

TURBO

news

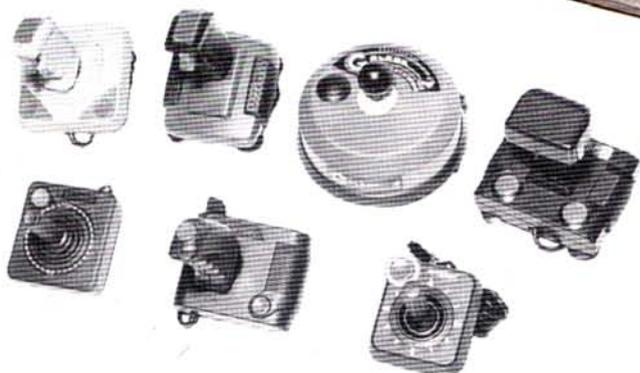
Revista para Computadores **ATARI** N° 10 - MAYO - JUNIO 1990

\$ 550



COMPUTACION

1 COMPUTADOR 65 XE
1 CASSETERA
2 JOYSTICKS
6 JUEGOS
\$ 67.500.

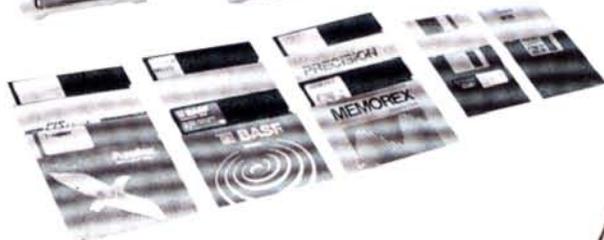


JOYSTICK

JOYSTICK GALAGA.....	\$ 3.750
BG - 201	
CON MICROSWITCH.....	\$ 6.040
BG - 747.....	\$ 3.000
BG - 105.....	\$ 2.500
BG - 124.....	\$ 1.500
QUICK SHOT I.....	\$ 2.200
QUICK SHOT II (AUTODISPARO).....	\$ 5.250

CAJAS PORTADISKETTES

3.1/2 40 UNIDADES.....	\$ 3.750
5.1/4 50 UNIDADES.....	\$ 3.650
5.1/4 100 UNIDADES.....	\$ 4.500



DISKETTES 5 1/5 2S-2D:

VERBATIM.....	\$ 390 c/u.
FUJI.....	\$ 355 c/u.
GOLDSTAR.....	\$ 225
PRECISION.....	\$ 225
MEMOREX.....	\$ 310 c/u.

DISKETTES 3,5"

FUJI.....	\$ 695
MEMOREX.....	\$ 780
PRECISION.....	\$ 715

PROGRAMAS EDUCATIVOS

SERIE TM (TELEMATICA)
DESDE \$ 1.790

CASSETTES EDUCATIVOS
TURBO SOFTWARE DESDE \$ 895



ELECTRONICA
CASA ROYAL LTDA.

PRIMER CENTRO ELECTRONICO CHILENO

AV. L. B. O'HIGGINS 845

CENTRAL: 6322313

FAX: 399047 - TELEX 340517

MONJITAS 813

DESPATCHOS A PROVINCIA PREVIO ENVIO DE CHEQUE, VALE VISTA O GIRO TELEGRAFICO A CORREO 21 - CASILLA 395 - V - STGO.

EDITORIAL

Aquí estamos nuevamente reencontrándonos con ustedes como todos los meses. En este número queremos unirnos a este gran acontecimiento que es el Campeonato Mundial de Fútbol Italia 1990 y por eso les entregamos un programa con el cual podrán ir llevando las estadísticas de este trascendental acontecimiento.

El curso de Player-Missile llega a su fin y por eso queremos darles un programa a modo de resumen de todo lo aprendido hasta el momento, así como una guía de referencia rápida del uso de

dichos gráficos.

En el curso de protección de software, les damos un programa que les permitirá engañar a los eventuales transgresores.

En el curso de Assembler continuamos conociendo el computador a través de un programa ejemplo.

Finalmente, como lo hacemos mes a mes, les entregamos el mapa de memoria, el ranking de los juegos, su descripción muchas otras novedades que a continuación pasamos a descubrir.

CONTENIDO

2 BASIC
(Lección 10)

4 BOOT FILES

7 MAPA DE MEMORIA

10 TECNICAS DE PROTECCION

14 GRAFICOS POR COMPUTADORA
(Quinta parte)

18 ASSEMBLER
(Lección 10)

24 RANKING DEL MES
DESCRIPCION DE JUEGOS

26 PROGRAMAS

TURBO

news

Circulación Mensual, Nacional e Internacional.
Destinada a los usuarios de computadores ATARI (R) como material didáctico de Programación. TURBO news (R) es una publicación de EDITORA TURBO LTDA.
Domicilio: Av. Fco. Bilbao 4226 - Teléfono: 486506.

DIRECTOR RESPONSABLE: Mauro Pieressa. REPRESENTANTE LEGAL: Marcelo Waldbaum. PRODUCCION: Marcelo Waldbaum y Mauro Pieressa. Programadores y Diseñadores de Computación. DIRECTORA DE ARTE: Odali Guerrero L. CORRECTOR: Marcial Valenzuela S. PUBLICIDAD Y RR.PP.: Liliana Muñoz Otárola, Hernán Vittini. COLABORACION: Mariana Pizarro. PUZZLE: Mario Calvo A. FOTOCOMPOSICION: Brubytes. IMPRESION: Servigraf. DISTRIBUCION: Alfa Ltda. Agradecemos la colaboración de COELSA S.A. Centro Atari. (Augusto Leguía Sur 75). Atari es marca registrada de ATARI CORPORATION. TURBO news es marca registrada de EDITORA TURBO LIMITADA. (Registro de Marca N° 342428 9-05-89).

LECCION 10

En este número, aprovechando la ocasión del Mundial de Fútbol, haremos un breve resumen de las instrucciones vistas hasta el momento, volcadas en un programa para ir llevando las estadísticas de este gran evento deportivo.

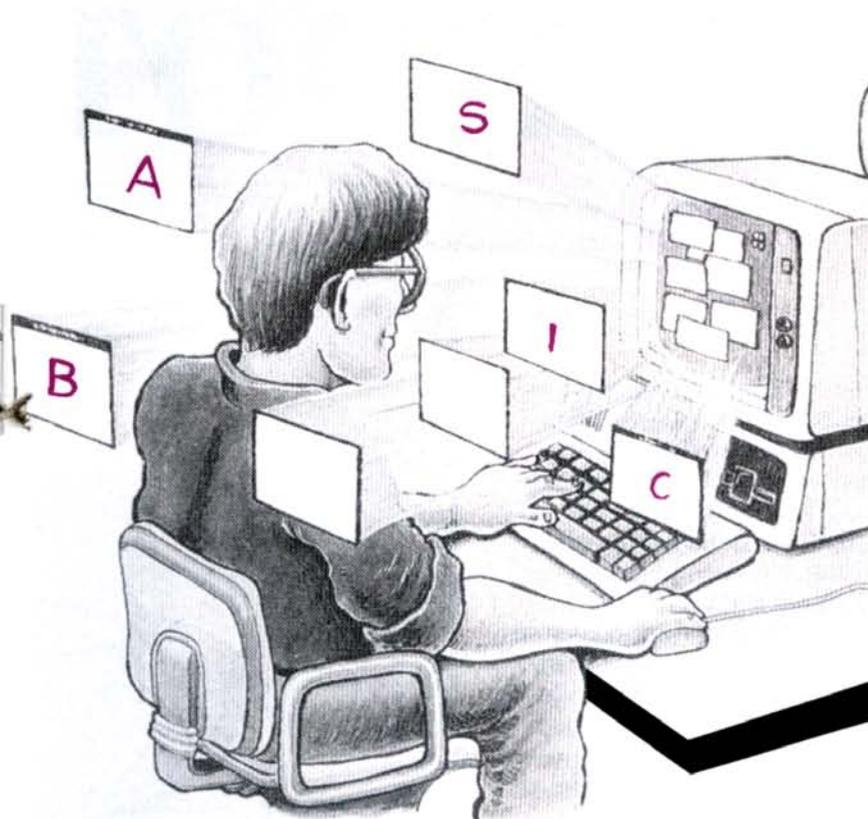
UN H S A B

Antes de entrar de lleno al programa, objetivo principal de este artículo, veremos un par de instrucciones que te serán muy útiles, no sólo para este programa, sino también para cualquiera que desees realizar. La primera de ellas es la instrucción DATA.

INSTRUCCION DATA

Hemos visto que podíamos guardar datos, tanto en diskettes como en cassettes. Sin embargo, hay ocasiones en que los datos para un programa son siempre los mismos, entonces en lugar de almacenarlos en alguno de estos medios, conviene directamente hacerlos formar parte del programa. Hasta el momento, si tuviéramos que hacer eso, lo solucionaríamos a través de asignaciones. Por ejemplo si queremos que el vector edad tenga los datos 23, 22, 21, 22, 24, deberíamos hacer:

```
EDAD(1)=23  
EDAD(2)=22  
EDAD(3)=21  
EDAD(4)=22  
EDAD(5)=24
```



Esto puede ser facilitado a través de la instrucción DATA que se utiliza junto con la instrucción READ. Esta última lo que hace es leer la información que se encuentra en las instrucciones DATA. Veamos cómo se resolvería el mismo ejemplo.

```
10 FOR I=1 TO 5:READ
A:EDAD(I)=A:NEXT I
20 DATA 23,22,21,22,24
```

También ayuda con el uso de estas dos instrucciones, el comando RESTORE. Este último permite que en el caso de existir en un programa varias

instrucciones DATA, podamos elegir a partir de cuál de todas ellas deseamos comenzar a leer. Esta instrucción no es obligatoria que aparezca. De no estar presente cuando el computador encuentre una instrucción READ va a comenzar a leer desde el primer DATA. Por ejemplo:

```
10 READ A:PRINT A
20 RESTORE 50
30 READ A:PRINT A
40 DATA 1,2,3,4
50 DATA 5,6,7,8
```

Va a imprimir 1 y 5. Si borramos la instrucción 20, va a imprimir 1 y 2.

A continuación vamos a ver otras instrucciones sumamente útiles. El ON var GOTO lista y ON var GOSUB lista.

Se utilizan de la siguiente manera: Dependiendo del valor que tome la variable var, va a ir al número de instrucción correspondiente de la lista. Por ejemplo:

```
ON I GOTO 100,200,300
ON I GOSUB 100,200,300
```

Si I vale 1, el programa va a ir a la instrucción 100, si vale 2 a la 200 y así sucesivamente. En este caso, si vale más de 3, sigue con la instrucción que venga a continuación. La diferencia entre que sea GOTO y GOSUB, la hemos visto en números anteriores.

Un ejemplo muy útil de aplicación para estas instrucciones, es cuando tenemos un menú en un programa y debemos realizar distintas acciones según la opción elegida. Por ejemplo, en los programas entregados en números anteriores de archivos, teníamos que seleccionar si deseábamos dar altas, modificaciones, o inicializar el diskette. Para ello debíamos hacer 3 instrucciones IF, para saber qué opción se había escogido. Exactamente lo mismo podemos realizarlo con una sola instrucción ON...GOTO.

Finalmente veremos dos instrucciones más: ASC y CHR\$. La primera de ellas se utiliza para que dado un caracter cualquiera nos diga cuál es el



BASIC

código interno que utiliza el computador para representarlo. Dicho código varía de 0 a 255 y se denomina ATASCII. La función CHR\$ funciona al revés. Dado un número de 0 a 255, el computador nos muestra qué carácter representa.

La función CHR\$ la vamos a utilizar para clarificar nuestros listados. Por ejemplo para limpiar la pantalla siempre hacíamos "PRINT" y una flecha apuntando a la esquina superior izquierda, que la lográbamos oprimiendo las teclas ESC, CONTROL y CLEAR. Ahora vamos hacer lo mismo con la instrucción PRINT CHR\$(125), ya que 125 es el código ATASCII correspondiente a ese gráfico. Prueba hacer PRINT CHR\$(125) en modo directo y verás su resultado.

El programa que damos al final tiene varias opciones. Primero puedes ir ingresando los resultados. Verás que si ingresas un equipo que no existe, Prasil por ejemplo, el computador te pedirá que lo corrijas. Una vez ingresados los resultados, podrás solicitarle la tabla de posiciones de cada uno de los grupos, la general, la de goles a favor y la de goles en contra. Tú puedes agregarle que dichas tablas las imprima en papel y que almacene los resultados en disco o cassette.

Analiza cuidadosamente el programa y cualquier duda no temas en escribirnos, que ésta será solucionada. ¡¡¡Hasta pronto!!!

En la Edición anterior de nuestra Revista Turbo News, hemos desarrollado la estructura de los Files Binarios.

Otra de las formas de cargar un programa en el computador, es utilizar el Formato Boot.

Cuando el computador se enciende, una vez que realiza todas las inicializaciones de las posiciones de memoria que afectan a los distintos Chips del Hardware testea si existe una carga de algún cassette o diskette con el formato Boot.

Si al encender el computador, la tecla Start está presionada sola o conjuntamente con la tecla Option, se intenta una carga de archivos desde la grabadora. Pero si sólo está presionada la tecla Option o bien ninguna de estas teclas está siendo presionada se intenta, si la diskettera está en línea, una carga directa desde el diskette en el formato Boot.

Un archivo Boot de disco está constituido por una sucesión de bytes que se encuentran almacenados consecutivamente en el diskette desde el sector 1.

Su estructura comienza con un encabezado de 6 bytes que le indican al Sistema Operativo en qué zona de la memoria se

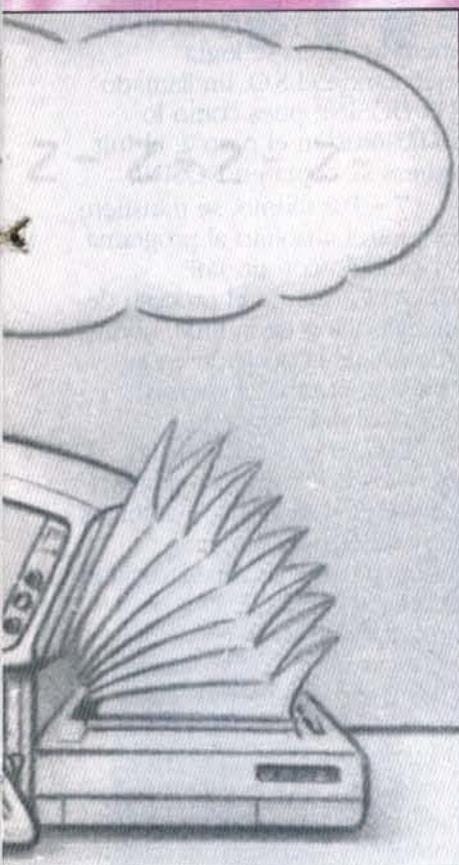
BOOT



va a ubicar el programa y cuántos sectores se van a leer. Para tener claro cuál es la función de cada uno de estos bytes los describiremos detalladamente:

- Byte 1 Cero
- Byte 2 Cantidad de sectores a leer
- Byte 3 LO-BYTE del Load Address
- Byte 4 HI-BYTE del Load Address
- Byte 5 LO-BYTE del Init Address
- Byte 6 HI-BYTE del Init Address

FILES:



El primer byte del primer sector generalmente no es utilizado para la carga de programas Boot y por lo general contiene un cero.

El segundo byte contiene la cantidad de sectores de 128 bytes cada uno que se leerán para la carga del archivo Boot.

El tercer y cuarto byte, le indican al Sistema Operativo del Computador, cuál es la primera posición de memoria en donde se comenzarán a cargar los bytes del archivo y el quinto y sexto byte le indican cuál es la posición de memoria a la cual se le transferirá el control una vez terminado el proceso de carga.

Cuando el proceso de carga se inicia, el computador realiza

los siguientes pasos:

1 - Lee el primer sector del disco y lo almacena en las posiciones de memoria consecutivas de \$400.

2 - Se procesan los primeros 6 bytes del sector, para lo cual:

- el primer byte se almacena en DFLAGS o \$240.

- el segundo byte se almacena en DBSECT o \$241.

- el Load Address se almacena en BOOTAD o \$242.

- el Init Address se almacena en DOSINI o \$C.

manquehue

CENTRO ATARI
Cursos 8 horas por la compra de su
Computador ATARI

- Impresoras • Juegos
- Disketteras • Educativos E. Básica
- Cassetteras • Educativos E. Media

A. VARAS 651 • FONOS: 255043 - 255450 • PUERTO MONTT



BOOT FILES

3 - Luego de procesar esta información, se almacenan los restantes 122 bytes leídos del primer sector del disco en las correspondientes direcciones según el Load Adress.

4 - Se leen los restantes sectores del disco directamente sobre las posiciones adecuadas según el Load Adress.

5 - Una vez cargado todo el programa en la memoria se ejecuta el proceso de inicialización. Para esto el Sistema Operativo, realiza un llamado a la Subrutina que comienza desde el séptimo byte de la carga, es decir, ejecuta un JSR LOAD ADDRESS+6. Este proceso de inicialización tiene como objetivo definir los valores de los punteros que administran las zonas de memoria libres en nuestro Atari.

MEMLO [\$2E7, \$2E8], una vez cargado todo el programa, debe apuntar a la primera posición disponible de la memoria RAM para que el Sistema Operativo proteja la zona del programa ya cargado. Para esto, en esta subrutina se debe ejecutar una asignación tanto a MEMLO como a APPMHI [\$0E, \$0F] definiendo a estos dos punteros direccionados a la primera posición fuera del programa cargado. Es decir, que si el programa finaliza con un rótulo END debemos ejecutar en esta subrutina las siguientes instrucciones:

```
LDA #<END
STA MEMLO
STA APPMHI
LDA #>END
STA MEMLO+1
STA APPMHI+1
```

Otra dirección que hay que modificar en la inicialización de la RAM es DOSVEC [\$0A, \$0B]. DOSVEC debe apuntar antes de terminar esta inicialización de la RAM a la posición dentro del programa en donde se encuentra el cuerpo principal del programa. Supongamos que el cuerpo se inicia en el rótulo INICIO, debemos ingresar el código:

```
LDA #<INICIO
STA DOSVEC
LDA #>INICIO
STA DOSVEC+1
```

Por último debemos ejecutar el regreso de esta subrutina pero dejando en cero el indicador de acarreo, pues



sino el Sistema Operativo asume que existió un Boot Error. Para esto ingresamos el código:

```
CLC
RTS
```

6 - Luego de inicializar la RAM, se ejecuta la inicialización del Software ya cargado en la memoria. Esto se logra ejecutando el S.O. un llamado JSR DOSINI, pues como lo indicamos en el paso 2, el Init Adress se copia en DOSINI.

7 - Por último, se transfiere el control absoluto al programa ya cargado con un JMP DOSVEC, que en el proceso de inicialización de la RAM, obtuvo el valor de la posición en la cual comenzaba el cuerpo principal del programa.

Esta es la estructura de archivos Boot. Parece compleja pero en la práctica son archivos muy útiles pues cargan automáticamente, sin necesidad de cargar ningún D.O.S. o programas cargadores. Para clarificar un poco la idea de los Files Boot, definiremos en la práctica la generación de un programa Boot (pág. 27)

Este programa está listo para compilarse y almacenarse consecutivamente en el diskette a partir del byte 1 del sector 1 y al encender el Computador, automáticamente correrá el programa cuyo cuerpo principal se encuentra a partir del rótulo PROGPRINCIPAL.

Esta estructura de archivos sólo es aplicable para los programas desarrollados en Lenguaje Assembler. Para que un programa Basic corra al encenderse el Atari debemos incorporar en el disco un programa especial que en las siguientes ediciones de tu Revista TURBO news te explicaremos cómo hacerlo.

Mapa de memoria

En esta Edición de tu Revista TURBO news, desarrollaremos las posiciones de memoria comprendidas entre 708 y 732.



CASA HUEPE

EL ROBLE 638 • FONOS: 221338 - 221536
CASILLA 238 • CHILLAN

- CASSETTES DE JUEGOS
- CASSETTES EDUCATIVOS
- LAPIZ DE LUZ

PARA SU ATARI

Mapa de memoria

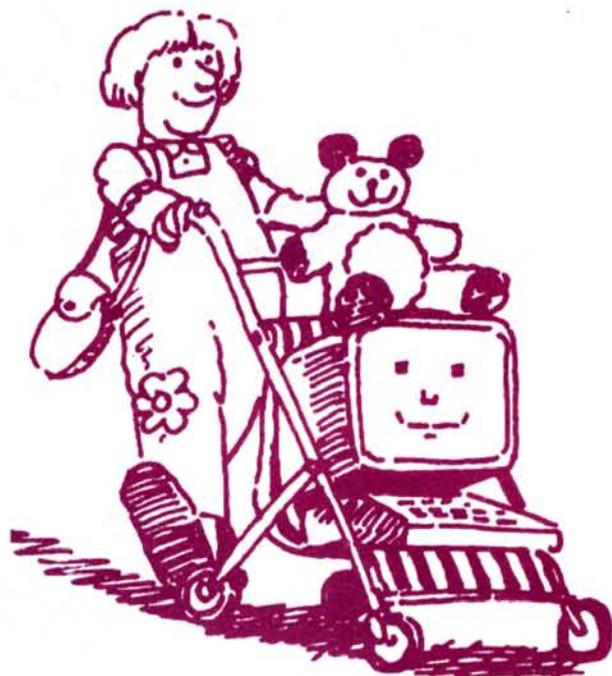
El Atari utiliza un método muy flexible para la generación de colores en la pantalla. En vez de almacenar el color de cada punto de la pantalla en la memoria de ésta, Atari se refiere a un registro de color. Cada ptxel de la pantalla, contiene el número del registro de color que lo identifica. Es decir que si deseamos cambiar el color del gráfico desplegado con el registro de color 1, lo único que deberemos realizar es almacenar en dicho registro el valor del color a utilizar.

Existen 5 registros de color, numerados del 0 al 4. El registro 4 es conocido como el registro de BACKGROUND porque a él se refieren todas las zonas de la pantalla en las cuales no hay nada graficado.

Los registros de color ocupan cada uno un byte de memoria, de los cuales los cuatro bits de alto orden definen el color de 0 a 15 y los cuatro de bajo orden el brillo del mismo. Pero de estos cuatro bits, sólo se ocupan 3 que definen 8 niveles de brillo. Por lo tanto, con los 16 colores cada uno de los cuales con 8 niveles de intensidad se pueden obtener 128 colores.

REGISTRO DE COLOR:

BIT	7 6 5 4 /	3 2 1 /	0
	COLOR /	BRILLO /	SIN USO



704 \$2C0 PCOLR0: Color del player/missile número 0

705 \$2C1 PCOLR1: Color del player/missile número 1

706 \$2C2 PCOLR2: Color del player/missile número 2

707 \$2C3 PCOLR3: Color del player/missile número 3

708 \$2C4 COLOR0: Este byte representa al registro de color cero y es la copia del registro de Hardware ubicado en la posición de memoria \$D016 ó 53270.

709 \$2C5 COLOR1: Representa al registro de color 1 cuyo registro de Hardware se encuentra en la posición \$D017 ó 53271.

710 \$2C6 COLOR2: Al igual que el byte anterior, representa al registro de color número 2 cuyo registro de Hardware se encuentra en \$D018 ó 53272.

711 \$2C7 COLOR3: Representa al registro de color 3 y su posición en el Hardware es \$D019 ó 53273.

712 \$2C8 COLOR4:

Representa al registro de color número 4 y como lo indicamos anteriormente es el background de la pantalla. Es un shadow del registro de Hardware ubicado en \$D01A ó 53274.

Cuando nos referimos a que determinada posición es un Shadow de otra, en realidad estamos indicando que es copia de la anterior. Esto es así, pues el Sistema Operativo de nuestro Atari, posee determinadas posiciones de memoria que son utilizadas para comunicarnos directamente con el Hardware y que son de sólo escritura o lectura. Por este motivo el S.O. trabaja con posiciones de memoria, copia de las anteriores que siempre tienen el mismo valor pero que pueden leerse y escribirse y es el mismo Sistema Operativo que las copia desde el Shadow al registro de Hardware cada

interrupción de Vertical Blank.

El Basic de nuestro Atari utiliza la instrucción SETCOLOR para definir el color que se almacenará en el registro de color. Pero conociendo las direcciones de los registros de color, el mismo trabajo se puede realizar con un POKE:

SETCOLOR 1,2,7 = POKE
709,(2*16)+7

Cuando encendemos el computador, los registros de color se definen con los siguientes valores:

REGISTRO	COLOR
708	40
709	202
710	148
711	70
712	0

729 \$2D9 KRPDEL: Cuando mantenemos presionada una misma tecla del teclado durante unos instantes, ésta comienza a repetirse en la pantalla. Esta posición de memoria maneja el tiempo que debe ser presionada antes de comenzar a repetirse. Pero este valor puede modificarse pokeando cualquier valor que va a indicar la cantidad de 1/60 segundos que el computador va a esperar antes de comenzar la repetición.

730 \$2DA KEYREP: Esta posición contiene la velocidad de repetición de las teclas al mantenerse oprimidas. Si en esta posición almacenamos un 0 le estamos indicando al S.O. que no repita teclas, es decir, que al presionar una tecla sólo se imprime un caracter, pero

cualquier valor puede ser almacenado en esta posición de memoria.

731 \$2DB NOCLIK: Si en esta posición almacenamos cualquier valor distinto de cero, el teclado deja de hacer su ruido característico cada vez que presionamos una tecla.

732 \$2DC HELPPFG: Esta posición de la memoria indica si la tecla HELP está o no siendo presionada. Si contiene un 17, la tecla HELP ha sido presionada sola, un 81 indica que fue presionada junto a la tecla SHIFT y un 145 con la tecla CONTROL.



CompuCenter

COLECCIONA TRES
CARATULAS DE CASSETTES
TURBO SOFTWARE CON EL
AUTOADHESIVO
PROMOCIONAL Y CON \$ 200
PODRAS OBTENER EN
NUESTRO LOCAL UN
CASSETTE DE JUEGOS
TURBO SOFTWARE.

- ATARI
- COMMODORE
- APPLE
- Equipos
- Suministros
- Software
- Materiales didácticos
- Programa IBM - MACINTOSH

**SUPER
OFERTA**

ATENCION TODOS LOS DIAS DEL AÑO
P. ARAUCO • LOCAL 247 A • FONO: 2420596



TECNICAS DE PROTECCION

En este número, continuamos con las técnicas de protección de Software. Veremos cómo evitar las copias realizadas desde el DOS 2.5 que es el que mayor difusión ha tenido.

Los DOS (Sistemas Operativos de Disco), poseen en su menú principal 2 opciones de copias de archivos. Son la opción O de Duplicate File y la C de Copy File. Las diferencias entre ellas tienen que ver con el uso de una o dos disketteras. Mediante ellas se pueden realizar copias de cualquiera de los archivos almacenados en disco. La técnica de protección consiste en "esconder" el directorio del diskette para que no se pueda acceder a sus archivos y poner en su lugar uno falso, que confunda a los transgresores. Cabe destacar que esta técnica no evita que se pueda copiar el disco entero.

Para entender el tema daremos primero una breve introducción de cómo maneja los archivos el computador ATARI. Para mayor información, rogamos referirse al artículo



correspondiente aparecido en números anteriores de nuestra revista.

El directorio del disco es la parte más ocupada de éste. Cada vez que se quiere leer, grabar, modificar algún archivo, la diskettera lo consulta antes de hacer nada. Se encuentra física y aproximadamente en el centro del diskette, para que su acceso sea lo más rápido posible. Esto es en los sectores 361 a 368. También existe un sector, el 360, denominado VIOC, que contiene la información de cuáles sectores se encuentran desocupados y cuáles no.

La técnica de esconder el directorio consta de siete pasos.

- 1 - Hacer una copia desprotegida del disco.
- 2 - Hacer que nuestro programa corra automáticamente al encender el computador (autorun).
- 3 - Buscar un espacio libre en el disco donde ubicar el nuevo directorio.
- 4 - Copiar el directorio a la nueva ubicación.
- 5 - Hacer que el DOS conozca la nueva ubicación.
- 6 - Poner un directorio disfrazado en su lugar.
- 7 - Borrar el DUPSYS.



PASO 1

Como el disco una vez protegido no va a ser modificable, conviene tener una copia por cualquier alteración que queramos hacerle en el futuro.

PASO 2

Cuando cargamos el diskette con el directorio modificado, si no se corre el programa automáticamente y sin que se le pueda detener (con BREAK, RESET, etc.), corremos el riesgo de que puedan averiguar con un simple PEEK, dónde está el directorio. Por ello debemos hacer que el programa sea un AUTORUN y protegerlo de que

pueda ser parado durante su funcionamiento. Para lo primero el DOS 2.5 tiene incorporado en el programa SETUP.COM, la opción de que un programa Basic sea un AUTORUN. Consulte su manual por cualquier duda. Para hacer que no se pueda detener, debe consultar el primer capítulo de este artículo descrito en el número anterior.

PASO 3

Debe buscar un lugar de ocho sectores libres consecutivos libres, para poner allí el directorio. El programa que va incluido en la revista lo hace automáticamente en la opción 1 del menú. Debe tener en cuenta que por una limitación que existe en la fórmula correspondiente al PASO 5, dichos sectores libres deben encontrarse entre los sectores 256 y 511.

PASO 4

Consiste en copiar los ocho sectores a la nueva posición. El programa que incluimos, también lo hace a través de la opción 2 del menú.

TECNOLOGIA DE AVANZADA

CASA MATRIZ

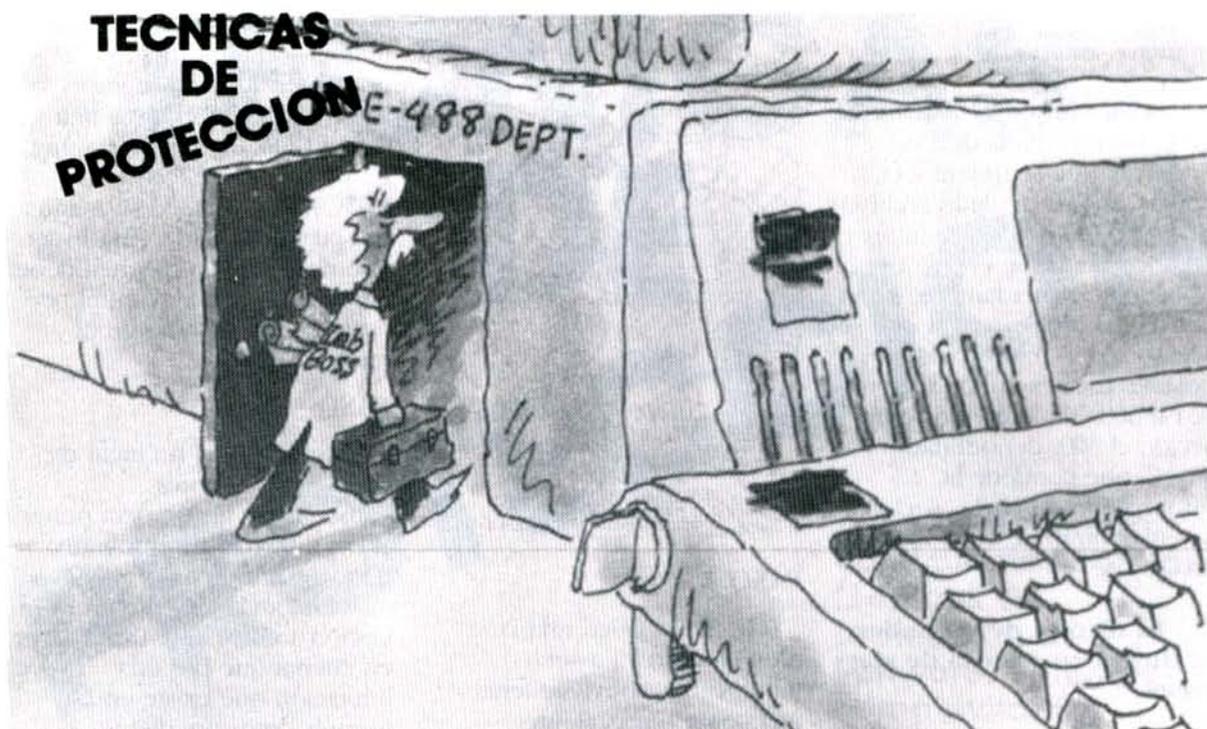
• SAN PABLO 1055 • FONOS: 6964285 - 714223

SUCURSALES

- EST. U. DE CHILE DEL METRO • Loc. 17 • FONONO: 724790
- PROVIDENCIA 2099 • FONONO: 2319883
- APOQUINDO 6029 • LOCAL 20 • FONONO: 2127179
- BARROS ARANA 435 - 439 • FONONO: 224 907 CONCEPCION
- VICUÑA MACKENNA 820

CASA MUSA

AHORA LES
OFRECE
CASSETTES
DE JUEGOS
ATARI



TECNICAS DE PROTECCION

PASO 5

Para que el DOS conozca la nueva posición del directorio, basta con pokear en la posición de memoria 4171 el resultado de la siguiente fórmula:

VALOR A POKEAR=105+(PRIMER SECTOR DEL DIRECTORIO ESCONDIDO-361)

Por ejemplo si queremos esconder el directorio a partir del sector 400 debemos hacer:

**105 +(400-361)=144
POKE 4171,144**

Como el valor a pokear va de 0 a 255, de ahí la limitación de dónde esconder el directorio de los sectores 256 a 511. Este POKE también lo realiza automáticamente el programa en la opción 2 del menú. A continuación, debemos grabar el disco con el nuevo DOS con la posición de memoria modificada. Para ello debemos ir al DOS y con la opción H grabarlo en el disco a

proteger.

PASO 6

A continuación viene la parte más entretenida, que consiste en poner un directorio falso en el disco. Aquí hay varias alternativas. Número 1, podemos poner el directorio de un disco en blanco. Número 2, poner el directorio de un disco cualquiera. Número 3, teniendo un editor de sectores, poner allí cualquier mensaje intimidatorio.

Los primeros dos casos, se realizan con el programa que adjuntamos en su opción tres, que permite copiar el directorio de cualquier otro disco, ya sea en blanco o no.

PASO 7

Hasta ahora hemos logrado lo siguiente: si encendemos el computador con el diskette modificado, vamos a poder ver ese directorio, pero no el de los demás discos, porque el DOS va a buscarlo donde está

el nuestro.

Lo mismo sucede si encendemos el computador con un DOS cualquiera. Cuando queramos ver el disco nuestro, vamos a ver el directorio disfrazado.

Para que el directorio real no sea visto nunca por nadie, lo que hay que hacer es borrar el DUP.SYS, que es un programa de apoyo del DOS.SYS. Es el que permite que veamos el menú con todas las opciones del DOS (ver directorio, copiar, borrar, etc.).

Una vez realizado todo esto, el disco va a funcionar perfectamente, pero no se va a poder ver su directorio y por ende tomar cada uno de sus archivos. Claro que con cualquier copiador de sectores, copiando el disco tal cual, sin ver su directorio puede ser copiado. En el próximo número veremos cómo evitar esto también.

PROMOCION

turbo

INFORNA



cia. de informática nacional sa.

**MEDIOS MAGNETICOS
ACCESORIOS DE COMPUTACION
DISTRIBUIDORES IBM-3M**

- CINTAS MAGNETICAS
- CARTUCHOS PARA IBM 3480
- DISKETTES DE 3,5" - 5,25" y 8"
- CINTAS IMPRESORAS
- IMPRESORAS
- MUEBLES DE COMPUTACION
- TERMINALES COMPATIBLES
- ACCESORIOS
- LINEA IBM
- LINEA 3M
- OTROS

SOFTWARE ATARI TURBO

PEDRO DE VALDIVIA NORTE 0119
HUERFANOS 1052 LOC. 27

L'Radio

CASA ELECTRONICA

- Toda la línea TURBO SOFTWARE para su ATARI
- Amplio stock en equipos de sonido, amplificadores, parlantes, micrófonos, etc.
- Repuestos Radio - TV

SAN ANTONIO 32
FONOS: 338010 - 393172
SANTIAGO - CHILE



**Electrónica
"SAN AGUSTIN"**

SAN ANTONIO 160
LOCAL 60 - TEL.: 336006

- JUEGOS Y EDUCATIVOS ATARI
- JOYSTICK
- DISKETTES DE JUEGOS ATARI
- ELECTRONICA - SONIDO
- COMPONENTES
- KITS PARA ARMAR
- AMPLIFICADORES
- PARLANTES - AMPLIFICADORES
- LUCES SECUENCIALES Y SICODELICAS
- BOCINAS - MEGAFONOS

IMPORTACION DIRECTA

INSIS

INGENIERIA DE SISTEMAS



la atención
especializada
para su computador
ATARI

VALPARAISO 363 LOC. 108
GALERIA FONTANA
FONOS: 901723 - 662199
VIÑA DEL MAR

Gatto's

**DISCOS
REGALOS
VIDEO**

AHORA CON SOFTWARE ATARI

O'HIGGINS 680 Local 6 y 14
CONCEPCION

**Señor
Avisador**

**Este espacio
está reservado
para usted**

**SOLICITE REPRESENTANTE DE
VENTAS AL
TELEFONO: 486506**

Gráficos por computadora

QUINTA PARTE

En el número pasado, terminamos de entregar todas las armas para poder usar los Players-Missiles. En este número les vamos a entregar una guía de referencia rápida, más un programa ejemplo de utilización de los Missiles.

Dado que la cantidad de posiciones de memoria que hay que conocer para poder hacer un programa que utilice Players-Missiles es mucha, en este número les entregamos una guía de referencia rápida, para evitar tener que consultar todas las revistas cada vez que quieras programar algo. Cuando una posición te provoque dudas, no temas en consultar la revista correspondiente donde encontrarás más detalles de ésta.

Comenzaremos con las posiciones de inicialización del programa:

Posición	Contenido
54279	Número de página de memoria donde comienza RAM de Player Missile. El valor se toma del PEEK(106)- la cantidad de páginas que deseemos reservar.
559	Aviso para Antic de habilitación de PM y resolución de los mismos. El valor se toma de la siguiente tabla:

NUMERO BIT	5	4	3	2	1	0
VALOR BIT	32	16	8	4	2	1

HABILITACION PANTALLA	1
RESOLUCION PM	1
HABILITACION PLAYER	1
HABILITACION MISSILE	1

El resto de los bits no son utilizados. Para saber qué valor pokear, se deben sumar los valores de los bits que queramos que estén en uno, más dos.

Posición	Contenido
53277	Aviso para GTIA de habilitación de PM. Un dos significa sólo Player y un tres Player y Missile.

Luego se debe hacer que coincida la zona de RAM de los PM con la de las variables. Se hace a través de las siguientes posiciones de memoria:

Posición	Contenido
134,135	Comienzo de la tabla de valores de las variables.
140,141	Comienzo de la zona donde está realmente el contenido de las variables

Para el desplazamiento horizontal del PM, tenemos la siguiente tabla:

POSICION	DESCRIPCION
53248	Posición Player 0
53249	Posición Player 1
53250	Posición Player 2
53251	Posición Player 3
53252	Posición Missile 0
53253	Posición Missile 1
53254	Posición Missile 2
53255	Posición Missile 3

Recuerda que los valores de 20 a 60, dependiendo del TV, y de 200 a 245, quedarán fuera de la pantalla.

Para el color del PM, existe la siguiente tabla:

POSICION	DESCRIPCION
704	COLOR PM 0
705	COLOR PM 1
706	COLOR PM 2
707	COLOR PM 3

Para el control de la anchura tenemos la siguiente:

POSICION	DESCRIPCION
53256	TAMAÑO PLAYER 0
53257	TAMAÑO PLAYER 1
53258	TAMAÑO PLAYER 2
53259	TAMAÑO PLAYER 3
53260	TAMAÑO TODOS LOS MISSILES



Un 0 o un 2 significa anchura simple. Un 1, anchura doble y un 3 anchura cuádruple.

La posición 623 se utiliza para el control de la prioridad. Puede tomar los siguientes valores

VALOR	PRIORIDAD
1	P0 P1 P2 P3 TODOS PLAYFIELDS
2	P0 P1 TODOS PLAYFIELDS P2 P3
4	TODOS PLAYFIELDS P0 P1 P2 P3
8	PF0 PF1 P0 P1 P2 P3 PF2 PF3



multiCentro

OFICINAS GENERALES:
5 ORIENTE N° 1042 • FONONO: 232549 • TALCA

TODOS PARA SU ATARI

TALCA : 1 SUR 1320
LINARES: INDEPENDENCIA 625
CONSTITUCION: FREIRE 676



Gráficos por computadora

Para la detección de colisiones tenemos la siguiente tabla:

POSICION	DESCRIPCIÓN
53252	PLAYER 0 CON PLAYFIELD
53260	PLAYER 0 CON PLAYER
53253	PLAYER 1 CON PLAYFIELD
53261	PLAYER 1 CON PLAYER
53254	PLAYER 2 CON PLAYFIELD
53262	PLAYER 2 CON PLAYER
53255	PLAYER 3 CON PLAYFIELD
53263	PLAYER 3 CON PLAYER
53248	MISSILE 0 CON PLAYFIELD
53256	MISSILE 0 CON PLAYER
53249	MISSILE 1 CON PLAYFIELD
53257	MISSILE 1 CON PLAYER
53250	MISSILE 2 CON PLAYFIELD
53258	MISSILE 2 CON PLAYER
53251	MISSILE 3 CON PLAYFIELD
53259	MISSILE 3 CON PLAYER

Los valores que puede tomar son los siguientes:

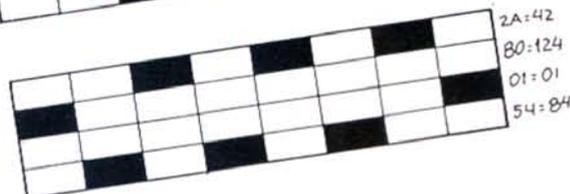
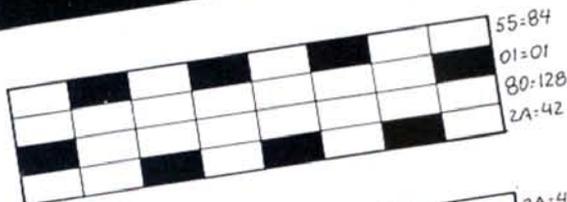
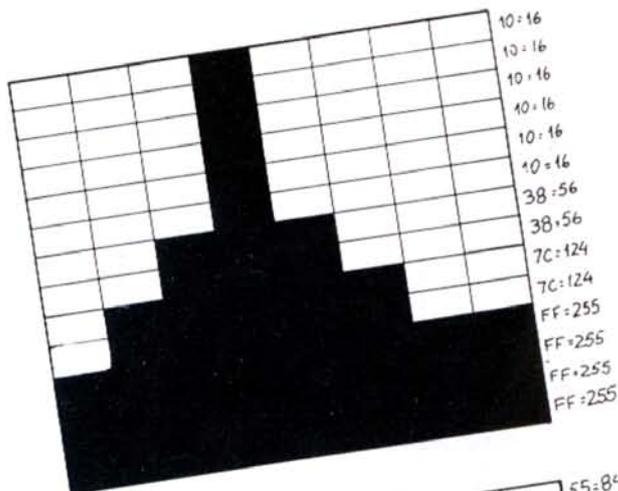
NUMERO DE BIT	3	2	1	0
VALOR DE BIT	8	4	2	1
PoPF0				1
PoPF1			1	
PoPF2		1		
PoPF3	1			

El valor final sale igual que en el caso del 559.

La posición 53278 se utiliza para el borrado de los registros de colisión. Esto se logra pokeándole cualquier valor.

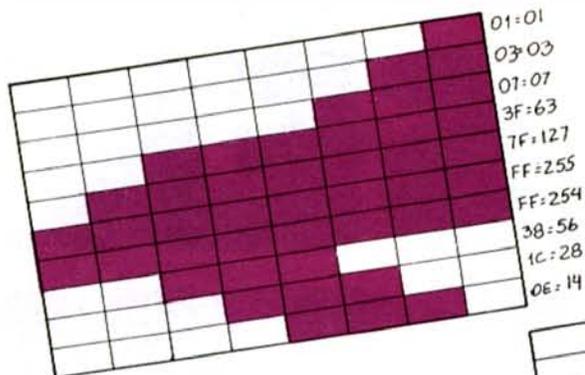
Al final les entregamos un programa con el cual

podrán practicar el uso de los misiles y la transformación de los PLAYERS. Pongan atención al efecto del movimiento de la rueda y la evolución de la explosión. Ustedes pueden agregarle más sonido y más pasos a dicha explosión. También pueden intentar agregarle un tanteador y un número limitado de disparos. En el próximo número continuaremos con la parte gráfica del computador, viendo cómo hacer los dibujos de fondo o Playfields.

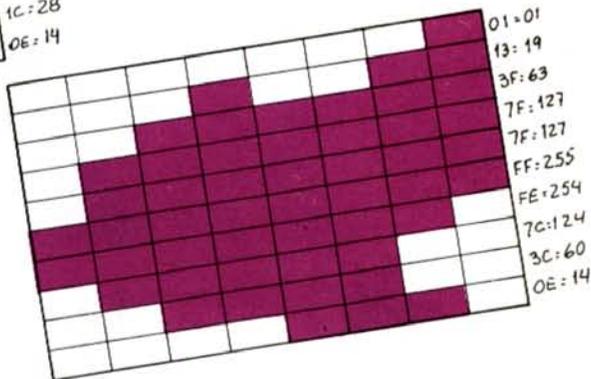


SOLUCION PUZZLE (Edición N° 9)

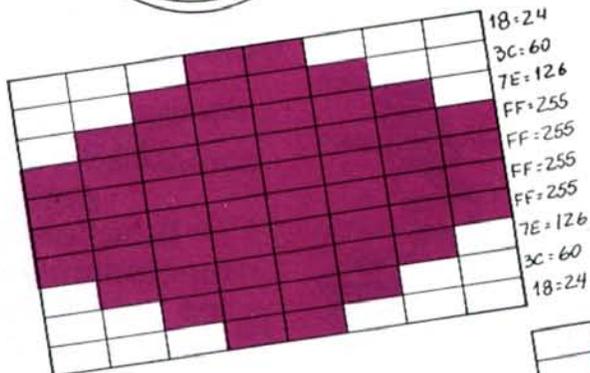
I	T	A	R	Z	A	N														
N	T	H	E				E	X	T	I	R	P	A	T	O	R				
T	A	R	K	A	N	O	I	D												
E	N	I	N	J	A															
R	P	A	C	H	A	N														
N	M	A	R	I	O				B	R	O	S	S							
A	B	R	U	C	E				L	E	E									
T	B	L	U	E					M	A	X									
I	F	O	O	T	B	A	L	L												
O	H	E	N	R	I	E			H	O	U	S	E							
N	L	I	T	T	L	E			D	E	V	I	L							
A	Z	O	R	R	O															
L	S	T	A	R					R	A	I	D	E	R	S					
K	P	O	L	A	R				P	I	E	R	R	E						
A	R	I	V	E	R				R	A	I	D								
R	M	O	N	T	E	Z	U	M	A											
A	S	U	P	E	R				S	O	C	C	E	R						
T	P	O	L	E					P	O	S	I	T	I	O	N				
E	T	A	B	L	E				S	O	C	C	E	R						



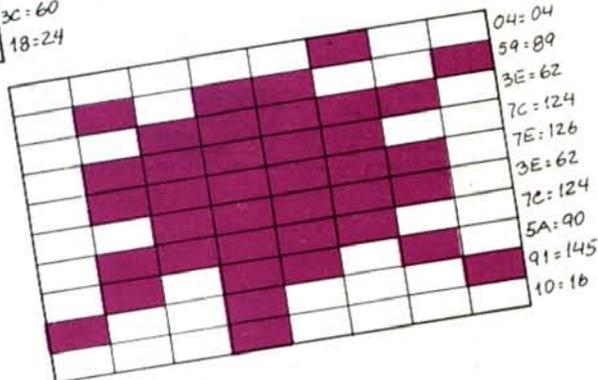
01:01
03:03
07:07
3F:63
7F:127
FF:255
FF:254
38:56
1C:28
0E:14



01:01
13:19
3F:63
7F:127
7F:127
FF:255
FE:254
7C:124
3C:60
0E:14



18:24
3C:60
7E:126
FF:255
FF:255
FF:255
FF:255
7E:126
3C:60
18:24



04:04
59:89
3E:62
7C:124
7E:126
3E:62
7C:124
5A:90
91:145
10:16

VCI
VIDEO CLUB
INTERNACIONAL
"EL COMPROMISO DE SER LIDER"

● VIDEO VIRGEN
● AUDIO VIRGEN
● CASSETTES
● MUSICA
● SOFTWARE ATARI

Le esperamos en
nuestros 22 locales.

- Vitacura 6430
- Parque Arauco, Local 176
- Parque Arauco 2 Local T - 29
- Edificio Panorámico Local 115
- Ahumada 254 Local 16
- Gran Avenida 5529 - A

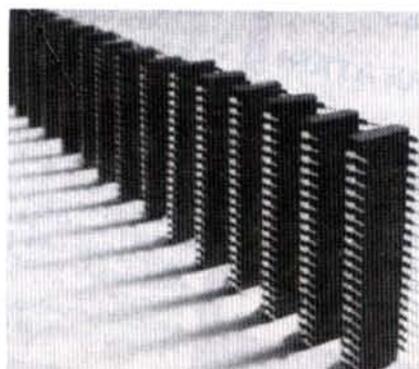
- Centro Comercial La Florida Local 36 y 37 (Al costado de Montserrat La Florida)
- Falabella Parque Arauco, Nivel 1
- Falabella Ahumada 218, 2º Piso
- Falabella Viña del Mar, 2º Piso
- Muncy Parque Arauco, Nivel 1
- Jumbo Bilbao
- Jumbo Kennedy
- Unimarc Tobalaba / Av. Apoquindo 4335
- Unimarc Portugal / Portugal 56
- Unimarc Manquehue / Av. Manquehue Sur 1700
- Unimarc Los Dominicos / Av. Apoquindo 7172
- Unimarc Puente Alto / Balmaceda 354
- Montserrat Independencia / Plaza Chacabuco
- Montserrat Walker Martinez / Walker Martinez 1650 (Quinta Normal)
- Montserrat Irarrázaval / Irarrázaval 1489
- Economax Las Rejas / Av. Ecuador 5455

ASSEM ASSEM

LECCION 10

En esta lección del curso de Assembler trataremos a fondo el sistema de comunicación del computador con sus periféricos.

Estudiaremos el funcionamiento de las rutinas del CIO (Central Input Output System) que manejan dicha comunicación y como caso práctico veremos el listado de la rutina TOMOLIN.MAC, que en nuestro sistema se utiliza para que el usuario ingrese una línea de programa o bien un comando.



BLER

MBLER

Uno de los temas más complicados de manejar desde el Assembler para nuestro Atari, es el manejo de la comunicación entre la C.P.U. y los periféricos.

Para nuestro computador, los periféricos son:

- | | |
|---------------------|----|
| - El Editor | E: |
| - El Teclado | K: |
| - La Pantalla | S: |
| - La Impresora | P: |
| - El Grabador | C: |
| - La Diskettera | D: |
| - La Interfaz RS232 | R: |

Cada uno de estos periféricos tienen una abreviatura que los menciona y que son el medio de reconocimiento para que el

computador sepa a cuál nos estamos refiriendo. Es por esto que cuando en el Basic queremos almacenar un programa en el Diskette, debemos ingresar la instrucción:

SAVE"D:NOMBRE

Cuando queremos desarrollar alguna operación de Entrada o Salida desde el computador a los periféricos o viceversa, debemos utilizar los IOCB (Input Output Control Blocks). Los IOCB, son tablas de bytes que le indican al computador los datos completos de la operación a realizar. Cada IOCB, está compuesto por 16 bytes y existen en el Sistema Operativo

8 IOCB que se encuentran localizados en las siguientes direcciones de memoria:

IOCB0	\$340
IOCB1	\$350
IOCB2	\$360
IOCB3	\$370
IOCB4	\$380
IOCB5	\$390
IOCB6	\$3A0
IOCB7	\$3B0

DIMARSA

TODO PARA SU **ATARI...** JUEGOS.

- Cassettes, Diskettes, Programas Educativos Accesorios, etc.
- Además línea completa Cassettes Audio y Video Maxell

Visite nuestra sección especializada.

CHILLAN 117 PUERTO MONTT

ASSEMBLER

ATASCII	BYTE	DIRECCION
P	50	30 E4
C	43	40 E4
E	45	00 E4
S	53	10 E4
K	4B	20 E4
	00	00 00
	00	00 00
	00	00 00
	00	00 00
	00	00 00
	00	00 00
	00	00 00
	00	00 00
	00	00 00
	00	00 00
	00	00 00

ENTRADAS LIBRES

Otro concepto importante para tener en cuenta en el sistema de comunicación son los device handler. Para cada uno de los periféricos del computador existe un handler. El handler es una tabla que informa al Sistema Operativo qué tipo de operaciones puede realizar el periférico y en qué direcciones del Sistema Operativo encuentra las rutinas que atienden a cada una de ellas.

Los handlers de los periféricos se encuentran ubicados en una tabla de handlers llamada HATABS que se encuentra localizada en la dirección \$31A. Esta tabla de Handlers tiene espacio para almacenar 14 entradas de periféricos, 5 de las cuales se definen automáticamente cuando se enciende el computador, que son los handlers que manejan la impresora, el grabador, el editor, la pantalla y el teclado.

La estructura de esta tabla de handlers es la siguiente. Cada entrada está definida por 3 bytes. El primero de ellos representa al caracter ATASCII de la abreviatura del periférico y los dos restantes representa la dirección en LO y HI bytes de la tabla de comandos que el periférico maneja. Luego de definirse las entradas de todos los periféricos en funcionamiento existe un cero que representa el final de la tabla. En el caso del Atari 800 XL la tabla de handlers está compuesta por los siguientes bytes, como se ve arriba

Si analizamos el handler de la impresora, vemos que la tabla de operaciones válidas para este periférico se encuentran almacenadas en el sistema operativo a partir de \$E430.

Siempre esta tabla de operaciones válidas tiene el mismo orden:

OPEN-1
CLOSE-1
GET-1
PUT-1
STATUS-1
SPECIAL-1
JMP INICIALIZACION

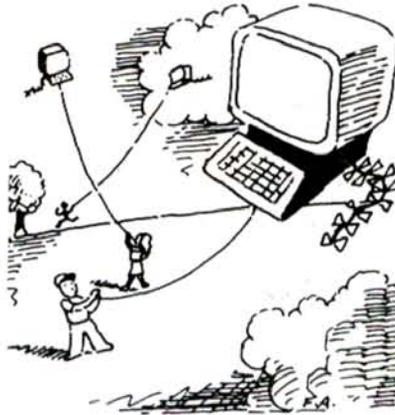
Cada periférico puede realizar sus operaciones. Para saber en qué posición del sistema operativo está la rutina que ejecuta el Open de la impresora, el S.O., primero ubica el periférico P en HATABS. Al hacerlo, sabe con los dos siguientes bytes dónde se encuentra la tabla de comandos válidos para la impresora y la rutina de open la obtiene pues está apuntada por los dos primeros bytes de esta tabla, siempre sumándole al valor de esta dirección un 1.

Esta estructura de funcionamiento de las tablas de periféricos es bastante complicada pero tiene su explicación. Si el usuario desea modificar el funcionamiento de

algún periférico lo puede hacer y bien sencillo sería. Lo único que debe hacer es modificar el contenido de la tabla HATABS y generar las rutinas que van a ocupar a este periférico. También sería fácil generar nuevos periféricos agregando entradas a la tabla HATABS.

Existe también otro bloque de información que es utilizado por este sistema de comunicación que se llama DCB o Device Control Block. Este bloque está constituido por 12 bytes y se encuentra localizado en la zona de memoria \$300.

Existen dos tipos de comunicaciones que se pueden establecer con los periféricos. La primera de ellas, que es la más utilizada es la comunicación con la estructura de un archivo que se transmite, en la cual se debe realizar primero un open, luego se ejecutan los correspondientes put bytes o put records y para cerrar el archivo, el close del mismo. Esta operación se ejecuta en Assembler con las rutinas llamadas CIO (\$E456) y utilizando los IOCB como canal de información hacia el CIO para que éste sepa qué tipo de operación tiene que realizar.



El otro tipo de comunicación es más sencillo pues en él sólo se realiza una entrada o salida de una sucesión de bytes sin estructura. Esta transmisión se ejecuta con las rutinas del SIO que se encuentran ubicadas en la dirección \$E459 y para saber qué datos se transmiten se utiliza el DCB que se encuentra en \$300. En realidad este tipo de comunicación es utilizada por el CIO pero conociendo sus parámetros puede ser

ocupada directamente por el usuario.

Antes de trabajar sobre los aspectos prácticos de las comunicaciones en Assembler, veremos la estructura detallada de los IOCB y el DCB.
IOCB 1: \$350

\$350 ICHID: Es un puntero a la tabla de periféricos, seteada por el Sistema Operativo.

\$351 ICDNO: Número de periférico como D1: o D2:

\$352 ICCOM: Código del comando a ejecutar.

\$353 ICSTA: Contiene el estado final de la última operación realizada.

\$354 \$355 ICBAL ICB AH: Es un puntero al buffer de donde se tomará o dejará la información de la operación de I/O.

\$356 \$357 ICPTL ICPTH: Estas dos posiciones contienen la dirección menos uno de la rutina de put byte definida por el Sistema Operativo.

FERIA del DISCO

M.R.

CASSETTES
VIDEO CLUB
SALON CLASICO
DISCOS COMPACTOS

PASEO AHUMADA 286 - ESTADO 350 - PROVIDENCIA esq. SUECIA

ASSEMBLER

\$358 \$359 ICBLL ICBLH:

Estas posiciones contienen la cantidad máxima de bytes a transferir por cada Get o Put. Es decir que definen la longitud del buffer a transmitir.

\$35A ICAX1: Es un byte auxiliar que es utilizado en el comando de Open para indicar si el archivo va a ser utilizado para lectura, escritura, para ambos o bien para abrir el directorio del disco.

Bit	7	6	5	4	3	2	1	0
					W	R	D	A

Cuando el bit W está en uno, el archivo fue abierto para escritura, cuando el bit R está en uno, lo fue para lectura, pero si el bit D fue seteado en uno, el archivo será utilizado

para abrir el directorio y la última operación posible la otorga el bit 0 que en uno, significa que el archivo fue abierto para Append, es decir, para incorporarle nuevos datos a éste.

\$35B ICAX2: Byte auxiliar 2.

\$35C \$35D ICAX3 ICAX4:

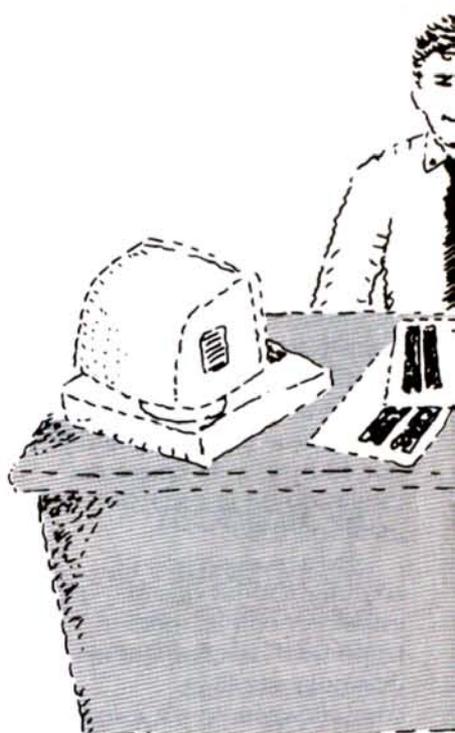
Bytes auxiliares que mantienen el número de sector dentro del disco para las instrucciones Note y Point.

\$35E ICAX5: Este byte indica cuál fue el último carácter leído dentro del sector en las operaciones de Note y Point.

\$35F ICAX6: Sin utilización.

Para el byte ICCOM existen una serie de valores asociados con las operaciones basic XIO que se corresponden con el siguiente listado:

COMANDO	DECIMAL
Open	3
Get Text Record	5
Get Binary Record	7
Put Text Record	9
Put Binary Record	11
Close	12
Rename	32



Erase	33
Protect	35
Unprotect	36
Point	37
Note	38
Format	254

Si queremos desarrollar solamente una operación de entrada o salida, siendo importante sólo la comunicación de un bloque de bytes y no una estructura de archivo, podemos ocupar el SIO (\$E459). Para esto lo único que tenemos que hacer es definir correctamente los parámetros del DCB y llamar a la rutina \$E459 o \$E453 si el periférico es la diskettera.

\$300 DDEVIC: Número de dispositivo.

Diskettera 1	49
Diskettera 2	50
Diskettera 3	51
Diskettera 4	52
Impresora	64
Cassettera	96

\$301 DUNIT: Número de unidad. 1 en el caso de la

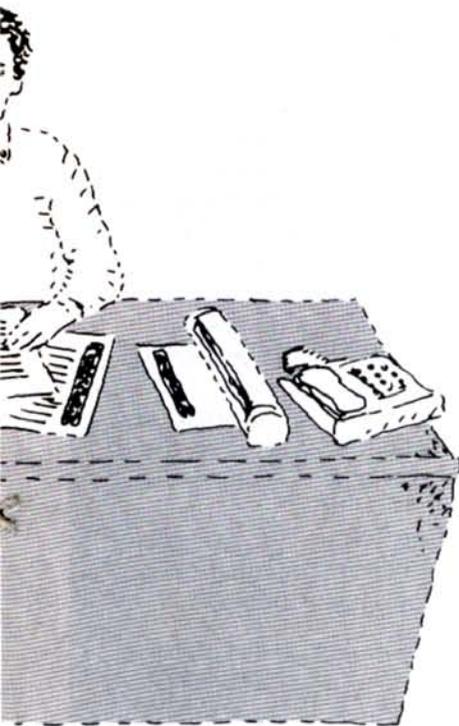
IMACO

M

R

El centro electrónico del centro de Santiago

ESTADO 46 - FONOS: 392835 - 394231



Diskettera 1, etc.

\$302 DCOMND: Código de la operación a realizar:

Read	82
Write	87
Status	83
Put	80
Format	33

\$303 DSTATS: Debe poseer un \$80 si la operación es de salida de datos y un \$40 si es una entrada de datos hacia el computador.

\$304 \$305 DBUFLO/HI: Estas posiciones apuntan a la zona de memoria de donde se sacan o dejan los datos de la operación.

\$306 DTIMELO: Es el valor máximo en tiempo que el sistema se toma para recibir datos de los periféricos.

\$307 Sin Uso.

\$308 \$309 DBYTLO/HI: Indican la cantidad de bytes a transmitir en LO y HI bytes.

\$30A \$30B DAUX1/2: Bytes auxiliares de la operación que en el caso de la diskettera indican el número de sector dentro del disco.

En el caso práctico de nuestro desarrollo, en la sección de programas de esta edición, incorporamos la rutina TOMOLIN.MAC, que está diseñada para recibir del usuario una línea por el Editor E:.

En este caso, no es necesario realizar la operación de Open pues el Editor ya se encuentra abierto por el Sistema Operativo cuando se enciende el computador.

La primera instrucción que ejecutamos es un LDX #0 para indicarle al computador que deseamos utilizar el IOCB 0 que generalmente se ocupa para el editor. Luego cargamos a \$342,X con un 5 que indica según la lista ya desarrollada un

Get Text Record.

Una vez definida la operación, con las instrucciones:

```
LDA #<LINEA
STA $344,X
LDA #>LINEA
STA $345,X
```

definimos hacia qué posiciones de memoria van a ir los bytes que el usuario ingrese. Luego con las instrucciones:

```
LDA #$FF
STA $348,X
LDA #$00
STA $349,X
```

definimos la cantidad máxima de bytes a recibir, que en este caso son 255.

Con estas instrucciones tenemos todos los parámetros de la operación listos y sólo nos queda invocar a la rutina del CIO con un:

JSR \$E456

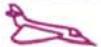
En esta sección del curso hemos hecho una introducción al sistema de comunicación del computador hacia sus periféricos, pero este es un tema bastante complejo para tratarlo en una sola lección, por lo tanto en la siguiente edición de Turbo News seguiremos analizando este tema.

COMERCIAL ESTADO

SU MEJOR ALTERNATIVA

APUMANQUE 6029 - EL FARO - TEL.: 212719
ESTADO 59 - TEL.: 339824
ESTADO 64 - TEL.: 718294



POS. DEL MES		POS. MES ANTERIOR	TITULO
1		18	RAID OVER MOSCOW
2		9	MONTEZUMA'S REVENGE
3		7	GREMLINS
4		1	HARDBALL
5		5	GHOSTCHASER
6		13	ZAXXON
7		21	SWAT
8		8	NINJA
9		2	INTERNATIONAL KARATE
10		-	DESPATCH RAIDER
11		11	BOULDER DASH I
12		-	RALLY SPEEDWAY
13		-	ELECTRICIAN
14		23	BMX SIMULATOR
15		-	WAR GAMES II
16		-	BOINA VERDE
17		10	FIGHTER PILOT II
18		-	ON TRACK
19		-	ELEKTRAGLIDE
20		19	SCREAMINGS WING 1942
21		-	TIME SLIP
22		24	STAR RAIDERS II
23		6	LEGACY II
24		3	CRYSTAL RAIDER
25		20	GHOSTBUSTER



ASCENSO



CONSTANTE

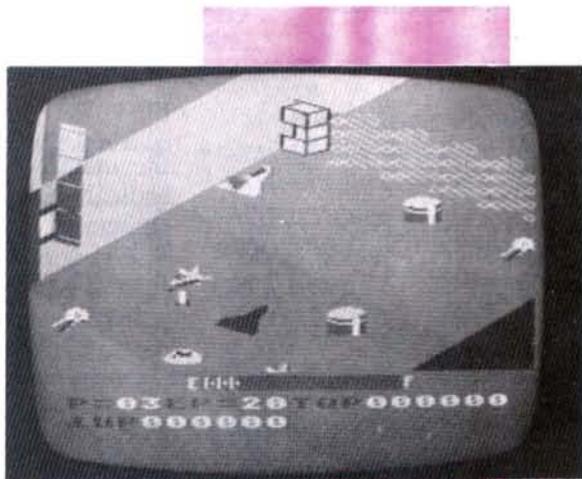


DESCENSO

ZAXXON

Programa espacial tri-dimensional. En éste, deberás descubrir la base donde se encuentra escondido el robot AL2000, que maneja todo el sistema ZAXXON. Tu misión es destruirla, pero para acercarte a ella, deberás esquivar murallas, campos magnéticos, radares, misiles anti-aéreos, ametralladoras y escuadrones de aviones cazas. No deberás descuidar el consumo de combustible, el que podrás obtener destruyendo los tanques que irán apareciendo a tu paso. Debes tener en cuenta que volando a mayor altura, el consumo de combustible es mucho mayor.

Tienes a tu izquierda un indicador de altitud que podrás utilizar para guiar tu paso y no chocar con los obstáculos que irán apareciendo a tu paso.



SONIDO	6.2
GRAFICACION	6.6
ADICION	6.6
PRESENTACION	6.6
PROMEDIO	6.5

Este es el Ranking correspondiente al mes de Mayo, obtenido en base a las estadísticas de ventas de cassettes Turbo Software en todo Chile. Recuerda que tus preferencias también serán tenidas en cuenta, para lo cual podrás escribir a Av. Fco. Bilbao 4226, Las Condes, con los juegos de tu elección.

RAID OVER MOSCOW

La tercera guerra mundial ha estallado y sólo tú, al mando de un escuadrón de combate, podrás detener los misiles con ojivas nucleares disparadas desde la Unión Soviética hacia los Estados Unidos. La única manera de detenerlos, una vez lanzados, es destruir la base que los lanzó. Tendrás tres visiones del combate. La primera, una situación general (mostrada en la foto) con la posición de tus naves defensivas, la posición de los misiles enemigos, la ciudad que los lanzó (destacada en color blanco), el nombre de ésta (en la foto Saratow), la ciudad amenazada (New York) y el tiempo de impacto (5:10). La segunda es la de tu propia base de despegue a la que podrás acceder oprimiendo la barra espaciadora. Deberás intentar hacer pasar tu nave por la compuerta sin chocarla. Una vez en el aire, conducirás tu nave hacia la ciudad atacante, luego de lo cual, pasarás automáticamente a la tercera pantalla. En ella realizarás un vuelo razante por territorio enemigo, destruyendo sus defensas. Una vez alcanzada la base tendrás que destruir sus cuatro cilos y por último la torre de control.

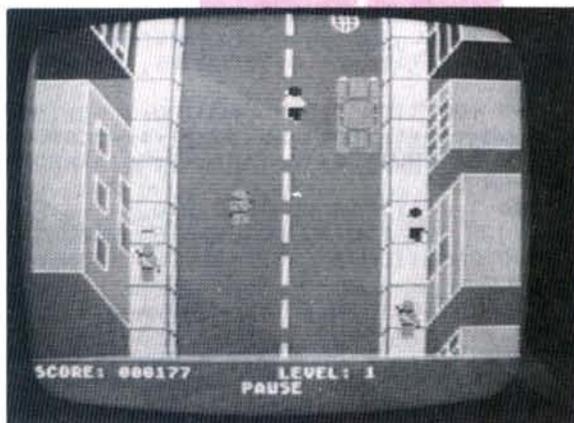
SONIDO	6.2
GRAFICACION	6.7
ADICION	6.7
PRESENTACION	6.6
PROMEDIO	6.55



SWAT

Te encuentras envuelto en una misión muy especial. Un grupo de terroristas se ha apoderado del barrio Oeste de Los Angeles. Debes dirigirte, junto con tu fuerza especial compuesta por tres hombres y rescatar los rehenes que encuentres, pero VIVOS.

Se supone que la población civil ha despejado la zona, pero siempre hay algunos que no lo consiguen. Deberás encontrar a los jefes de las bandas y matarlos, evitando herir a los ciudadanos inocentes. También deberás cuidarte de no ser golpeado por los criminales armados de garrotes, por las granadas que éstos te lanzarán y por los francotiradores que te dispararán desde los tejados.



SONIDO	6.2
GRAFICACION	6.6
ADICION	6.6
PRESENTACION	6.6
PROMEDIO	6.5

BASIC

```

10 DIM EQUIPO$(384),TABLA(7,24),NOM$(
15),BLANCO$(16),ZON$(1):BLANCO$(1)=
" ":BLANCO$(16)=" "
15 BLANCO$(2)=BLANCO$:EQUIPO$(1)=" "
:EQUIPO$(384)=" ":EQUIPO$(2)=EQUIPO$
17 FOR I=1 TO 7:FOR J=1 TO 24:TABLA(
I,J)=0:NEXT J:NEXT I
20 FOR I=1 TO 24
30 READ NOM$:EQUIPO$((I-1)*16+1,I*16
)=NOM$:NEXT I
40 ? CHR$(125):POSITION 10,3:? "MENU
PRINCIPAL"
50 POSITION 3,7:? "1- INGRESO RESULT
ADOS"
60 POSITION 3,9:? "2- TABLA POSICION
ES GENERALES"
70 POSITION 3,11:? "3- TABLA POSICIO
NES POR GRUPO"
80 POSITION 3,13:? "4- TABLA GOLES F
AVOR GENERALES"
90 POSITION 3,15:? "5- TABLA GOLES E
N CONTRA GENERALES"
100 TRAP 100:POSITION 5,18:? "INGRES
E SU OPCION";
110 INPUT OPCION
120 ON OPCION GOTO 200,390,500,650,7
00
130 GOTO 100
200 ? CHR$(125):TRAP 40000
210 POSITION 3,3:? "EQUIPO 1 ";;INPU
T NOM$:I=1:NOM$(LEN(NOM$)+1)=BLANCO$
220 IF EQUIPO$((I-1)*16+2,I*16)=NOM$
THEN 250
230 I=I+1:IF I<25 THEN 220
240 ? "EL EQUIPO NO EXISTE":GOTO 210
250 POSITION 30,3:? "GOLES ";;INPUT
GOL1
260 POSITION 3,5:? "EQUIPO 2 ";;INPU
T NOM$:J=1:NOM$(LEN(NOM$)+1)=BLANCO$
270 IF EQUIPO$((J-1)*16+2,J*16)=NOM$
THEN 300
280 J=J+1:IF J<25 THEN 270
290 ? "EL EQUIPO NO EXISTE":GOTO 260
300 POSITION 30,5:? "GOLES ";;INPUT
GOL2
310 IF GOL1>GOL2 THEN TABLA(2,I)=TAB
LA(2,I)+1:TABLA(4,J)=T
ABLA(4,J)+1:TABLA(7,I)=TABLA(7,I)+23
20 IF GOL2>GOL1 THEN TABLA(4,I)=TABL
A(4,I)+1:TABLA(2,J)=T
ABLA(2,J)+1:TABLA(7,J)=TABLA(7,J)+2
330 IF GOL2=GOL1 THEN TABLA(3,I)=TAB
LA(3,I)+1:TABLA(3,J)=T
ABLA(3,J)+1:TABLA(7,I)=TABLA(7,I)+1:
TABLA(7,J)=TABLA(7,J)+1
340 TABLA(1,I)=TABLA(1,I)+1:TABLA(1,
J)=TABLA(1,J)+1
350 TABLA(5,I)=TABLA(5,I)+GOL1:TABLA
(5,J)=TABLA(5,J)+GOL2
360 TABLA(6,I)=TABLA(6,I)+GOL2:TABLA
(6,J)=TABLA(6,J)+GOL1
370 GOTO 40
390 AUXI=7
400 ? CHR$(125):POSITION 6,6:? "UN M
OMENTO POR FAVOR..."
410 FOR I=1 TO 23
420 FOR J=I+1 TO 24
430 IF TABLA(AUXI,I)<TABLA(AUXI,J) T
HEN GOSUB 2000
440 NEXT J:NEXT I
450 ? CHR$(125);"EQUIPO          J  G
E  P  PTS  GF  GC";CHR$(29)
460 FOR I=1 TO 20
470 ? EQUIPO$((I-1)*16+2,I*16);TABLA
(1,I);" ";TABLA(2,I);" ";TABLA(3,I
);" ";TABLA(4,I);" ";TABLA(7,I);"
";
480 ? TABLA(5,I);" ";TABLA(6,I):NE
XT I:? CHR$(29);"OPRIMA RETURN PARA
CONTINUAR";:INPUT ZON$:? CHR$(125)
482 FOR I=21 TO 24
484 ? EQUIPO$((I-1)*16+2,I*16);TABLA
(1,I);" ";TABLA(2,I);" ";TABLA(3,I
);" ";TABLA(4,I);" ";TABLA(7,I);"
";
486 ? TABLA(5,I);" ";TABLA(6,I):NE
XT I:? CHR$(29);"OPRIMA RETURN PARA
CONTINUAR";:INPUT ZON$
490 GOTO 40
500 ? CHR$(125);:POSITION 2,4:? "ING
RESE ZONA";:INPUT ZON$:POSITION 10,1
0:? "UN MOMENTO..."
510 IF ZON$>"F" OR ZON$<"A" THEN 500
520 FOR I=1 TO 23
530 FOR J=I+1 TO 24
535 AUXI=(I-1)*16+1:AUXJ=(J-1)*16+1
540 IF EQUIPO$(AUXI,AUXI)=ZON$ AND E
QUIPO$(AUXJ,AUXJ)=ZON$ AND TABLA(7,I
)<TABLA(7,J) THEN GOSUB 2000
560 NEXT J:NEXT I
570 ? CHR$(125);"EQUIPO          J  G
E  P  PTS  GF  GC"
580 FOR I=1 TO 24
590 IF EQUIPO$((I-1)*16+1,(I-1)*16+1
)=ZON$ THEN GOSUB 620
600 NEXT I:? CHR$(29);"OPRIMA RETURN
PARA CONTINUAR";:INPUT ZON$
610 GOTO 40
620 ? EQUIPO$((I-1)*16+2,I*16);TABLA
(1,I);" ";TABLA(2,I);" ";TABLA(3,I
);" ";TABLA(4,I);" ";TABLA(7,I);"
";
630 ? TABLA(5,I);" ";TABLA(6,I):RE
TURN
650 AUXI=5:GOTO 400
700 ? CHR$(125):POSITION 6,6:? "UN M
OMENTO POR FAVOR..."
710 FOR I=1 TO 23
720 FOR J=I+1 TO 24
730 IF TABLA(6,I)>TABLA(6,J) THEN GO
SUB 2000
740 NEXT J:NEXT I
750 GOTO 450
1000 DATA AITALIA,AAUSTRIA,AEE.UU.,A
CHECOSLOVAQUIA
1010 DATA BARGENTINA,BCAMERUN,BURSS,
BRUMANIA
1020 DATA CBRASIL,CSUECIA,CCOSTA RIC
A,CESCOCIA
1030 DATA DE. ARABES,DCOLOMBIA,DALEM
ANIA OCC.,DYUGOSLAVIA
1040 DATA EBELGICA,ECOREA DEL SUR,EU

```

```

RUGUAY,EESPAÑA
1050 DATA FINGLATERRA,FIRLANDA,FHOLA
NDA,FEGIPTO
2000 NOM$=EQUIPO$( (I-1)*16+1,I*16 ):E
QUIPO$( (I-1)*16+1,I*16 )=EQUIPO$( (J-1
)*16+1,J*16 ):EQUIPO$( (J-1)*16+1,J*16
)=NOM$
2020 FOR X=1 TO 7:AUX=TABLA(X,I):TAB
LA(X,I)=TABLA(X,J):TABLA(X,J)=AUX:NE
XT X
2030 RETURN

```

BOOT FILES

```

100 LOADADDRESS=$2000
110 MEMLO=$2E7
120 APPMHI=$0E
130 DOSVEC=$0A
140 *$=LOADADDRESS
150 .BYTE $0 ; PRIMER BYTE DEL VE
CTDR
160 .BYTE LOADADDRESS-END+127/128
; CANTIDAD DE SECTORES
170 .WORD LOADADDRESS
180 .WORD INITADRES
190 INICIALIZACION-DE-RAM
200 LDA #<END
210 STA MEMLO
220 STA APPMHI
230 LDA #>END
240 STA MEMLO+1
250 STA APPMHI+1
260 LDA #<PROGPRINCIPAL
270 STA DOSVEC
280 LDA #>PROGPRINCIPAL
290 STA DOSVEC+1
300 CLC
310 RTS
320 INITADRES
330 ;
340 ; Aquí viene la rutina que
350 ; inicializa nuestro programa
360 ; principal si es necesario.
370 ;
380 RTS
390 ;
400 PROGPRINCIPAL
410 ;
420 ; Aquí viene el software princip
al
430 ;
440 ;
450 END

```

TECNICAS DE PROTECCION

```

10 DCB=768
20 DIM BUFFER$(128),BLANCO$(125),AUX
$(2),PRG$(6)
30 BUFFER$(1,1)=" ":BUFFER$(128,128)
=" ":BUFFER$(2)=BUFFER$:BLANCO$=BUFF
ER$
35 BLANCO$(1,1)=CHR$(0):BLANCO$(125,
125)=CHR$(0):BLANCO$(2)=BLANCO$

```

```

37 FOR I=1 TO 5: READ A: PROG$(I,I)=
CHR$(A): NEXT I
40 ? CHR$(125):POSITION 12,3:? "MENU
PRINCIPAL"
50 POSITION 1,7:? "1)BUSQUEDA DE SEC
TORES EN BLANCO"
60 POSITION 1,10:? "2)MOVER DIRECTOR
IO"
70 POSITION 1,13:? "3)DISFRAZAR DIRE
CTORIO"
80 POSITION 3,18:? "SU OPCION ";
90 TRAP 90:INPUT OPCION
100 ON OPCION GOTO 200,400,600
110 GOTO 90
200 GOSUB 1000:? CHR$(125):POSITION
10,8:? "TRABAJANDO...":CANT=0
205 POKE DCB+2,82
210 FOR SECTOR=255 TO 510
220 GOSUB 2000
230 IF BUFFER$(1,125)<>BLANCO$ THEN
CANT=CANT+1:GOTO 245
240 CANT=CANT+1:IF CANT=8 THEN POP :
GOTO 260
245 POSITION 5,14:? "SECTOR: ";SECTO
R:" BLANCOS: ";CANT
250 NEXT SECTOR
260 POSITION 5,20:? "MUEVA AL SECTOR
: ";SECTOR-7
270 FOR I=1 TO 1000:NEXT I:GOTO 30
400 ? CHR$(125);"INGRESE NUMERO DE S
ECTOR INICIAL: ";
410 INPUT OPCION
420 FOR I=1 TO 8
430 POKE DCB+2,82:GOSUB 1000
440 SECTOR=360+I:GOSUB 2000:POSITION
5,10:? "SECTOR: ";360+I:? CHR$(29);B
UFFER$
450 POKE DCB+2,87:GOSUB 1000
460 SECTOR=OPCION-1+I:GOSUB 2000
470 NEXT I:POKE 4171,105+OPCION-361:
GOTO 30
600 FOR I=0 TO 8
610 ? CHR$(125);"INGRESE DISCO CUALO
UIERA Y OPRIMA RETURN";:INPUT AUX$
620 POKE DCB+2,82:GOSUB 1000
630 SECTOR=360+I:GOSUB 2000:POSITION
5,10:? "SECTOR: ";360+I:? CHR$(29);B
UFFER$
635 ? "INGRESE DISCO A PROTEGER Y OP
RIMA RETURN";:INPUT AUX$
640 POKE DCB+2,87:GOSUB 1000
650 SECTOR=360+I:GOSUB 2000
660 NEXT I:GOTO 30
1000 ADDR=ADR(BUFFER$)
1010 ADDRHI=INT(ADDR/256)
1020 ADDRLO=ADDR-(ADDRHI*256)
1030 POKE DCB+4,ADDRLO
1040 POKE DCB+5,ADDRHI
2000 SECTORHI=INT(SECTOR/256)
2100 SECTORLO=SECTOR-(SECTORHI*256)
2200 POKE DCB+10,SECTORLO
2300 POKE DCB+11,SECTORHI
2400 X=USR(ADR(PROG$)):RETURN
2410 DATA 104,32,83,228,96

```

GRAFICOS X COMPUTADORA

```

10 DIM NAVE$(256),TANQUE$(256),MI$(2
56),CUE
RPO$(14),AVION$(10),RUEDA$(8),BLANCO
$(256),MISIL$(10),EXPLO$(30)
30 BLANCO$(1)=CHR$(0):BLANCO$(255)=C
HR$(0):BLANCO$(2)=BLANCO$:GRAPHICS 3
+16:BASE=PEEK(106)-16
40 POKE 54279,BASE
50 PMBASE=BASE*256:FOR I=PMBASE+768
TO PMBASE+1536:POKE I,0:NEXT I:REM B
ORRA ZONA DE PLAYERS
60 POKE 559,62:REM HABILITACION PLAY
ERS-MISSILES RESOLUCION 2 LINEAS
70 POKE 53277,3:REM HABILITACION PLA
YERS Y MISSILES
80 REM MODIFICACION DE VVTP PARA QUE
APUNTE A ZONA DE PLAYERS
90 STARP=PEEK(140)+PEEK(141)*256
100 VVTP=PEEK(134)+PEEK(135)*256
110 I=PMBASE+1024-STARP
120 HB=INT(I/256)
130 POKE VVTP+2,I-HB*256
140 POKE VVTP+3,HB
150 I=PMBASE+1280-STARP
160 HB=INT(I/256)
170 POKE VVTP+10,I-HB*256
180 POKE VVTP+11,HB
182 I=PMBASE+768-STARP
184 HB=INT(I/256)
186 POKE VVTP+18,I-HB*256
188 POKE VVTP+19,HB
190 FOR I=1 TO 14:READ A:CUERPO$(I,I)
)=CHR$(A):NEXT I
192 FOR I=1 TO 8:READ A:RUEDA$(I,I)=
CHR$(A):NEXT I
200 FOR I=1 TO 10:READ A:AVION$(I,I)
)=CHR$(A):NEXT I
202 FOR I=1 TO 10:READ A:MISIL$(I,I)
)=CHR$(A):NEXT I
204 FOR I=1 TO 30:READ A:EXPLO$(I,I)
)=CHR$(A):NEXT I
205 POKE 710,66:POKE 704,15:POKE 705
,7:GOSUB 3000
210 TANQUE$(200)=CUERPO$:CLA=1
220 HPOS=50:HPOS2=220:RAN=3:RAN2=10
230 JOY=STICK(0)
240 IF JOY=7 THEN HPOS=HPOS2-(HPOS>
200)*2:CLA=(CLA=1)*2+(CLA=2):GOTO 26
0
250 IF JOY=11 THEN HPOS=HPOS2-2+(HPOS
<50)*2:CLA=(CLA=1)*2+(CLA=2)
260 TANQUE$(215)=RUEDA$((CLA-1)*4+1,
CLA*4):HPOS2=HPOS2-RAN
265 IF HPOS2<10 THEN HPOS2=220:RAN=I
NT(RND(0)*3)+1:RAN2=I
NT(RND(0)*13)+1:NAVE$=BLANCO$:NAVE$(
4+RAN2*10)=AVION$
270 POKE 53249,HPOS:POKE 53248,HPOS2
275 IF CLA2=1 THEN VPOS=VPOS-6:MI$(V
POS)=MISIL$:IF VPOS<7 THEN CLA2=0:PO
KE 53253,0
276 IF PEEK(53257)<>0 THEN GOTO 1000
280 IF STRIG(0)=1 OR CLA2=1 THEN 230

```

```

290 POKE 53253,HPOS:CLA2=1:VPOS=200
300 GOTO 230
1000 SOUND 1,12,8,10
1010 NAVE$(4+RAN2*10)=EXPLO$(1,10):F
OR I=1 TO 60:NEXT I
1020 SOUND 1,16,8,10
1030 NAVE$(4+RAN2*10)=EXPLO$(11,20):
FOR I=1 TO 60:NEXT I
1040 SOUND 1,12,8,10
1050 NAVE$(4+RAN2*10)=EXPLO$(21,30):
FOR I=1 TO 60:NEXT I:SOUND 1,0,0,0:PO
KE 53253,0:HPOS2=5:POKE 53278,0:GOT
O 230
3000 POSITION 0,21:? #6;"AAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA"
3020 ? #6;"AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA"
3030 ? #6;"AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA"
3050 FOR I=1 TO 3:POSITION 32,10+I*2
:? #6;"AAAAAAA":POSITION 32,11+I*2:?
#6;"A A A":NEXT I
3060 POSITION 32,18:? #6;"AAAAAAA":F
OR I=1 TO 2:POSITION 32,18+I:? #6;"A
A AA":NEXT I
3070 FOR I=1 TO 2:POSITION 22,15+I*2
:? #6;"BBBBBBBBBB":POSITION 22,16+I*
2:? #6;"B BB B":NEXT I
3140 POSITION 0,15:FOR I=1 TO 3:? #6
;"
CCCCCC":? #6;"
C C C":NEXT I
3150 POSITION 0,10:FOR I=1 TO 4:? #6
;"
BBBBBBB":? #6;" B.B
B B":NEXT I:? #6;" BBBBBBB"
3160 FOR I=1 TO 2:? #6;" BB
BB":NEXT I
3170 POSITION 0,17:? #6;"AAAAAA "
3180 ? #6;"AAAAAAA":FOR I=1 TO 2:?
#6;"AA AA":NEXT I
4000 RETURN
10000 DATA 16,16,16,16,16,16,56,56,1
24,124,255,255,255,255
10010 DATA 84,1,128,42,42,128,1,84
10030 DATA 01,03,07,63,127,255,254,5
6,28,14
10040 DATA 4,4,4,4,0,0,0,0,0,0
10050 DATA 1,19,63,127,127,255,254,1
24,60,14,24
,60,126,255,255,255,126,60,24,4,
89,62,124,126,62,124,90,145,16

```

ASSEMBLER I

```

0100 ;SAVE#D:TOMOLIN.MAC
0110 TOMOLINEA
0120 ;
0130 ; RUTINA PARA EL INGRESO DE UNA
0140 ; LINEA DESDE EL TECLADO QUE SE
0150 ; ALMACENARA EN LA DIRECCION DE
0160 ; MEMORIA LINEA.
0170 ;
0180 LDX #0
0190 LDA #$05
0200 STA $0342,X
0210 LDA # <LINEA
0220 STA $0344,X

```

```

0230 LDA # >LINEA
0240 STA $0345,X
0250 LDA #$FF
0260 STA $0348,X
0270 LDA #0
0280 STA $0349,X
0290 ;
0300 ; LLAMADA AL CID.
0310 ;
0320 JSR $E456
0330 RTS
0340 TOMOLINEA:
0350 ;
0360 ; RUTINA PARA EL INGRESO DE UNA
0370 ; LINEA DESDE EL TECLADO QUE SE
0380 ; ALMACENARA EN LA DIRECCION DE
0390 ; MEMORIA LINEA + EL CONTENIDO
0400 ; DE GUARDOY, EN EL CASO QUE EL
0410 ; PROGRAMA HAYA COLOCADO EL NU-
0420 ; MERO DE LA INSTRUCCION EN LA
0430 ; PANTALLA.
0440 ;
0450 LDX #0
0460 LDA #$05
0470 STA $0342,X
0480 CLC
0490 LDA # <LINEA
0500 ADC GUARDOY
0510 STA $0344,X
0520 LDA # >LINEA
0530 ADC #0
0540 STA $0345,X
0550 LDA #$FF
0560 STA $0348,X
0570 LDA #0
0580 STA $0349,X
0590 ;
0600 ; LLAMADA AL CID
0610 ;
0620 JSR $E456
0630 RTS

0100 ;SAVE#D:XREN.MAC
0110 ;
0120 ; RUTINA DE RENUMERACION DEL
0130 ; LISTADO
0140 ;
0150 REN
0160 JSR TOMOPARAM
0170 JSR XREN
0180 RTS
0190 ;
0200 ; SUBROUTINA QUE SOLICITA AL
0210 ; USUARIO LOS PARAMETROS DE
0220 ; LA RENUMERACION
0230 ;
0240 TOMOPARAM
0250 ;
0260 ; IMPRIME EN LA PANTALLA EL
0270 ; MENSAJE CONTENIDO EN ROTDESDE
0280 ;
0290 LDA # <ROTDESDE
0300 STA MANDO
0310 LDA # >ROTDESDE
0320 STA MANDO+1

0330 JSR IMPRIMO
0340 JSR BORROAUX
0350 ;
0360 ; LE PIDE AL USUARIO EL NUMERO
0370 ; INICIAL PARA LA RENUMERACION
0380 ; SI ES UN RETURN ASUME UN 100
0390 ;
0400 JSR INGRESONUMERO
0410 LDX GUARDOX
0420 CPX #0
0430 BNE NODEFAULT
0440 LDA #100
0450 STA DESDE
0460 LDA #0
0470 STA DESDE+1
0480 LDA # <ROT100
0490 STA MANDO
0500 LDA # >ROT100
0510 STA MANDO+1
0520 JSR IMPRIMO
0530 JMP INCREM
0540 NODEFAULT
0550 ;
0560 ; CON LAS RUTINAS DEL FLOATING
0570 ; POINT CONVIERTE EL NUMERO
0580 ; INGRESADO EN ATASCII EN DOS
0590 ; BYTES.
0600 ;
0610 LDA #0
0620 STA CIX
0630 JSR $D800
0640 JSR $D9D2
0650 LDA FRO
0660 STA DESDE
0670 LDA FRO+1
0680 STA DESDE+1
0690 INCREM
0700 JSR BORROAUX
0710 ;
0720 ; IMPRIME EN LA PANTALLA EL
0730 ; CONTENIDO DE ROTINCRE PARA
0740 ; QUE EL USUARIO DEFINA EL
0750 ; INCREMENTO ENTRE LINEA Y
0760 ; LINEA.
0770 ;
0780 LDA # <ROTINCRE
0790 STA MANDO
0800 LDA # >ROTINCRE
0810 STA MANDO+1
0820 JSR IMPRIMO
0830 ;
0840 ; LE PIDE AL USUARIO QUE INGRESE
0850 ; EL INCREMENTO. SI PRESIONA RE-
0860 ; TURN, ASUME QUE ESTE ES 10.
0870 ;
0880 JSR INGRESONUMERO
0890 LDX GUARDOX
0900 CPX #0
0910 BNE NODEFAULT2
0920 LDA #10
0930 STA INCREMENTO
0940 LDA #0
0950 STA INCREMENTO+1
0960 LDA # <ROT10
0970 STA MANDO
0980 LDA # >ROT10
0990 STA MANDO+1
1000 JSR IMPRIMO

```

ASSEMBLER II

```

1010      JMP SALIDA
1020 NODEFAULT2
1030 ;
1040 ; CONVIERTE EL INCREMENTO EN
1050 ; DOS BYTES.
1060 ;
1070      LDA #0
1080      STA CIX
1090      JSR $DB00
1100      JSR $D9D2
1110      LDA FR0
1120      STA INCREMENTO
1130      LDA FR0+1
1140      STA INCREMENTO+1
1150 SALIDA
1160      LDA #$9B
1170      JSR $F2B0
1180      RTS
1190 INGRESONUMERO
1200      LDX #0
1210      STX GUARDOX
1220 LOPING
1230      JSR $F2F8
1240      STA GUARDOA
1250      CMP #'0
1260      BCC ESRET
1270      CMP #' :
1280      BCS ESRET
1290      JSR $F2B0
1300      LDA GUARDOA
1310      LDX GUARDOX
1320      STA AUXILIAR,X
1330      LDX GUARDOX
1340      INX
1350      STX GUARDOX
1360      CPX #5
1370      BNE LOPING
1380      JMP FINING
1390 ESRET
1400      CMP #$9B
1410      BEQ FINING
1420      CMP #'
1430      BNE LOPING
1440      LDX GUARDOX
1450      CPX #0
1460      BEQ LOPING
1470      DEX
1480      STX GUARDOX
1490      LDA #$9B
1500      STA AUXILIAR,X
1510      LDA #'
1520      JSR $F2B0
1530      JMP LOPING
1540 FINING
1550      LDA # <AUXILIAR
1560      STA INBUFF
1570      LDA # >AUXILIAR
1580      STA INBUFF+1
1590      RTS
1600 XREN
1610 ;
1620 ; GUARDA LOS REGISTROS EN LA
1630 ; PILA.
1640 ;
1650      TYA
1660      PHA
1670      TXA
1680      PHA
1690 ;
1700 ; CORREDOR=TABLA
1710 ;
1720      LDA # <TABLA
1730      STA CORREDOR
1740      LDA # >TABLA
1750      STA CORREDOR+1
1760 ;
1770 ; TABLA VACIA?
1780 ;
1790      LDA CORREDOR
1800      CMP FINTAB
1810      BNE INCRE
1820      LDA CORREDOR+1
1830      CMP FINTAB+1
1840      BEQ RECREG
1850 ;
1860 ; INCREMENTO=0 ?
1870 ;
1880 INCRE
1890      LDA INCREMENTO
1900      BNE COLOCAR
1910      LDA INCREMENTO+1
1920      BNE COLOCAR
1930      JSR PARAINV
1940      JMP RECREG
1950 ;
1960 ; ASIGNA EL NUEVO VALOR DEL
1970 ; NUMERO DE LINEA A LA
1980 ; INSTRUCCION.
1990 ;
2000 COLOCAR
2010      LDY #0
2020      LDA DESDE
2030      STA (CORREDOR),Y
2040      LDY #1
2050      LDA DESDE+1
2060      STA (CORREDOR),Y
2070 ;
2080 ; DESPLAZA A CORREDOR HASTA LA
2090 ; PROXIMA INSTRUCCION
2100 ;
2110      CLC
2120      LDY #2
2130      LDA (CORREDOR),Y
2140      ADC CORREDOR
2150      STA CORREDOR
2160      LDA CORREDOR+1
2170      ADC #0
2180      STA CORREDOR+1
2190 ;
2200 ; CORREDOR=CORREDOR+3
2210 ;
2220      CLC
2230      LDA CORREDOR
2240      ADC #3
2250      STA CORREDOR
2260      LDA CORREDOR+1
2270      ADC #0
2280      STA CORREDOR+1
2290 ;
2300 ; CORREDOR=FINTAB ?
2310 ;
2320      LDA CORREDOR
2330      CMP FINTAB
2340      BNE PROX
2350      LDA CORREDOR+1
2360      CMP FINTAB+1

```

```

2370      BEQ  RECREG
2380 PROX
2390      CLC
2400      LDA  DESDE
2410      ADC  INCREMENTO
2420      STA  DESDE
2430      LDA  DESDE+1
2440      ADC  INCREMENTO+1
2450      STA  DESDE+1
2460      BCC  COLOCAR
2470      JSR  OVERFLOW
2480 ;
2490 ; RECUPERA LOS REGISTROS
2500 ;
2510 RECREG
2520      PLA
2530      TAX
2540      PLA
2550      TAY
2560      RTS
2570 PARAINV
2580      LDA  # <ROTPARAINV
2590      STA  MANDO
2600      LDA  # >ROTPARAINV
2610      STA  MANDO+1
2620      JSR  IMPRIMO
2630      RTS
2640 ROTDESDE
2650      .BYTE $9B,"DESDE: *"
2660 ROTINCRE
2670      .BYTE $9B,"INCREMENTO: *"
2680 AUXILIAR
2690      .BYTE " "
2700 ROTPARAINV
2710      .BYTE $9B,"2 PARAMETROS INV
ALIDOS", $9B, "*"
2720 DESDE
2730      .BYTE " "
2740 INCREMENTO
2750      .BYTE " "
2760 BORROAUX
2770      LDY  #0
2780 LOPBOR
2790      LDA  #0
2800      STA  AUXILIAR,Y
2810      INY
2820      CPY  #5
2830      BNE  LOPBOR
2840      RTS
2850 ROT100
2860      .BYTE "100*"
2870 ROT10
2880      .BYTE "10*"
0200 ; LLAMADA: .INCLUDE#D:CONTAB.MAC
0210 ;
0220 ; JSR CONTAB
0230 ;
0240 ; ENTRADA:
0250 ;
0260 ; *ACUMULADOR=CANTIDAD DE
0270 ; BYTES A BORRAR
0280 ;
0290 ; *FUNCON=DIRECCION
0300 ; EN LA TABLA DEL PRIMER BYTE A
0310 ; BORRAR
0320 ;
0330 ; *FINTAB=PUNTERO AL
0340 ; PRIMER BYTE LIBRE DE LA TABLA
0350 ;
0360 ;
0370 ; SALIDA:
0380 ;
0390 ;
0400 ; *FINTAB=NUEVO VALOR
0410 ;
0420 ;
0430 CONTAB
0440 ;
0450 ; GUARDA LOS REGISTROS EN LA PILA
0460 ;
0470      STA  HUECO
0480      TYA
0490      PHA
0500      TXA
0510      PHA
0520      LDA  PUNCON
0530      STA  CORREDOR
0540      LDA  PUNCON+1
0550      STA  CORREDOR+1
0560 ;
0570 ; DEFINE EL NUEVO VALOR DE
0580 ; FINTAB
0590 ;
0600      SEC
0610      LDA  FINTAB
0620      SBC  HUECO
0630      STA  FINTAB
0640      LDA  FINTAB+1
0650      SBC  #0
0660      STA  FINTAB+1
0670 ; CORRE LOS BYTES PARA RELLENAR
0680 ; EL ESPACIO GENERADO
0690 ;
0700 TRAE
0710      LDY  HUECO
0720      LDA  (CORREDOR),Y
0730      LDY  #0
0740      STA  (CORREDOR),Y
0750      CLC
0760      LDA  CORREDOR
0770      ADC  #1
0780      STA  CORREDOR
0790      LDA  CORREDOR+1
0800      ADC  #0
0810      STA  CORREDOR+1
0820      LDA  CORREDOR+1
0830      CMP  FINTAB+1
0840      BCC  TRAE
0850      LDA  CORREDOR
0860      CMP  FINTAB
0870      BCC  TRAE

```

ASSEMBLER III

```

0100 ;SAVE#D:CONTAB.MAC
0110 ;
0120 ; PROPOSITO:
0130 ;
0140 ; BORRA BYTES DE
0150 ; LA TABLA DE INSTRUCCIONES
0160 ; PARA PODER INSERTAR UNA
0170 ; NUEVA LINEA MAS CORTA QUE
0180 ; LA QUE REEMPLAZA
0190 ;

```

```

0100 ;SAVE#D:CONTAB.MAC
0110 ;
0120 ; PROPOSITO:
0130 ;
0140 ; BORRA BYTES DE
0150 ; LA TABLA DE INSTRUCCIONES
0160 ; PARA PODER INSERTAR UNA
0170 ; NUEVA LINEA MAS CORTA QUE
0180 ; LA QUE REEMPLAZA
0190 ;

```

```

0880 ;
0890 ; RECUPERA DE LA PILA LOS
0900 ; REGISTROS
0910 ;
0920     PLA
0930     TAX
0940     PLA
0950     TAY
0960     RTS

```

ASSEMBLER IV

```

0100 ;SAVE#D:EXTAB.MAC
0110 ;
0120 ; PROPOSITO:
0130 ;
0140 ;             GENERA UN HUECO EN
0150 ; LA TABLA DE INSTRUCCIONES
0160 ; PARA PODER INSERTAR UNA
0170 ; NUEVA LINEA
0180 ;
0190 ; LLAMADA: .INCLUDE#D:EXTAB.MAC
0200 ;
0210 ;             JSR EXTAB
0220 ;
0230 ; ENTRADA:
0240 ;
0250 ;             *ACUMULADOR=CANTIDAD DE
0260 ; BYTES A INSERTAR
0270 ;
0280 ;             *POINTERAUX=DIRECCION.
0290 ; EN LA TABLA DEL PRIMER BYTE A
0300 ; INSERTAR
0310 ;
0320 ;             *FINTAB=PUNTERO AL
0330 ; PRIMER BYTE LIBRE DE LA TABLA
0340 ;
0350 ;
0360 ; SALIDA:
0370 ;             *POINTERAUX=INTACTO
0380 ;
0390 ;             *FINTAB=NUEVO VALOR
0400 ;
0410 EXTAB
0420 ;
0430 ; GUARDA LOS REGISTROS
0440 ;
0450     STA HUECO

```

```

0460     TYA
0470     PHA
0480     TXA
0490     PHA
0500     LDA FINTAB
0510     STA CORREDOR
0520     LDA FINTAB+1
0530     STA CORREDOR+1
0540 ;
0550 ; DEFINE A FINTAB CON SU NUEVO
0560 ; VALOR.
0570 ;
0580     CLC
0590     LDA FINTAB
0600     ADC HUECO
0610     STA FINTAB
0620     LDA FINTAB+1
0630     ADC #0
0640     STA FINTAB+1
0650 ;
0660 ; DESPLAZA LOS BYTES PARA
0670 ; GENERAR EL ESPACIO NECESARIO
0680 ;
0690 DESPLAZA
0700     SEC
0710     LDA CORREDOR
0720     SBC #1
0730     STA CORREDOR
0740     LDA CORREDOR+1
0750     SBC #0
0760     STA CORREDOR+1
0770     LDY #0
0780     LDA (CORREDOR),Y
0790     LDY HUECO
0800     STA (CORREDOR),Y
0810     LDA CORREDOR
0820     CMP POINTERAUX
0830     BNE DESPLAZA
0840     LDA CORREDOR+1
0850     CMP POINTERAUX+1
0860     BNE DESPLAZA
0870 ;
0880 ; RECUPERA LOS REGISTROS
0890 ;
0900     PLA
0910     TAX
0920     PLA
0930     TAY
0940     RTS

```



La nueva generación de Software para Computadores Atari

ADQUIERALOS EN LOS SIGUIENTES PUNTOS DE VENTAS

• **ANTOFAGASTA:** COOPERCARAB / KW VIDEO / LA ESPAÑOLA • **VIÑA DEL MAR:** FALABELLA VIÑA / INSIS / MPR COMPUTACION • **VALPARAISO:** COMPUTRONIC • **SANTIAGO:** AUDIO BICICLETA INTERNAC / CASA ROYAL / CENTRO ATARI / COMERCIAL ESTADO / COMPUMANQUE / COMPUCENTER / FALABELLA AHUMADA / FALABELLA P. ARAUCO / IMACO / INFOGROUP / PC STORE / PETERSEN / ROLEC / SUPERMERCADOS UNIMARC / TASCO / VIDEO CLUB INTERNACIONAL • **RANCAGUA:** CASA ZUNIGA • **CURICO:** MULTIHOJAR • **TALCA:** LIBRERIA "EL AHORRO" / MULTICENTRO / VIDEO CLUB CASSAL • **CHILLAN:** CASA EDISON • **CONCEPCION:** COOPERCARAB / DISMAR / DISMAR 2 / EQUUS / PHANTER / RAPSODIA / SESCO • **LOS ANGELES:** DISTRIBUIDORA MERINO • **ANGOL:** SCORPIO • **VICTORIA:** CASA SIGMUND • **TEMUCO:** COMERCIAL MANQUEHUE / ESTABLECIMIENTOS GEJMAN / FALABELLA • **PUCON:** EL TIT • **VILLARRICA:** JOYERIA KETTERER • **VALDIVIA:** ELECTROMUSICA • **LA UNION:** IMPORTADORA COSMOS • **OSORNO:** CASA REAL / FOTO EXPRESS • **PUERTO VARAS:** ELECTRO HORN • **PUERTO MONTT:** COMERCIAL MANQUEHUE / DIMARSA • **COYHAIQUE:** FACI HOGAR • **PUNTA ARENAS:** BALFER LTDA.

QuickShot[®]

by Bondwell

REPRESENTANTE EXCLUSIVO PARA CHILE

LUIS A. VERDUGO AGUILERA

MONJITAS: 843 OF. 911 - C

TELEFONOS: 339219 - 6321966

FAX: 02-395977

TELEX: 242283 VCF CL

SANTIAGO - CHILE



■ QS-16



□ QS-119



■ QS-120



■ QS-127
INALAMBRICO

QuickShot[®]
20,000,000
JOYSTICKS SOLD WORLDWIDE



■ QS-125



■ QS-129F



■ QS-1



■ QS-II TURBO



■ QS-130F



APPLE
QS-X



■ QS-131

- Atari y Commodore. ■ Sega, Atari, Commodore, Amstrad y MSX
- IBM PC/XT/AT. □ IBM PC/XT/AT y Apple IIe/c
- Sega, Sears, Atari, Commodore, MSX y Nintendo Entertainment System

DISTRIBUIDORES:

SANTIAGO: A. PARIS • FALABELLA • CASA ROYAL • SUPERTIENDAS ABC • DISTRIBUIDORA DELANO • RIPLEY • HITES • IMACO • ESTADO OFERTA • TELEAUDIO • COMPUMANQUE • COMPUCENTER • MICROCENTRO • INFOLAND-E. YAÑEZ • **VIÑA DEL MAR:** INSIS • SELECTRONIC • SUPERTIENDAS ABC • FALABELLA • RIPLEY • **VALPARAISO:** SELECTRONIC • COMPUTRONIC • SUPERTIENDAS ABC • **TEMUCO:** FALABELLA • SUPERTIENDAS ABC • **CONCEPCION:** FALABELLA • SUPERTIENDAS ABC • RIPLEY • SERV. COMPUTACIONAL CONCEPCION LTDA. • CRECIC S.A. • **VALDIVIA:** DOMBURGO.