialmente en lo que se refiere a los formatos de entrada). Supercharge es compatible con SuperBasic.

COMPILER
COMPILADOR

Un programa que traduce un lenguaje de alto nivel a código máquina. En el caso del Supercharge, el lenguaje de alto nivel es el Super-Basic y el código máquina el propio del microprocesador M68008.

COMPILE-TIME
TIEMPO DE COMPILACION

Es el tiempo durante el cual es lenguaje de alto nivel es traducido a código máquina.

CONCURRENCE
CONCURRENCIA

El desarrollo de dos o más trabajos en paralelo sobre un mismo ordenador.

CONDITIONAL
CONDICIONAL

Una sentencia lógica que comprende una condición, que si es satisfecha, dará como resultado una determinada acción.

CONTIGUOUS
CONTIGUO

Inmediatamente adyacente

CORRUPT
CORROMPIDO

Cuando se aplica a un programa o a unos datos, significa "Cambiado de alguna forma". Suele ser sinónimo de espanto para el programador que olvidó efectuar una copia de seguridad. Una corrupción puede ser provocada por un fallo de la máquina o por un fallo en el medio de almacenamiento. Los usuarios del QL, que utilizan microdrives, suelen estar familiarizados con este término.

CURSOR
CURSOR

Un símbolo sobre la pantalla del monitor o televisor que indica la posición activa y que normalmente indica que el ordenador está listo para recibir una entrada.

DEBUGGED PROGRAM
PROGRAMA DEPURADO

Un programa en una situación tal, que Sinclair podría pensar razonablemente en retirarlo.

DEVICE INDEPENDENCE
INDEPENDENCIA DE DISPOSITIVOS

La habilidad dentro de un lenguaje de programación de utilizar los mismos comandos de control en la entrada / salida para diferentes dispositivos. El nombre del dispositivo es lo suficientemente representativo para el control del sistema. SuperBasic y Superchange son independientes respecto a los dispositivos.

DIGITAL PRECISION
DIGITAL PRECISION

Nosotros

DIMENSIONING
DIMENSIONADO

Operación llevada a cabo para definir una ordenación con objeto de definir el número de dimensiones que deberá tener y además asignar el máximo valor del índice correspondiente a dicha dimensión.

DISCRETE
DISCRETO

La capacidad de no variar y no ser capaz de variar en una forma continua, en lugar de en etapas.

ELEMENT
ELEMENTO

Un miembro de una ordenación n-dimensional, referenciado por un conjunto ordenado de n índices.

EPROM
EPROM

Iniciales de : Erasable Programmable ROM. Un dispositivo electrónico capaz de ser programado repetidas veces por el usuario mediante una exposición controlada de rayos ultravioletas.

EXPRESSION
EXPRESSION

Un conjunto de números, cadenas, variables y paréntesis unidos mediante operadores y capaz de ser evaluado.

EXTENSIONS
EXTENSIONES

Otra forma de llamar a los comandos añadidos.

EXTERNAL REFERENCE
REFERENCIA EXTERNA

Una referencia dentro de un programa a otro fuera de él. Hasta que la referencia exacta de la localización del segundo programa es accesible para el primero, se dice que la referencia no está resuelta.

FLOATING POINT
COMA FLOTANTE

Una representación de los números reales dentro de un margen de magnitud.

FUNCTION
FUNCION

Una subrutina que devuelve un valor que puede ser utilizado dentro de una expresión.

GLOSSARY
GLOSARIO

Lo que está viendo ahora

HARDWARE
HARDWARE

En oposición al software, aquellas partes del ordenador que son capaces de ser tocadas. Por ejemplo un microdrive. El hardware es toda la parte mecánica y electrónica que forma y rodea un ordenador.

HEX
HEXADECIMAL

Que está expresado en base 16. (Solo se permiten los dígitos del 1 al 9 y las letras de la A a la F).

HIGH LEVEL LANGUAGE
LENGUAJE DE ALTO NIVEL

Un lenguaje en el que tanto los datos como la estructura, reflejan las necesidades del programador más que las del ordenador. En el QL el SuperBasic es un lenguaje de alto nivel. Supercharge traduce este lenguaje a código máquina.

IDENTIFIER
IDENTIFICADOR

Palabra para designar a una variable

INLINE CODE
CODIGO EN LINEA

Código que no realiza llamadas a una serie de subrutinas

INTEGER
ENTERO

Número que se encuentra en el margen de -32768 a -32767 (ambos inclusive), en lo que a SuperBasic se refiere.

INTERACTIVE
INTERACTIVO

Un modo de trabajo en el que existe una respuesta inmediata o casi inmediata a las órdenes, una vez ha sido introducida una entrada. Así, el intérprete de SuperBasic permite la corrección de errores en modo interactivo. Precisamente estos comandos no están soportados por Supercharge. No obstante se supone que cuando un programa va a ser compilado, el programador se ha asegurado que dicho programa funciona correctamente en Basic y se encuentra depurado de errores.

INTERMEDIATE CODE
CODIGO INTERMEDIO

Una definición muy concisa del código puente generado durante la compilación.

INTERPRETER
INTERPRETE

Un programa para analizar una unidad de código (en SuperBasic una sentencia), en un lenguaje de alto nivel y posteriormente ejecutarla. En el caso del QL, el intérprete de SuperBasic se suministra dentro del propio ordenador. La decisión de suministrar un intérprete en lugar de un compilador es que el intérprete es más sencillo de escribir.

INTERPRETER DATA STRUCTURE
ESTRUCTURA DE DATOS DEL INTERPRETE

La información que posee el intérprete, para mantener una pista de las modificaciones y posteriormente ser capaz de ejecutar el programa.

JOB
TRABAJO

Un conjunto de tareas y de todos sus datos. A veces se utiliza como sinónimo de "task".

KLUDGE
KLUDGE

EPROM de Sinclair

LINE
LINEA

Una sentencia ordenada de uno o más comandos en un programa Basic, referenciados colectivamente por un número.

LINKING
UNION

La combinación de uno o más programas en código máquina, resolviendo todas las referencias externas entre ellos. La unión se utiliza tanto para añadir rutinas ya escritas, como para ensamblar programas complejos. Esta característica no está soportada por Supercharge, ya que el SuperBasic no lo hace. Los programas externos pueden activarse mediante los comandos CALL o EXEC.

LOOP
BUCLE

Una secuencia de instrucciones, dentro de un programa, que se repiten hasta que determinada condición se cumple.

LOW LEVEL LANGUAGE
LENGUAJE DE BAJO NIVEL

Un lenguaje de programación, típicamente el ensamblador o el código máquina, que controla los datos y la estructura y está en contacto directo con la arquitectura del microprocesador.

MACHINE CODE
CODIGO MAQUINA

El lenguaje de operación de una máquina y específicamente de la unidad central de proceso. En el QL el código máquina es el propio del microprocesador M68008, idéntico al M68000 excepto en el número de ciclos de reloj, que algunas instrucciones utilizan en su funcionamiento.

MATHS STACK
PILA MATEMATICA

El lugar en donde se almacenan los cálculos intermedios.

MICROPROCESSOR
MICROPROCESADOR

Un elemento electrónico con un conjunto de operaciones fijas. Se caracteriza por su velocidad, longitud de palabra interna, longitud de palabra externa y arquitectura. El M68008 posee un reloj de 7.5 MHz, palabra interna de 16/32 bits, bus de datos externo de 8 bits y arquitectura de 32 bits.

MOTOROLA
MOTOROLA

Fabricante estadounidense de la familia de microprocesadores M68xxx. Además fabrica una gran variedad de equipos para comunicaciones.

MULTITASKING
MULTITAREA

La ejecución simultánea de varios trabajos. SuperBasic no es capaz de efectuarla, aunque Supercharge produce códigos que si la admiten.

NAME
NOMBRE

Una secuencia de caracteres alfanuméricos y símbolos de subrayado (_). El primer carácter no debe ser numérico.

NATIVE CODE
CODIGO NATIVO

Programa en un formato adecuado para ser directamente ejecutado por la unidad central de proceso.

NESTING
ANIDAMIENTO

La inclusión de una estructura dentro de otra. Se suele aplicar este término a las estructuras de bucles o lazos.

OBVIOUSLY
OBVIAMENTE

No estamos en absoluto seguros de ello y no podemos probarlo de ninguna forma.

OPERATING SYSTEM
SISTEMA OPERATIVO

Todo el software que controla el funcionamiento de un ordenador. En el QL el sistema operativo se conoce con el nombre de QDOS.

OPERATION CODE
CODIGO DE OPERACION

Todas las instrucciones que llevan a cabo determinada operación en una cierta unidad central de proceso, y que determinan el repertorio de operaciones que pueden realizarse.

OPERATOR
OPERADOR

Una entidad, generalmente representado por un símbolo, que puede aplicarse a uno o más operadores (constantes, variables, funciones y expresiones). Se llama operador unitario a aquel que se aplica a un solo operando, (por ejemplo NCT). Se llama operador binario a aquel que se aplica a dos operadores, (por ejemplo "+").

OBJECT CODE
CODIGO OBJETO

El programa final de un compilador. En el caso de Supercharge el programa que puede ser ejecutado directamente por el microprocesador M68008.

OPTIMISATION
OPTIMIZACION

La producción del código objeto haciendo el mejor uso de las características del ordenador. Generalmente se aplica a la minimización del tiempo de ejecución, especialmente cuando las necesidades de memoria no son restrictivas. La optimización puede ser global o local. Supercharge está diseñado especialmente para lograr una excelente optimización local y una correcta optimización global. La optimización de tiempo y espacio puede realizarse mediante el uso de las sentencias REM- y REM+.

PARAMETER
PARAMETRO

Información que es pasada a un procedimiento, subrutina o función.

PARENTHESSES
PARENTESIS

Otra forma de denominar los símbolos "(" y ")"

PARSING
PARSING

El proceso de decidir si una secuencia de instrucciones son correctas bajo un punto de vista semántico y sintáctico.

PASS
PASADA

Una etapa en el proceso de compilación que conlleva una búsqueda a través de todo el programa Basic o de su correspondiente código intermedio. Toda compilación realiza normalmente varias de estas pasadas.

PRECEDENCE
JERARQUIA

El orden convencional o secuencia en la que han de ser evaluados los operadores dentro de una expresión.

PRIORITY
PRIORIDAD

Un valor numérico utilizado para señalar la cantidad de recursos que el sistema asigna a determinado trabajo.

PROCEDURE
PROCEDIMIENTO

Una sección de un programa identificada por un nombre, capaz de ser llamada desde cualquier lugar del programa (incluyendo el propio procedimiento), y que puede llevar a cabo operaciones.

PROGRAM
PROGRAMA

Un conjunto de instrucciones que pueden ser efectuadas por un ordenador. Así los programas SuperBasic son un conjunto de instrucciones que posibilitan que el QL lleve a cabo determinadas operaciones con objeto de conseguir unos determinados resultados.

QDOS
QDOS

Sistema operativo del QL. No es accesible a través del SuperBasic

RAM
RAM

Siglas de Random Access Memory o Memoria de Acceso Aleatorio. Es una memoria de lectura/escritura.

REGISTER
REGISTRO

Una entidad utilizada para almacenar información dentro de un ordenador.

RELOCATABLE
RELOCALIZABLE

Cuando se aplica a un programa, significa que este puede trabajar estando situado en cualquier lugar de la memoria. Supercharge produce programas relocalizables.

REMARK
OBSERVACION

Un comentario, insertado en un programa de un lenguaje de alto nivel, con objeto de hacerlo más comprensible. Supercharge los ignora, excepto los REM- y REM+.

ROM
ROM

Siglas de Read Only Memory o Memoria de Solo Lectura

RUN-TIME
TIEMPO DE EJECUCION

El tiempo durante el cual el programa es ejecutado

SEMANTICS
SEMANTICA

La parte de la definición de un lenguaje que concierne al significado o efecto de un texto. La semántica comprende el análisis de los símbolos tomados en el contexto y no de forma individual

SINCLAIR RESEARCH
INVESTIGACIONES SINCLAIR

Las gentes que hicieron posible lo imposible

**SLICE
PARTICION**

Una parte de una ordenación obtenida mediante el uso restrictivo de los índices. Esta partición es un subconjunto del conjunto principal.

**SOFTWARE
SOFTWARE**

Aquella parte de un ordenador que es intangible, como los programas o los bits.

**SOURCE CODE
CODIGO FUENTE**

El programa de entrada a un compilador para ser posteriormente traducido a código máquina. El Supercharge es código fuente es el programa escrito en SuperBasic

**STACK
PILA**

Una pila de datos, según la filosofía LIFO, (Last In First Out), último en entrar primero en salir, en donde todos los accesos, modificaciones e inserciones se efectúan desde la parte superior de la pila. Dentro de la memoria del QL existen varias pilas de este tipo.

**STATEMENT
SENTENCIA**

Cada una de las unidades en que está dividido un programa de alto nivel.

**STRING
CADENA**

Una secuencia ordenada de caracteres

**STRUCTURE
ESTRUCTURA**

Cuando se aplica a un programa, es sinónimo de diseño, enfatizando tanto sus componentes como los interrelaciones entre ellos. En programas perfectamente estructurados pueden dividirse en pequeñas partes perfectamente diferenciadas, pudiendo seguirse el orden y la filosofía del programa. SuperBasic es un lenguaje que permite una programación estructurada. Supercharge admite cualquier comando o sentencia estructurada que pueda ser construida por SuperBasic.

SUBROUTINE
SUBROUTINA

Una sección independiente de un programa que se ejecuta en el programa principal cuando se efectua una llamada a dicha sección. Una vez ejecutada regresa a la instrucción siguiente a la que provocó la llamada. En SuperBasic subrutina es sinónimo de procedimiento. Las subrutinas generalmente ahorran espacio y hacen el programa mas inteligible.

SUBSCRIPT
SUBSCRITO

Un medio para referenciar un elemento en una ordenación, mediante el uso del nombre de la ordenación y del subíndice adecuado.

SUBSTRING
SUBCADENA

Una cadena que está contenida exactamente dentro de la cadena principal.

SUPERBASIC
SUPERBASIC

El lenguaje de programación suministrado con el QL. Es una variación muy potente y estructurada del Basic.

SUPERCHARGE
SUPERCHARGE

Un producto de Digital Precisions del que puede Vd. considerarse ahora un orgulloso propietario.

SUPERCHARGED
SUPERCARGADO

Se aplica a los programas que han sido compilados con Supercharge.

SYNTAX
SINTASIS

El conjunto de leyes que gobiernan y definen la secuencia de instrucciones en un lenguaje de programación. Al contrario que la semántica, la sintaxis no interviene en el significado de los términos sino solamente en su forma.

**SYSTEM
SISTEMA**

Lo que se obtiene por la unión del software y el hardware.

**SYSTEM VARIABLE
VARIABLE DEL SISTEMA**

Un valor mantenido por el sistema en base a su estado de almacenamiento, valores por omisión y otros datos de uso por los recursos del sistema.

**TAG
MARCA**

Un identificador utilizado para discriminar entre variantes del mismo tipo

**TASK
TAREA**

Un conjunto de instrucciones de un programa que tienen entidad por ellas mismas. Supercharge permite la concurrencia ejecutándose al mismo tiempo.

**TASK IDENTIFIER
IDENTIFICADOR DE TAREA**

Dos números enteros, el primero de los cuales es un índice del número de tarea y el segundo es una marca. Ambos identifican una determinada tarea.

**THEREADED CODE
CODIGO ENLAZADO.**

Un código conteniendo una secuencia de puntos de entrada para las rutinas. Un salto incondicional se efectúa a una rutina cuya dirección se indica por una palabra en el propio código. Para finalizar la rutina se termina mediante otro salto incondicional a un nuevo punto de entrada indicado por la próxima palabra dentro del código.

**TOKENISATION
TOKENIZACION**

La conversión de un programa en forma de texto a forma binaria, en unidades llamadas "tokens".

**TRAP
TRAMPA**

Un estado del sistema, comandado por señales del microprocesador, por la cual una secuencia de instrucciones ha de ser abandonada y sustituida por una secuencia correcta. A menudo esta última secuencia cae dentro del sistema operativo. Los "traps" son utilizados para el manejo de errores.

VARIABLE
VARIABLE

Es valor almacenado en algún lugar del ordenador y capaz de ser modificado.
Las variables son referenciadas por sus nombres.

WORKSPACE
AREA DE TRABAJO

Un área de trabajo reservada para una determinada tarea, para leer un programa, escribir sobre él o manipularlo.