



APPLE II

Guia do Operador
APPLEII Plus e APPLEIIe

SEGUNDA EDIÇÃO



COMANDOS BÁSICOS



OSBORNE/ McGRAW-HILL

ZELDA GIFFORD

Guia de Referência — Apple II

CIP-Brasil. Catalogação-na-Publicação
Câmara Brasileira do Livro, SP

Gifford, Zelda.
G338g Guia de referência-Apple II / Zelda Gifford ; tradução José
Ildeberto Barros, Nelson Luiz F. Barros. -- São Paulo : McGraw-Hill
do Brasil, 1984.

1. Apple II (Computador) -- Programação I. Título. II. Título:
Apple II.

83-2112

17. CDD-651.8

18. -001.642

Índices para catálogo sistemático:

1. Apple II : Computadores eletrônicos : Programação :
Processamento de dados 651.8 (17) 001.642 (18.)
2. Programação : Apple II : Computadores eletrônicos :
Processamento de dados 651.8 (17.) 001.642 (18.)

Guia de Referência — Apple II

Zelda Gifford

Tradução:

José Ildeberto Barros

Nelson L. F. Barros

McGRAW-HILL

*São Paulo • Rio de Janeiro • Lisboa • Porto • Bogotá • Buenos Aires • Guatemala
• Madrid • México • New York • Panamá • San Juan • Santiago*

*Auckland • Hamburg • Johannesburg • Kuala Lumpur • London • Montreal •
New Delhi • Paris • Singapore • Sydney • Tokyo • Toronto*

Do original

Apple II Diskguide

Copyright © 1983 by McGraw-Hill, Inc.

Copyright © 1984 da Editora McGraw-Hill do
Brasil, Ltda.

Todos os direitos para a língua portuguesa reservados
pela Editora McGraw-Hill do Brasil, Ltda.

Nenhuma parte desta publicação poderá ser reprodu-
zida, guardada pelo sistema “retrieval” ou transmitida
de qualquer modo ou por qualquer outro meio, seja
este eletrônico, mecânico, de fotocópia, de gravação,
ou outros, sem prévia autorização, por escrito, da
Editora.

Conselho Editorial:

Prof. Gastão de Almeida Rocha (UERJ)

Prof. Oscar Benedicto Júnior (FATEC)

Prof. João José Neto (USP)

Prof. Arnaldo Milstein Mefano (UERJ)

Eng^o Paulo Borelli

Editor: José Martins Braga

Supervisora de Produção: Maria Celina Jurado

SUMÁRIO

<i>Como Usar Este Guia</i> , 1
Comandos de edição de linha, 2
BASIC,
Elementos do BASIC, 6
Ambiente de programação, 7
Declarativos de programação, 9
Comandos de depuração, 15
Funções, 16
Funções definidas pelo usuário no Applesoft, 18
Operadores de precedência, 19
Exibição de texto, 20
Gráficos, 22
Tabelas de formato, 26
<i>DOS</i> , 33
Convenções e <i>defaults</i> (alternativos), 34
Usando discos e arquivos, 36
Ferramentas de programação, 38
Comandos binários de disco, 39
Programas utilitários, 40
Dispositivos de E/S, 42
Criando e usando arquivos de texto, 44
Arquivos EXEC, 47
<i>O Programa Monitor</i> , 48
Comandos do monitor, 49
Usando registradores, 49
<i>Posições úteis PEEK e POKE</i> , 52
<i>Códigos de caracteres ASCII</i> , 56
<i>Índice analítico</i> , 57

COMO USAR ESTE GUIA

- Para corrigir erros de digitação, veja Comandos de edição de linha, p. 2
- Para iniciar um programa, veja (RUN), p. 8, 39
- Para programação de gráficos, veja p. 22
- Para usar cassetes, veja Comandos de cassete, p. 30
- Para usar o Sistema de Operação em Disco (DOS), veja p. 33
- Para formatar um disco, veja (INIT), p. 36, 37
- Para copiar um disco, use (COPY), p. 41
- Para códigos de erro, veja Posições úteis PEEK e POKE, p. 52

Este guia abrange o Applesoft e o Integer BASIC em todos os modelos dos computadores Apple II e versão 3.3 do DOS.

pgm	Indica um comando que é emitido somente a partir de um programa
imm	Indica um comando que é emitido somente a partir do teclado em modo imediat o; um comando ou declarativo sem a notação (pgm) ou (imm) pode ser usado tanto em modo de programa quanto em modo imediato

- IntB** Indica que um conceito ou declarativo é válido somente no **integer BASIC** e não-válido no Applesoft BASIC; o Prompt do Integer BASIC é “>”
- AppB** Indica que um conceito ou declarativo é válido somente no **Applesoft BASIC** e não-válido no Integer BASIC; o prompt do Applesoft BASIC é “]”; entradas sem (IntB) ou (AppB) são disponíveis em ambas as linguagens
- imm in IntB** Indica o modo imediato no Integer BASIC
- imm,pgm in AppB** Indica os modos imediato e de programa no Applesoft
- ital* Indica o valor que você fornece.
Exemplos usam valores típicos (ex.: um número de slot válido tipifica todos os números de slot)
- RETURN** Indica a chave RETURN

COMANDOS DE EDIÇÃO DE LINHA

- Move o cursor para a direita na linha de exibição; copia cada caractere na memória (nenhuma mudança para a exibição)

- ← **Retorna o cursor**, apaga o caractere da memória mas não da exibição
- ↑ Move o **cursor para cima** uma linha (somente no Apple IIe)
- ↓ Move o **cursor para baixo** uma linha (somente no Apple IIe)
- CTRL-X** Cancela uma linha; move o cursor para a próxima linha
- REPT** **Repete o caractere** enquanto ambas as chaves estão pressionadas (pressione o caractere-chave e então pressione a chave REPT)
- 25 RETURN** Deleta a linha 25 do programa
- 25 END RETURN** Acrescenta o declarativo *END* como linha 25 do programa

Seqüência de Duas Chaves

Pressione a tecla ESC e solte-a. Então, pressione a segunda chave.

- ESC A** Move o **cursor para a direita**, uma posição
- ESC B** Move o **cursor para a esquerda**; uma posição
- ESC C** Move o **cursor para baixo**, uma linha
- ESC D** Move o **cursor para cima**, uma linha
- ESC E** **Deleta** (apaga) caracteres a partir do cursor até o fim da linha de exibição

- ESC F** Deleta caracteres a partir do cursor até o fim da tela
- ESC @** Limpa a tela, move o cursor para o canto superior esquerdo

Inserindo Caracteres

Use a seqüência ESC de duas chaves. Mova o cursor para o início da linha para receber novo texto e siga os seguintes passos:

- Copia os caracteres corretos
- ESC D** Move o cursor para cima uma linha
- ESC E** Apaga o texto à direita na nova linha; digite a inserção
- ESC B** Move o cursor para trás até o primeiro caractere após a inserção
- ESC C** Abaixa o cursor para pegar o resto da linha original
- Copia o resto da linha

Seqüência de Chave Única

Para Apples com monitor Autostart, entre no modo de edição para controle do cursor de chave única.

- ESC** Entra no modo de edição
- I** Move o cursor para cima, uma linha

J	Move o cursor para a esquerda, uma posição
K	Move o cursor para a direita, uma posição
M	Move o cursor para baixo, uma linha
A	Sai do modo de edição (aperte qualquer chave exceto I, J, K, M, REPT, CTRL ou SHIFT)

BASIC

Elementos do BASIC

String

Um caractere ou seqüência de caracteres (0-255); contido entre aspas; uma expressão string pode ser uma seqüência constante, variável ou expressão

Inteiro

Não tem ponto decimal; faixa de valor -32767 até 32767

Número Real

Pode ter ponto decimal e parte fracionária; notação científica tem a forma *número* $E \pm ee$, onde *número* contém os dígitos significativos do número; se não aparece ponto decimal, este é assumido ser à direita de *número*; o

sinal + ou - é opcional; *ee* é um *expoente de um ou dois dígitos, especificando a magnitude de número*; limite de variação da notação E: -1E+38 até +1E+38 (AppB)

Variável

Um símbolo que representa uma localização na memória, um lugar onde um valor pode ser armazenado; IntB aceita variáveis inteiras e string; AppB aceita variáveis reais, inteiras e string

Nome de Variável

Tem de um a cem caracteres; o primeiro é uma letra; o último é \$ para um string (dimensionado antes de usar); variável numérica prescinde do \$ final; (IntB) tem de 1 a 238 caracteres; primeiro uma letra e por último o tipo (\$ para um string, % para um inteiro, branco para real) (AppB)

Expressão

Combina variáveis e operadores (AppB aceita qualquer constante, variável ou expressão, tanto numérica quanto string, relacional ou booleana; ou qualquer combinação válida disso)

Expressão string

Pode ser uma seqüência constante, variável ou expressão

Expressão numérica

Qualquer constante numérica, variável ou expressão

Array

Um tipo de variável usada para representar uma lista de valores chamada arranjo de elementos ligados pelo mesmo nome; IntB aceita somente arrays numéricos, que devem ser dimensionados antes do uso; arrays AppB podem ser variáveis inteiras, reais ou string, de dimensões até 88; arrays até 11 elementos não têm de ser dimensionados primeiro

Constante

Um valor que permanece sem mudar durante a execução de um programa; pode ser numérico ou string

Ambiente de Programação

AUTO 500,5 Fixa numeração automática de linha, começando com 500 e variando de 5 em 5 (default = 10); CTRL-X apaga a linha (IntB)

CLEAR Atribui 0 a variáveis numéricas e elementos do array; atribui nulo a strings variáveis e aos elementos do array (AppB)

CLR	Atribui 0 a todas as variáveis numéricas e elementos do array e um valor nulo aos strings (IntB)
DEL 5,7	Suprime as linhas de 5 a 7, inclusive; o programa Applesoft dá uma pausa após tal supressão (imm in IntB) (imm, pgm in AppB)
FP	Muda para Applesoft BASIC (ponto flutuante); restaura a memória (imm); para trazer Applesoft do disco, veja FP sob DOS, p. 38
INT	Muda para Integer BASIC , restaura a memória (imm)
IN# 4	Capta a entrada a partir do dispositivo conectado através do slot 4
IN#0	Capta entrada a partir do teclado
PR#2	Envia saída para o dispositivo conectado através do slot 2
PR#0	Envia saída para a tela
MAN	Desliga a numeração automática de linha ; retorna ao manual (IntB)
NEW	Suprime programa e variáveis da memória; restaura LOMEM (imm in IntB) (imm, pgm in AppB)
RUN 5	Inicia a execução de programa na linha 5 (default = primeira linha) (IntB: número da linha pode ser uma expressão

numérica) (imm in IntB) (imm, pgm
in AppB)

Declarativos de Programação

END Faz parar um programa (requerido em
IntB)

REM SUBR STARTS HERE

Coloca o comentário *SUBR STARTS
HERE* no programa

Armazenando e lendo valores

DIM R\$(5),I(9) Reserva espaço para um string
array unidimensional *R\$*, e um array
numérico unidimensional *I(9)*; numera
de 0 a 5 os elementos de um string
array na memória; numera de 0 a 9
os elementos de um array numérico
na memória; faixa subscrita: 1-255;
arrays múltiplos são opcionais (IntB)

DIM I(6,2),A\$(10,5),B%(7)

Reserva espaço para um array numérico
bidimensional de 6 por 2, *I(6,2)*, um
string array de 10 por 5, *A\$(10,5)*,
e array de inteiros unidimensional de
8 elementos, *B%(7)*; default subscrito =
= 10 (Appb)

$O=5 * R ^ 2$ Atribui o valor da expressão numérica $5 * R ^ 2$ a Q (LET $Q=5 * R ^ 2$ opcional)

POKE 16383,5 Armazena um byte de dados, 5, na localização 16383 da memória

DATA 1.3,"MAY 1"

Especifica um valor numérico 1.3 e um valor de string *MAY 1* para READ; para incluir aspas no string, use CHR\$(34) (AppB)

READ J, D\$ Atribui o valor seguinte do declarativo DATA à variável(is) listadas(s) $J, D\$$; variável(is) deve(m) combinar com o tipo de valor de DATA (AppB)

RESTORE Retorna ao primeiro valor no primeiro declarativo DATA, para posterior READ (AppB)

INPUT"ENTER DATA", A, B\$

Requisita entrada do teclado para a variável inteira A e variável string $B\$$; exibe *ENTER DATA* somente para A ; subseqüentemente exibe um ? somente para variáveis inteiras; o prompt do string é opcional (pgm, IntB)

INPUT"ENTER DATA"; A, B\$

Requisita entrada do teclado para a variável inteira A e variável string $B\$$;

- exibe *ENTER DATA* somente para *A*;
exibe um ? para variáveis sucessivas;
múltiplas variáveis podem aparecer
numa linha (pgm, AppB)
- GET B\$** Aceita um caractere a partir do teclado
e o atribui ao string variável *B\$*; CTRL-@
atribuirá nulo; RETURN não é
necessário (pgm, AppB)

Exibindo e Imprimindo

- LIST 5** Exibe a linha 5 do programa (default =
= o programa inteiro)
- LIST 60,74** Exibe as linhas 60 a 74, inclusive, do
programa (AppB aceita hífen ou
vírgula)
- LIST 60,** Exibe da linha 60 até a última linha
(AppB)
- LIST ,74** Exibe da primeira linha até a linha 74
(AppB)
- PRINT X** Dá saída ao valor de *X* para a tela ou
outro dispositivo de saída (default=uma
nova linha se nenhuma variável é
fornecida); AppB aceita ? para PRINT
- PRINT P\$,G,** Imprime as variáveis *P\$* e *G* nas
posições AppB 1 e 17; considera até
33 (por causa da vírgula final); substitua

a vírgula final por ponto e vírgula para suprimir o retorno do carro (IntB imprime em 1 e 9; considera até 17)

PRINT "SNAP"; "SHOT"

Imprime *SNAPSHOT* (nenhum ponto e vírgula é necessário no AppB)

Loops e Desvios

CALL 10 Desvia para a sub-rotina de linguagem de máquina na posição *10* da memória

FOR J=J1 TO J2 STEP J3

Executa um loop de declarativos para baixo até o declarativo **NEXT**, incrementando *J* de *J3* cada vez, até que *J* seja igual a *J2*; atribui o valor de *J1* a *J* quando o primeiro é executado (default=incremento=1) (IntB=variáveis inteiras; AppB=variáveis reais; aninhando acima de 10 níveis no AppB; acima de 16 no IntB) (pgm, IntB)

NEXT J,K Termina o loop iniciado por um **FOR**, incrementa *J* como definido no **FOR**; múltiplos loops variáveis **FOR** são permitidos; *K* (pgm, IntB)

GOSUB 2*M Desvia para o número da linha=valor de *2*M* (IntB)

GOTO A1 Desvia para o número da linha = valor de *A1* (IntB)

GOTO 99 Desvia para a linha 99

IF J=5 THEN GOTO 45

Desvia para a linha 45 se $J=5$ for verdadeiro (default = próximo declarativo do programa); GOTO é opcional (IntB)

**IF J=5 THEN
GOTO } 35
THEN GOTO }**

Desvia para a linha 35 se $J=5$ for verdadeiro (default = próximo declarativo do programa); a expressão pode ser relacional, string, expressão string ou numérica (AppB)

**IF J THEN
GOTO } 35
THEN GOTO }**

Desvia para a linha 35; se $J=0$ (AppB)

IF J=5 THEN PRINT Z

Executa o declarativo *PRINT Z* se $J=5$ for verdadeiro; se falso, executa o próximo declarativo do programa (IntB)

IF J=5 THEN PRINT X:PRINT Z

Executa os declarativos subsequentes *PRINT X*, *PRINT Z* se $J=5$ for verdadeiro; se falso, executa o próximo declarativo do programa; a expressão

pode ser relacional, string, expressão string ou numérica (AppB)

ON J GOTO 50,60,70

Vai para a linha 50 se o valor de $J=1$; linha 60 se 2 etc.; executa a próxima linha se $J=0$ ou se $J >$ número da linha final; J =constante numérica, variável ou expressão (AppB)

ON K GOSUB 77,88,99

Chama a sub-rotina na linha 77 se $K=1$; na linha 88 se $K=2$ etc.; executa a próxima linha se $K=0$; faixa de $K:0-255$ (AppB)

RETURN Retorna para o declarativo seguinte ao último GOSUB executado

POP Esquece a posição de retorno para o GOSUB mais recente

ONERR GOTO 99

Desvia para a rotina de erro na linha 99 quando ocorre um erro; PEEK(222) exhibe o código do erro; ver Posições úteis PEEK e POKE (pgm, AppB)

RESUME Recomeça a execução no início do declarativo no qual ocorreu um erro (Pgm, AppB)

Comandos de Depuração

CON	Recomeça a execução do programa no próximo declarativo após uma parada CTRL-C (imm, IntB)
CONT	Recomeça a execução do programa no próximo declarativo após um STOP, CTRL-C ou END (AppB)
DSP <i>Q</i>	Exibe <i>Q</i> ou seu número de linha sempre que o valor de <i>Q</i> muda; cancelado por RUN; CON ou GOTO continua o programa (IntB)
NODSP <i>Q</i>	Cancela o efeito do DSP para <i>Q</i> (IntB)
STOP	Faz parar a execução do programa (AppB)
TRACE	Exibe os números das linhas dos declarativos como eles são executados
NOTRACE	Desliga o modo de rastreamento iniciado por TRACE
WAIT <i>\$3000, MASK, M</i>	Faz parar um programa até que a posição <i>\$3000</i> atinja o valor binário de <i>M</i> (default=0); <i>MASK</i> determina quais bits comparar; CTRL-RESET interrompe o WAIT (AppB)

Funções

Funções substituem variáveis ou constantes em qualquer lugar num declarativo BASIC, exceto à esquerda de um sinal de igualdade. Exemplos:

```
L=LEN(D$) PRINT SQR(A)
IF LOG(X)<0 GOTO T
```

- ABS *B*** Determina o valor absoluto (sem sinal) da variável *B*
- ASC ("A")** Converte o string "A" em 65, seu código ASCII
- ATN (*THETA*)** Calcula o arco-tangente do ângulo *THETA* em radianos (faixa = $-\pi/2$ à $\pi/2$) (AppB)
- CHR\$ (69)** Fornece o caractere E do ASCII para 69, uma expressão numérica (ver Códigos de caractere ASCII) (AppB)
- COS (*THETA*)** Calcula o co-seno do ângulo *THETA* em radianos (AppB)
- EXP (3)** Eleva e (2.718282) à terceira potência (AppB)
- FRE (0)** Informa quantos bytes estão disponíveis para um programa abaixo do espaço de memória do string e acima do espaço de memória do array; se um valor

negativo é informado, some 65536 a ele para o cálculo (AppB)

INT (2.5) Fornece o maior inteiro igual ou menor que 2.5 (AppB)

LEFT\$ ("TOOTH", 2)

Extrai *TO*, os dois caracteres mais a esquerda do string *TOOTH* (AppB)

LEN ("AB C") Informa 4 caracteres como o tamanho de uma expressão string

LOG (900) Fornece o logaritmo natural de 900 (AppB)

MID\$ ("PEARL", 2,3)

Extrai os três caracteres do meio, *EAR* (default = fim do string), iniciando com o *segundo* caractere (AppB)

PDL (PAD1) Informa o valor do controlador do jogo (faixa: 0-255), *PAD1*; o valor é baseado na rotação do número *PAD1* do controle manual (PADDLE) ou na resistência do dispositivo conectado ao soquete do controle do jogo *PAD1* (faixa de valores = 1-3)

PEEK (200) Exibe o conteúdo (decimal) da posição 200 da memória

RIGHT\$ ("PHONE", 3)

Extrai *ONE*, os três caracteres mais à direita do string *PHONE* (AppB)

RND (GEN) Fornece um número randômico

SGN (N) Informa o sinal da variável N (+1 = positivo; -1 = negativo; 0 = zero)

SIN (THETA) Calcula o seno do ângulo THETA em radianos (AppB)

SQR (J) Retorna a raiz quadrada de J , um número positivo (AppB)

STR\$ (14) Converte o numérico 14 para o string "14" (AppB)

TAN (THETA) Retorna a tangente do ângulo THETA em radianos (AppB)

VAL ("198") Converte o valor do string "198" para o numérico 198 (AppB)

Funções Definidas pelo Usuário no Applesoft

DEF FN FUN (DUMMY) = X ^ 2
 Define uma função de propósito especial numa linha (pgm)

X = FN FUN (DUM)
 Invoca a função chamada *FUN* (definida no DEFFN acima); fornece ao *DUM* o valor de $X ^ 2$ e avalia o resultado

Z = USR (A + B)
 Guarda o valor de 8 bytes da expressão aritmética $A + B$ na posição \$9D-\$A3 da memória (o "acumulador de ponto

Operadores e Precedência

<i>Operadores</i>	<i>Precedência</i>	<i>Integer</i>	<i>Applesoft</i>	<i>Função</i>	<i>Exemplo</i>
<i>Operadores Aritméticos</i>	9	()	()	Ordem de avaliação	$3 \times (4 + 5) = 27$
	8	^	^	Exponenciação	$3^4 + 5 = 17$
	7	-	-	Unary negativo	$3^2 = 9$
	6	*	*	Multiplicação	$-3 = -3$
	6	/	/	Divisão	$6 * 2 = 12$
	6	MOD	/	Resto da divisão	$6 / 3 = 2$
	5	+	n.d.†	Adição	$9 \text{MOD} 6 = 3$
	5	-	+	Subtração	$3 + 6 = 9$
	4	=	-	Igualdade	$3 - 6 = -3$
	4	# ou < >	=	Diferente de A	$2 = 3$ falso
<i>Operadores Relacionais</i>	4	<	<	Menor que	$2 < 3$ VERDADEIRO
	4	>	>	Maior que	$2 > 3$ FALSO
	4	<=	<= ou =<	Menor ou igual	$2 <= 3$ VERDADEIRO
	4	>=	>= ou =>	Maior ou igual	$2 >= 3$ FALSO
	3	NOT	NOT	Complemento lógico	NOT (7) >= 6 VERDADEIRO
	2	AND	AND	AND lógico	5 AND (A\$ = B\$)
<i>Operadores Booleanos</i>	1	OR	OR	OR lógico	VERDADEIRO SE A\$ = B\$ ("A" = "B") OR (16 = 2 ^ 5) FALSO

flutuante”) e desvia para a posição 10 (\$0A) da memória, a qual contém um JMP para a sub-rotina de linguagem assembly

Relações verdadeiras retornam o valor lógico 1; relações falsas retornam o valor lógico “0”. (IntB = strings podem ser comparadas somente com =, #, ou <>. AppB = inteiro e real podem estar presentes na mesma expressão real, relacional ou booleana; em expressões mistas, o BASIC converte as expressões booleanas primeiro).

Exibição de Texto

A tela de exibição permite até 24 linhas por 40 colunas com a linha 1 no alto e a coluna 1 na esquerda (ver janela de texto e controle do cursor sob Posições úteis PEEK e POKE para valores permitidos).

Controle do Cursor

HOME	Limpa a tela e move o cursor para o canto superior esquerdo da janela de texto (AppB) (IntB usa CALL-936)
HTAB (3)	Põe o cursor na coluna 3 da linha corrente (AppB)

C=PEEK(36)	EXAMINA a posição 36 e informa na posição do cursor na coluna corrente (IntB)
R=PEEK(37)	EXAMINA a posição 37 e informa na posição do cursor na linha corrente (IntB)
Y=POS(DUM)	Salva na variável Y a posição do cursor na coluna , faixa = 0, mais à esquerda até 39; <i>DUM</i> é um pseudo-argumento (AppB)
PRINT SPC(5)	Imprime 5 espaços em branco (AppB)
TAB 40	Põe o cursor na coluna 40 na linha corrente (IntB)
PRINT TAB(Q)	Apaga caracteres até a coluna <i>Q</i> (faixa=1-255) (AppB)
TEXT	Move o cursor para a última linha da tela e retorna para modo de texto em tela completa sem limpar a tela
VTAB(24)	Move o cursor para a linha 24 na coluna corrente (AppB)

Efeitos Especiais

INVERSE	Liga o vídeo inverso (preto no branco) (AppB)
FLASH	Liga o vídeo cintilante (AppB)

NORMAL	Retorna dos modos de vídeo cintilante e inverso (AppB)
SPEED (2)	Fixa a velocidade de exibição do caractere em 2; faixa: 0 (a mais lenta) até 255 (AppB)

Janela de texto

Este exemplo mostra como alterar a janela de texto

```

1010T =10:W=20:LM=11:B=13 Fixa o topo 10,
    largura 20, margens esquerdas 11 & fundo 13
1020 CALL-936 Limpa a tela
1030 POKE 32,T: POKE 33,W: POKE 34,LM:
    POKE 35,B Fixa para input
1040 VTAB T+2 Move o cursor dentro da janela
    de texto
1050 INPUT M1 Dá entrada no valor de M1 para
    dentro dos limites da janela
    de texto
1060 POKE 32,0 : POKE 33,40 Restaura a
1070 POKE 34,0 : POKE 35,24 tela integral
1080 VTAB 23 Move o cursor para a linha
    mais baixa

```

Gráficos

GR Converte a tela para baixa resolução (40X 40); para tela completa (40X 48), use POKE-16301

- HGR** Converte a tela para **alta resolução** (280×160); use POKE-16302,0 para mudar para tela completa de alta resolução; use POKE -16301,0 para restaurar a janela de texto; fixe LOMEM em 16384 e HIMEM em 8192 com 32K de memória e DOS presente (AppB)
- HGR2** Converte a tela em **tela completa, alta resolução** (280×192); fixa LOMEM em 24576, HIMEM em 16384 com 48K (AppB)
- HIMEM:8192** Fixa o **limite superior da memória** em 8192 para programas e variáveis (faixa = -65535 a +65535; -32767 a 32767 (IntB) (imm in IntB) (imm, pgm in AppB); para o valor corrente do HIMEM, use PEEK(116)*256 + PEEK(115) AppB;
PEEK(77) *256 + PEEK(76) IntB
- LOMEM:2048** Fixa o **limite inferior da memória** em 2048 para variáveis e programas (faixa = -65535 a +65535; -32767 a 32767 (IntB) (imm in IntB) (imm, pgm in AppB); para o LOMEM corrente, use PEEK(106)*256 + PEEK(105)

Pontos e Linhas

HLIN 5, 9 AT 8 Desenha uma **linha horizontal** da quinta à nona coluna na oitava linha na tela de baixa resolução na cor corrente

HPLOT 25, 5 Põe um **ponto** na coluna 25 (faixa=0-279) e linha 5 (faixa=0-191) na tela na cor corrente (HCOLOR) (AppB)

HPLOT TO 5,8 Desenha uma **linha** do último **ponto marcado** até a coluna 5, linha 8 na cor corrente (AppB)

HPLOT 20,35 TO 100,100

Desenha uma **linha** entre as **coordenadas** 20,35 e 100,100, na cor corrente (AppB)

PLOT 15,7 Põe um **ponto** na coluna 15 [faixa=0 (esquerda) a 39] na linha 7 [faixa=0 (topo) a 47] na tela de baixa resolução na cor corrente

Exibindo um caractere na tela em modo de texto ou na janela de texto

Primeiro, encontre o caractere na tabela. Leia acima e a esquerda para encontrar as cores para marcar o caractere.

Modo de vídeo			Caractere															
Inverso	Cintilante	Normal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	8 12	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	5	9 13	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	-
2	6	10 14		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	7	11 15	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

Para exibir um sinal de porcentagem (%) cintilante, na coluna 15, linha 40, ver tabela precedente e digite:

COLOR = 5

COLOR = 6

PLOT = 15,40

PLOT = 15,41

VLIN 7,35 AT 20

Desenha uma linha vertical da linha 7 à linha 35 na coluna 20 na tela na cor corrente, em baixa resolução

Cor

COLOR=15 Fixa a cor *branca* (ver a tabela seguinte de Códigos de cor) para marcar e desenhar com PLOT, VLIN e HLIN em baixa resolução; uma expressão maior que 15 repete a cor na tabela (ex.: 0 e 16 são pretos)

HCOLOR=5 Fixa a cor *laranja* (ver tabela de Códigos de cor) para marcar e desenhar com HPLOT e DRAW em alta resolução (AppB)

PRINT SCR_N (38,24)

Comunica o código de cor do ponto de baixa resolução na coluna 38 linha 24

$$C = \text{SCR}_N(38,24) + 16 * \text{SCR}_N(38,25)$$

Determina o código ASCII do caractere na coluna 38, linha 24; para o próprio caractere, use CHR\$(C)

Tabela de Códigos de Cor

<i>Cores</i>	<i>Baixa Resolução</i>	<i>Alta Resolução</i>		<i>Cores</i>	<i>Baixa Resolução</i>	<i>Alta Resolução</i>	<i>XDRAW</i>
Preto	0	0 e 4	Branca	Marron	8	n. d.	
Margenta	1	n. d.		Laranja	9	5	Azul
Azul escuro	2	n. d.		Cinza #2	10	n. d.	
Púrpura	3	2	Verde	Rosa	11	n. d.	
Verde escuro	4	1	Violeta	Verde claro	12	n. d.	
Cinza #1	5	n. d.		Amarelo	13	n. d.	
Azul médio	6	6	Laranja	Água	14	n. d.	
Azul claro	7	n. d.	Branca	Branco	15	3 e 7	Preto

Tabelas de Formato

Criando um formato

Primeiro desenhe o formato num papel quadriculado. Use uma seta em outra cor para indicar em que caminho o cursor (vetores) se moverá. Esquematize as linhas exibidas como setas sólidas e o movimento do cursor não exibido como setas pontilhadas (--->). Marque cada vetor com seu código binário.

<i>Vetor</i>	<i>Ação</i>	<i>Binário</i>	<i>Vetores para um formato quadrado</i>
↑	Mover para cima; não marcar	000	<p>Binário: 101</p> <p>Binário: 100</p> <p>Binário: 110</p> <p>Binário: 111</p>
→	Mover para direita; não marcar	001	
↓	Mover para baixo; não marcar	010	
←	Mover para esquerda; não marcar	011	
↑	Mover para cima e marcar	100	
→	Mover para direita e marcar	101	
↓	Mover para baixo e marcar	110	
←	Mover para esquerda e marcar	111	

Definindo um formato

Ordene os códigos binários para cada vetor, transportando o código do vetor de marcação de 3 bits em bytes de 8 bits que podem ser armazenados na memória. Converta cada byte em hexa; termine a definição do formato com um byte de 0. Por exemplo: para codificar o formato quadrado ilustrado acima:

		<i>Vetores</i>	<i>8 Bit</i>	<i>Binário</i>	<i>Código</i>	<i>Código Hexa</i>
Byte 0 Nada	↑	→	00	100	101	25
Byte 1 Nada	↓	←	00	110	111	37
Byte 2 Nada	Nada	Nada	00	000	000	00

Organizando a Tabela de Formato na memória

Uma tabela de formato consiste num diretório que aponta para os formatos individuais na tabela, seguido pelas definições reais de formatos. Esta é uma tabela para dois formatos.

<i>Formato da Tabela de Formato</i>		
<i>Byte</i>	<i>Conteúdo</i>	<i>Descrição</i>
0	02	Número de formatos na tabela (máximo = 255)
1	00	Sempre 0
2	06	Endereço compensado do formato 1 a partir do byte 0 (Primeiro a ordem mais baixa)
3	00	
4	09	Endereço compensado do formato 2 a partir do byte 0
5	00	
6	25	Definição do formato para quadrado (hexa)
7	37	
8	00	Formato termina com 00
9	2E	
10	35	
11	2E	
12	3F	Definição #2 de formato
13	3C	
14	24	
15	00	Formato termina com 00

Entrando com a Tabela de Formato na memória

1. Calcule o tamanho da tabela de formato em bytes, incluindo o diretório.
2. Restaure HIMEM para permitir espaço para a tabela de formato. Restaure HIMEM abaixo do endereço de início do DOS ou antes da página de alta resolução que você quer usar. Para calcular o novo HIMEM, use: `PRINT PEEK(116)*256+`

PEEK(115)-X, onde é o tamanho da tabela de formato incluindo o diretório.

3. Fixe HIMEM para este valor computado
4. Coloque a tabela de formato na memória. Use ou uma série de declarativos BASIC POKE ou comandos do monitor.
5. Salve a tabela de formatos num disco com BSAVE ou numa fita.

Este exemplo grava na memória uma tabela de formato contendo a definição de um quadrado.

POKE 768,01 Número de formatos na tabela
POKE 769,00 Preenche o byte não usado
POKE 770,04 Introduz endereço de compensação do formato 1 a partir do byte 0, primeiro byte
POKE 771,00 Introduz seu segundo byte
POKE 772,37 Introduz a definição de formato, primeiro byte
POKE 773,55 Introduz a definição de formato, segundo byte
POKE 774,00 Introduz 00 para terminar o formato

Este programa em hexa grava a mesma tabela de formato na memória através do monitor

6000: 01 00 04 00 Introduz o diretório
 : 25 37 00 Introduz o quadrado

Salvando a Tabela de Formato em disco

1. Calcule o tamanho da tabela de formato
2. Digite BSAVE SQUARE, A\$6000, L7 (grava 7 bytes, iniciando no endereço \$6000). O arquivo é chamado SQUARE.

Carregando uma Tabela de Formato a partir do disco

1. Fixe HIMEM
2. Digite BLOAD SQUARE, A\$3000. Carregue o arquivo SQUARE na memória iniciando em \$3000. Se o endereço é omitido, use o endereço de BSAVE.
3. Carregue o endereço da tabela de formato em 232 e 243.

Comandos do cassete

- LOAD** Carrega um programa do cassete
- RECALL AR** Recupera um array numérico, *AR*, da fita; dimensione o ARRAY antes de usar RECALL (AppB)
- SAVE** Salva o programa que está correntemente na memória numa fita cassete; o segundo sinal beep significa que o salvamento foi feito; pressione o botão RECORD antes de digitar SAVE (imm, IntB), (imm, pgm in AppB)

- SHLOAD** Carrega uma tabela de formato de alta resolução da fita para a memória e ajusta o HIMEM (AppB)
- STORE AR** Salva o array numérico, *AR*, na fita; para armazenar string arrays, converta com ASC primeiro (AppB)

Salvando uma Tabela de Formato em fita

1. Calcule o tamanho da tabela de formato.
2. Introduza o monitor. Iniciando no endereço 0 da memória, introduza o tamanho da tabela como um número hexa de 2 bytes: *00:07 00.
3. Reenrole a fita até o início. Pressione o botão RECORD. Grave na fita: *0.1W 6000.6006 W (grave o tamanho da tabela de formato e os 7 bytes da tabela de formato exemplificada, do endereço \$6000 até \$6006).

Carregando uma Tabela de Formato da fita

1. Reenrole a fita.
2. Digite SHLOAD.

Comandos de Formato

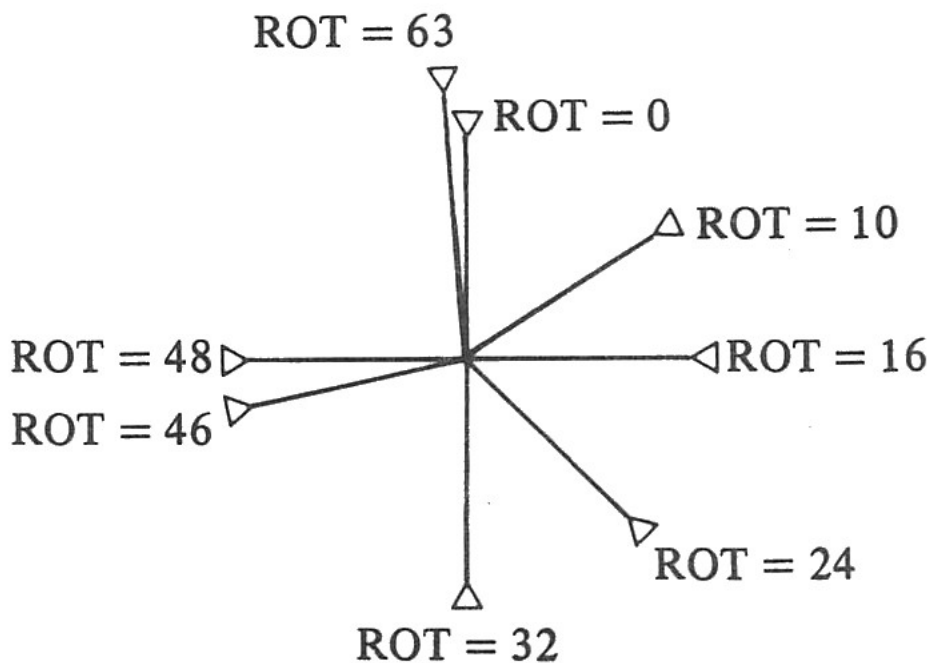
DRAW 5 AT 40,24

Desenha o formato 5 de alta resolução (faixa = 0-255, um formato numa

tabela de formato) na tela em 40,24 (default=último ponto marcado), na cor corrente; use SCALE e ROT antes de DRAW (AppB)

ROT= 32 Fixa em 32 a rotação de qualquer formato desenhado subseqüentemente; se SCALE=1, a expressão pode ser um destes quatro valores: 0,16, 32 ou 48; SCALE=2 permite um em oito valores; SCALE=3 permite um em 16 valores; SCALE=5 ou maior permite todos os 64 valores (AppB)

Rotação de Formas (SHAPE)



SCALE=2 Duplica o tamanho dos formatos subseqüentes de alta resolução desenhados através de DRAW ou XDRAW (faixa = 0-255; 1 desenha o formato como definido) (AppB)

XDRAW 5 AT 48,24

Desenha o formato 5 de alta resolução na tela, na coluna 48, linha 24, em complemento da cor na tela (ver a Tabela de códigos de cor, p. 26) (AppB)

DOS

DOS organiza a superfície do disco em 35 trilhas (16 setores por trilha). Um setor armazena até 256 bytes; 496 setores estão disponíveis ao usuário. Para usar o DOS, ponha seu cartão de controle de disco no slot 6 e conecte seu drive primário à conexão drive #1 naquele cartão.

Partidas frias

Coloque um disco DOS no drive #1

- Ligue a chave de energia. Isto inicia o DOS nos Apples com monitores de partida automática.

- Digite IN #6 ou PR#6 para iniciar o DOS se você vir o prompt do BASIC.
- Digite CTRL-P6 se você vir o prompt do monitor.
- Simule uma partida fria no Apple IIe, uma vez que o DOS está na memória. Pressione OPEN-APPLE. Pressione e libere CTRL-RESET e, então, libere o OPEN-APPLE.

Partida quente

No Apple II PLUS e IIe, pressione CTRL-RESET. No Apple II, somente pressione o RESET.

Convenções e defaults

DOS aceita números decimais e hexa. Uma entrada hexa consiste em um sinal de dólar, dígitos hexadecimais de 0 a 9 e A até F (ex.: \$FE).

Sintaxe dos comandos do DOS

No comando, FILENAME,S6,D2,V9,

FILENAME segue a palavra de comando como primeiro argumento. Contém de 1 a 30 caracteres. Inicia com uma letra. Consiste em qualquer caractere, exceto a vírgula (ex.: A FILE, AN.EXCEEDINGLY.LONG.NAME)

- S6 Estabelece que a marca 2 contém o cartão de controle do disco; defaults para o slot usado por último (faixa = 0-7 no Apple II e Apple II PLUS; 1-7 no Apple IIe) (argumento opcional)
- D2 Estabelece que o drive #2 contém o disco a ser acessado; defaults para o drive acessado por último; sempre posicionar no drive #1 (faixa = 1 ou 2) (argumento opcional)
- V9 Estabelece que o número de volume do disco é 9 (atribuído pelo usuário); defaults para 254 com INIT; ignora número de volume em todos os outros comandos (argumento opcional)

Argumentos adicionais para os comandos dos DOS:

- A3000** Especifica um endereço 3000; defaults para os endereços dos quais foram BSAVED (faixa = 0-65535)
- B4** Especifica um **byte** ou **caractere** numérico (faixa = 0-32767; default=0; no acesso seqüencial B é um byte absoluto no arquivo; em acesso randômico, é um byte absoluto com o registro indicado pela opção R)
- L8** Especifica o **tamanho**, 8 bytes ou caracteres (faixa = 0-32767; default=1)
- R5** Especifica **5 campos** ou **linhas** para saltar da posição corrente no arquivo, nos comandos

POSITION e EXEC; (faixa = 0-32767;
default = 0 o início do arquivo)

R1 Especifica o **registro número 1**, nos comandos READ e WRITE de acessos randômicos; (faixa = 0-32767; default = 0 no OPEN, então para o último registro especificado)

Usando um comando DOS do programa

Encaixe o comando DOS no declarativo PRINT. Siga o PRINT com um CTRL-D e o string que contém o comando DOS.

```
10 PRINT "CATALOG" Invisível CTRL-D antes  
de C no AppB e IntB  
10 D$=CHR$(4) : PRINT D$; "CATALOG"  
Numa variável AppB
```

Usando discos e arquivos

CATALOG Exibe o número de volume, nomes de arquivo, o estado de travamento do arquivo (indicado por * no catálogo), tipos de arquivo, e tamanho nos setores; os tipos de arquivo são A (AppB), I (IntB), T (Texto), B (Binário) e R (Relocável)

INIT HELLO Formata um disco; apaga todos os dados previamente salvos; copia o

DOS e o programa na memória para o disco, dá nome ao programa *HELLO*; DOS corre *HELLO* quando o disco é posicionado (imm)

Procedimento de formatação

Inicie o DOS. Coloque um disco vazio no drive #1. Limpe a memória (NEW) e digite num programa de saudação

```
10 PRINT "WELCOME TO OSBORNE/MCGRAW-  
HILL" : END
```

Para formatar o disco e para dar nome ao programa *HELLO*, digite *INIT HELLO*. O programa *HELLO* inicia automaticamente quando você posiciona o disco.

DELETE *BADFILE*

Remove o arquivo *BADFILE*

LOCK *IMPT* Trava o arquivo *IMPT* protegendo-o de mudanças

RENAME *OLD, NEW*

Muda o nome do arquivo de *velho* para *novο*

UNLOCK *PUB* Remove a proteção do arquivo travado, *PUB*

VERIFY MY.DISK.FILE

Testa *MY.DISK.FILE* para leitura na memória

Ferramentas de programação

CHAIN LIBFILE Corre o programa BASIC

LIBFILE; não fecha arquivos abertos ou apaga variáveis da memória (IntB) (AppB encadeado: ver programas utilitários, CHAIN.)

FP, S6, D1 Lê Applesoft BASIC na memória do disco no drive 1 conectado ao slot 6; restaura o sistema (imm)

INT Muda para Integer BASIC, restaura o sistema (imm)

IN#3 Posiciona um disco conectado através do slot #3 e obtém a entrada subsequente do mesmo

PR#1 Posiciona um disco conectado através do slot #1 e envia a saída subsequente para aquele slot

LOAD ANIMALS

Copia o programa *ANIMALS* do disco para a memória (imm in IntB)

MAXFILES 6 Especifica um máximo de 6 buffers de arquivos para serem ativados de uma

vez (faixa = 1-16); quando o MAXFILES é executado, o DOS reserva 6 buffers de arquivos (6*595 bytes na memória); MAXFILES move HIMEM, apagando programas Integer e strings Applesoft

MON C,I,O Comandos de disco de ecos (C), entrada em disco (I), e saída em disco (O) (requer uma opção)

NOMON C,I,O

Cancela a exibição de comandos de disco (C), entrada em disco (I), e saída em disco (O) (requer uma opção)

RUN SLOT#

Copia o programa BASIC, *SLOT#*, do disco para a memória e o executa

SAVE DATAFILE

Grava o programa BASIC da memória para o disco e salva-o como *DATAFILE* (imm in IntB) (imm, pgm in AppB)

Comandos Binários de Disco

BRUN CALCROUTINE, A3000

Trasnfere um arquivo binário *CALCROUTINE* do disco para a memória, iniciando no endereço *A3000* e executa-o

BLOAD SHAPETABLE,A3000

Transfere um arquivo binário *SHAPETABLE* do disco para a memória, iniciando no endereço 3000; não o executa

BSAVE SHAPETABLE,A900,L8

Transfere dados binários da memória para o disco, iniciando no endereço 900 para 8 bytes; se o arquivo ainda não existe, o DOS o cria, arquivo tipo B; se ele existe, o DOS grava sobre ele

Programas utilitários

Estes programas são fornecidos no DOS 3.3 SYSTEM MASTER. A menos que indicado de outra forma, entre o *programa chamado* RUN.

CHAIN Carrega e corre um segundo programa Applesoft, mantendo as variáveis e arrays do primeiro programa (AppB)
ex.: 50 PRINT D\$; "BLOAD CHAIN, A520": CALL 520"SECOND.FILE")

CONVERT 13

Converte um disco de 13 para 16 setores; aceita cartões virgens nos nomes do

- arquivo ou usa BRUN MUFFIN para correr a versão binária
- COPY** Cópia discos inteiros (IntB)
- COPYA** Cópia discos inteiros; usado para tirar backups de discos (AppB)
- FILEM** Executa várias funções de suporte em disco como copiar um arquivo individual, aceitar cartões virgens nos nomes de arquivo, ex.: A=F, selecionar todos os arquivos que iniciam com A e terminam com F, ou usar BRUN FID para processar a versão binária
- MASTER** Converte um disco formatado num disco mestre (com DOS relocável nele); sem mudanças para os outros programas no disco; ou usa BRUN MASTER CREATE para correr a versão binária
- RENUMBER** Renumerar e intercala linhas de programas
- &RETURN** Renumerar o programa correntemente na memória: inicia na primeira linha (default = 10); termina com a última linha (default = 63999); atribue 10 para a primeira linha (default = 10); incrementa cada linha de 10 (default = 10)

&S500,E900,F100

Renumerar a antiga faixa de linha
500–900 para iniciar na linha número
100; põe o programa na memória para
o buffer de retenção

&HOLD Intercala dois programas, um que é
carregado na memória e o segundo
que reside no buffer de retenção

SLOT# Informa-lhe o slot corrente e os números
do drive

START13 Lê e grava 13 setores; ou usa BRUN
BOOT13 para correr a versão binária

Dispositivo de E/S

Saída para a impressora com o DOS presente

10 D\$=" " ;CTRL-D para os comandos do DOS

20 PRINT D\$;"PR#1

;Quando o cartão de controle da
impressora está no slot#1

;Permanecendo PRINT para os
declarativos da impressora

99 PRINT D\$;"PR#0"

;Dá saída para a tela novamente

Saída para a impressora sem o DOS presente

```
10 PRINT "PR#1"
```

;Cartão de controle da impressora
está no slot#1

;Permanecendo PRINT para os
declarativos da impressora

```
99 PRINT "PR#0"
```

Dá saída para a tela novamente

Saída para impressora programável

Este exemplo controla o Epson MX-80 de um
programa Applesoft com o DOS presente

```
10 D$=" " ;CTRL-D para os comandos do DOS
```

```
20 PRINT D$;"PR#1"
```

;O cartão de controle da impressora
está no slot#1

```
30 PRINT CHR$(27);"C"
```

;CHR\$(27) é ESC

;ESC-C,0 especifica o tamanho da
página em polegada

;CHR\$(6) especifica uma página de
6 polegadas de comprimento

```
40 PRINT CHR$(27);"N";CHR$(3)
```

;ESC-N pula a perfuração

;CHR\$(3) gera 3 linhas vazias depois
do texto

50 PRINT CHR\$(12)

;Posiciona o formulário para o início
da saída

99 PRINT D\$;"PR#0

;Dá saída para a tela novamente

Usando LIST para enviar uma listagem para a impressora

LOAD PGM Carrega o programa na memória

PR#1 Seleciona a impressora

LIST Envia uma listagem de programa para
a impressora

PR#0 Envia saída para a tela novamente

Criando e usando arquivos de texto

Um arquivo de texto (arquivo tipo T) é uma série de campos (1-32767 caracteres delimitados por um retorno de carro). Para usar o arquivo de texto, especifique o arquivo de trabalho (declarativo OPEN), o arquivo de entrada (declarativo READ), e o arquivo de saída (declarativo WRITE). Use INPUT e GET para captar os dados; use PRINT para gravar dados. Sempre feche (CLOSE) o arquivo após executado.

Acesso seqüencial

Não especifique o tamanho do registro quando você abre o arquivo. DOS guarda cada campo seqüencialmente. Cada vez que você abre o arquivo, o DOS inicia lendo ou gravando no início do arquivo.

APPEND *SAFILE*

Abre o arquivo de acesso seqüencial *SAFILE* para gravação, iniciando no fim do arquivo; o arquivo já deve existir (pgm)

CLOSE *DOCUMENT*

Fecha o arquivo de texto *DOCUMENT* e libera os buffers dos arquivos associados (default = todos os arquivos de texto)

OPEN *SAFILE* L8

Abre ou cria o arquivo de texto *SAFILE* para acesso seqüencial (pgm)

POSITION *SAFILE*, R5

Estabelece a posição para acessar o arquivo de acesso seqüencial *SAFILE* no *sexto* campo pulando 5 campos (pgm)

READ *SAFILE*, B4

Lê arquivo seqüencial *SAFILE* iniciando no byte 4 dentro do arquivo (pgm)

WRITE *SAFILE*, B4

Grava o arquivo de acesso seqüencial *SAFILE* iniciando no byte 4 (pgm)

Acesso Randômico:

Especifica o tamanho do registro quando você abre o arquivo. Um registro é um campo ou uma coleção de campos. O DOS pode iniciar lendo ou gravando um registro específico, locado em qualquer lugar no arquivo.

OPEN *RAFILE*, L8

Abre ou cria *RAFILE* com um tamanho de registro de 8 bytes para acesso randômico (pgm)

READ *RAFILE*, R1, B4

Lê do arquivo de acesso randômico *RAFILE*, começando no byte 4 dentro do registro 1 (default = ao primeiro byte do primeiro registro) (pgm)

WRITE *RAFILE*, R1, B4

Grava o arquivo de acesso randômico *RAFILE*, iniciando no byte 4 no registro 1 (default = ao primeiro byte do primeiro registro) (pgm)

Arquivo EXEC

Arquivos EXEC correm outros programas, introduzem informações repetidamente, convertem um programa BASIC em um arquivo de texto, inserem uma sub-rotina num programa etc. Seu programa BASIC que cria um arquivo EXEC pode conter comandos do DOS, declarativos BASIC, e qualquer coisa que você possa digitar do teclado. Um comando DOS num arquivo EXEC não é precedido por CTRL-D. Por exemplo, para criar um arquivo EXEC denominado MYEXEC, digite

```
10 PRINT D$=CHR$(4)
20 PRINT D$;"OPEN MYEXEC"
30 PRINT D$;"WRITE MYEXEC"
40 PRINT "CATALOG"
99 PRINT D$;"CLOSE MYEXEC"
```

Correndo este programa cria-se o arquivo MYEXEC e se coloca o mesmo no disco. Para correr MYEXEC, use o comando EXEC.

EXEC MYEXEC, R5

Corre o arquivo EXEC, *MYEXEC*, a partir da linha 5; (default = início do arquivo EXEC); se o arquivo EXEC corre outro programa, quaisquer declarativos INPUT neste segundo

programa são lidos do arquivo EXEC
ao invés do teclado

O PROGRAMA MONITOR

O programa monitor reside numa memória read-only (ROM), separada das áreas nas quais o BASIC e o DOS são guardados, e age como supervisor controlando todos os outros programas. Ele executa tarefas, como iniciar o BASIC residente, controlar a comunicação entre o teclado e a tela, entre a memória e os dispositivos periféricos, e ler o DOS para a memória. Acessa o monitor para criar tabelas de formato, examinar a memória para isolar programas de hardware ou programar na linguagem Assembly. O prompt do monitor é *.

Notações e convenções

<i>loc</i>	Significa um endereço hexa de memória
\$	Indica um dado hexa; nos exemplos os dígitos hexa são mostrados como letras minúsculas e as cartas de comando como letras maiúsculas

Comandos do monitor

CALL-151 Introduz o monitor

CTRL-C Retorna ao BASIC do monitor

CTRL-B Retorna ao Applesoft com as variáveis ainda intactas

3CTRL-K Lê entradas a partir do dispositivo conectado através do slot#3 (faixa Apple II e Plus = 1-7; faixa IIe = 0-7); use 0 CTRL-K para dar entrada do teclado novamente

3DOG Retorna do monitor para Applesoft baseado em disco com variáveis e programa intactos

6CTRL-P Envia saída ao dispositivo conectado através do slot#6 (faixa Apple II e Plus = 1-7; faixa IIe = 0-7); use 0CTRL-P para enviar saída para a tela novamente

CTRL-Y Salta para o endereço 3f8, onde há espaço para uma instrução de salto de linguagem de máquina definida pelo usuário

Usando registradores

CTRL-E Exibe o acumulador, os registradores índice X e Y, o registrador de status

do processador e o indicador de pilha (stack pointer)

CTRL-E: *9f ef 54 ab* RETURN

Altera os registradores iniciando com o acumulador; coloca *9f* no acumulador, *ef* no registrador X, *54* no registrador Y e *ab* no registrador de status do processador

5a6b Exibe o conteúdo da posição *5a6b* e fixa o indicador de endereço para *5a6b*

RETURN Exibe a próxima posição e até 7 bytes mais

:b6 Muda o valor da posição corrente para *b6* e incrementa o indicador de endereço para apontar a próxima posição

fe96:a5 Muda o valor da posição *fe96* para *a6*

:aa 85 33 20 67 fd

Muda o valor da posição corrente para *aa*, move para a próxima posição e repõe o valor com *85* etc.; máximo de 83 locações sucessivas

7f+8a Adiciona dois números hexa; se o resultado for maior que FF, trunca o dígito mais significativo, exhibe os primeiros 8 bits

Oa-2d Subtrai *2d* de *od*; exhibe o resultado; se o resultado for menor que 0, exhibe o complemento de um do resultado

3d0 G Salta para o endereço *3d0* e executa a instrução de linguagem de máquina; quando o DOS está na memória, este exemplo salta para o DOS

1200<2000.2100M

Movimenta dados da faixa de endereços *2000-2100* para a memória iniciando em *1200*; para zerar um bloco da memória

1d00 :00 Introduz zeros na posição *1d00*

1d01<1d00.1dfeM

Repete zeros da posição *1d01* até *1dfe*

2000.20ff R RETURN

Lê os dados da fita para a memória, da posição *2000* até *20ff*; primeiro posicione a fita ao sinal sonoro, então pressione PLAY; para operação em disco, veja BLOAD e BRUN

32d0<0.cV

Compara o conteúdo da posição *32d0* com 0; continuando por simples incrementos de byte até que *32de* seja alcançado; lista discrepâncias

- 2200.2fffW** Grava o conteúdo das locações 2200 até 2fff e um byte de checksum para fita; primeiro pressione os botões RECORD e PLAY, então pressione RETURN (para disco, veja BSAVE)
- I** Fixa o modo inverso de vídeo
- N** Fixa o modo normal de vídeo

POSIÇÕES ÚTEIS PEEK E POKE

Janela de texto e controle do cursor

<i>Loc</i>	<i>Controles</i>	<i>Faixa permitida de valores</i>
32	Margem esquerda da janela de texto	0 a 39
33	Largura da janela de texto	1 a 40
34	Margem superior da janela de texto	0-23
35	Margem inferior da janela de texto	0-23
36	Posição horizontal do cursor	0-39
37	Posição vertical do cursor	0-23

Condições de erro

216	Flag de erro
218,219	Linha causadora do erro; para ler, use PEEK(219)*256+PEEK(218)
222	Tipo do erro (1-15 são DOS; o resto é Applesoft):

0	NEXT sem FOR
1	Linguagem não-disponível
2 ou 3	Erro de faixa
4	Gravação protegida
5	Fim dos dados
6	Arquivo não encontrado
7	Volume não combina
8	Erro de E/S
9	Disco cheio
10	Arquivo travado
11	Sintaxe
12	Buffers não-disponíveis
13	Tipo de arquivo não combina
14	Programa muito grande
15	Sem comando direto
16	Sintaxe
22	RETURN sem GOSUB
42	Fora dos dados
53	Quantidade ilegal
69	Overflow
77	Fora da memória
90	Declarativo não definido
107	Má subscrição
120	Array redimensionado
133	Divisão por zero
163	Tipo não combina
176	String muito longo

191	Fórmula muito complexa
224	Função não definida
254	Resposta ruim a um INPUT
255	CTRL-C estava pressionado

Endereços (decimal) maiores que 32767 são expressos como números negativos. Para obter o positivo equivalente, some 65536 ao número negativo.

Posições do teclado

-16384	Último caractere lido (teclado) (conteúdo - 128) produto ASCII
-16368	Flag do teclado

Posições de controle de jogo

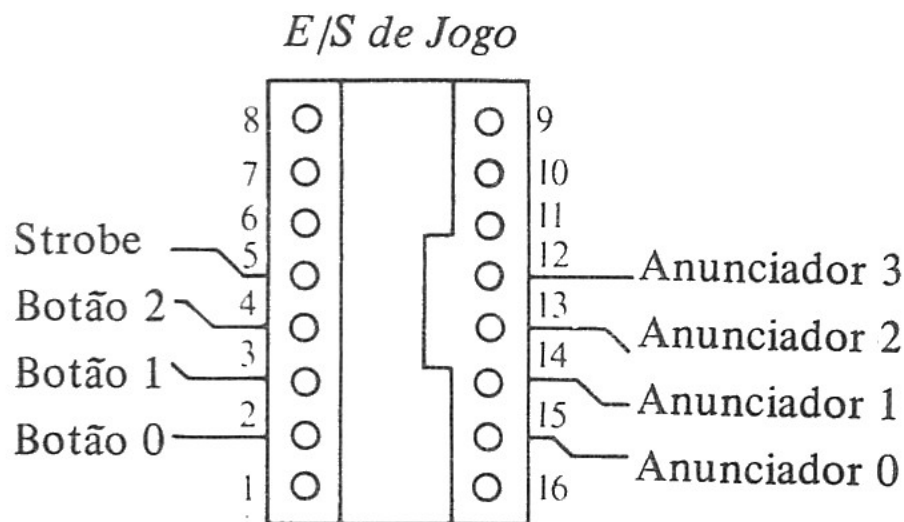
<i>Desligado</i>	<i>Ligado</i>	<i>Função</i>
-16296	-16295	anunciador 0
-16294	-16293	anunciador 1
-16292	-16291	anunciador 2
-16290	-16289	anunciador 3
-16287		lê o botão 0
-16286		lê o botão 1
-16285		lê o botão 2
-16272		dispara o strobe

Localizações de saída de click

- 16352 produz um click na tomada de saída do cassete
- 16336 produz um click no auto-falante interno

Chaves de exibição

- 16304 fixa o modo gráfico
- 16303 fixa o modo de texto
- 16302 fixa gráficos de tela completa
- 16301 fixa janela de texto de 4 linhas, gráficos mais texto
- 16300 fixa gráficos ou página 1 do texto
- 16299 fixa gráficos ou página 2 do texto
- 16298 fixa gráficos de baixa resolução
- 16297 fixa gráficos de alta resolução



CÓDIGOS DE CARACTERES ASCII

Os códigos 96-127 produzem os mesmos caracteres que os códigos 32-63. Os códigos 96-12 estão em letras minúsculas em alguns dispositivos. Os códigos 128-255 repetem os códigos 0-127 (n.d. = não disponível no teclado do Apple II)

Código	Caract	Chaves	Código	Caract	Chaves	Código	Caract	Chaves	Código	Caract	Chaves
0		CTRL-@	24		CTRL-X	48	0	0	72	H	H
1		CTRL-A	25		CTRL-Y	49	1	1	73	I	I
2		CTRL-B	26		CTRL-Z	50	2	2	74	J	J
3		CTRL-C	27		ESC	51	3	3	75	K	K
4		CTRL-D	28		n.d.	52	4	4	76	L	L
5		CTRL-E	29		CTRL-SHIFT-M	53	5	5	77	M	M
6		CTRL-F	30		CTRL-^	54	6	6	78	N	N
7	(bell)	CTRL-G	31		n.d.	55	7	7	79	O	O
8	←	CTRL-H	32	ESPAÇO	ESPAÇO	56	8	8	80	P	P
9		CTRL-I	33	!	!	57	9	9	81	Q	Q
10	LINEFD	CTRL-J	34	"	"	58	:	:	82	R	R
11		CTRL-K	35	#	#	59	;	;	83	S	S
12		CTRL-L	36	\$	\$	60	<	<	84	T	T
13	RETURN	CTRL-M	37	%	%	61	=	=	85	U	U
14		CTRL-N	38	&	&	62	>	>	86	V	V
15		CTRL-O	39	,	,	63	?	?	87	W	W
16		CTRL-P	40	((64	@	@	88	X	X
17		CTRL-Q	41))	65	A	A	89	Y	Y
18		CTRL-R	42	*	*	66	B	B	90	Z	Z
19		CTRL-S	43	+	+	67	C	C	91	[n.d.
20		CTRL-T	44	,	,	68	D	D	92	\	n.d.
21	→	CTRL-U	45	-	-	69	E	E	93]	SHIFT-M
22		CTRL-V	46	.	.	70	F	F	94	^	^
23		CTRL-W	47	/	/	71	G	G	95		n.d.

ÍNDICE ANALÍTICO

A

A, 35, 41
ABS, 16
AND, 19
APPEND, 45
ASC, 16
AT, 24, 31, 33
ATN, 16
AUTO, 7

B

B, 35
BOOT, 42
BRUN, 39
BLOAD, 30, 40
BSAVE, 30, 40

C

CALL, 12, 22
CALL-151, 49
CATALOG, 36
CHAIN, 38, 40
CHR\$, 16, 26, 43
CLEAR, 7
CLOSE, 45
CLR, 8
COLOR=, 25
CON, 15

CONT, 15
CONVERT, 40
COPY, 41
COPYA, 41
COS, 16
CTRL-B, 49, 56
CTRL-C, 49, 56
CTRL-D, 36, 56
CTRL-E, 49, 50, 56
CTRL-K, 49, 56
CTRL-P, 34, 49, 56
CTRL-X, 3, 56
CTRL-Y, 49, 50

D

D, 35
DATA, 10
DEF, 18
DEL, 8
DELETE, 37
DIM, 9
DRAW, 31
DSP, 15

E

END, 9
EXEC, 47
EXP, 16

F

FID, 41
FILEM, 41
FLASH, 22
FN, 18
FOR, 12
FP, 8, 38
FRE, 16

G

G, 49
GET, 11
GOSUB, 12
GOTO, 13, 14, 15
GR, 22

H

HCOLOR=, 25
HGR, 23
HGR2, 23
HIMEM:, 23
HLIN, 24
HOME, 20
HPLOT, 24
HTAB, 20

I

IF, 13
IN#, 8, 38
INIT, 36, 37
INPUT, 10

INT, 8, 17, 38
INVERSE, 21

L

L, 35
LEFT\$, 17
LEN, 17
LET, 10
LIST, 11
LOAD, 30, 38
LOCK, 37
LOG, 17
LOMEM:, 23

M

M, 51
MAN, 8
MASTER, 41
MAXFILES, 38
MID\$, 17
MOD, 19
MON, 39

N

NEW, 8, 37
NEXT, 12
NODSP, 15
NOMON, 39
NORMAL, 22
NOT, 19
NOTRACE, 15

O

ON, 14
ONERR, 14
OPEN, 36, 45, 46
OR, 19

P

PDL, 17
PEEK, 14, 17, 21, 52
PLOT, 24
POKE, 10, 22, 52
POP, 14
POS, 21
POSITION, 45
PR#, 8, 34, 38
PRINT, 11, 42

R

R, 35, 36, 46, 51
READ, 10, 36, 46
RECALL, 30
REM, 9
RENAME, 37
RENUMBER, 41
RESTORE, 10
RESUME, 14
RETURN, 41, 50
RIGHT\$, 17
RND, 18
ROT=, 32
RUN, 8, 39

S

S, 35
SAVE, 30, 39
SCALE=, 32, 33
SCRN(, 26
SGN, 18
SHLOAD, 31
SIN, 18
SLOT, 42
SPC(, 21
SPEED(, 22
SQR, 18
START, 42
STEP, 12
STOP, 15
STORE, 31
STR\$, 18

T

TAB(, 21
TAB, 21
TAN, 18
TEXT, 21
THEN, 13
TO, 12, 24
TRACE, 15

U

UNLOCK, 37
USR, 18

V

V, 35, 51

VAL, 18

VERIFY, 38

VLIN, 25

VTAB, 21

W

W, 31, 52

WAIT, 15

WRITE, 36, 46, 47

X

XDRAW, 33

Impresso nas oficinas da
EDITORA PARMA LTDA.
Fone: 209-5077
Av. Antônio Bardella, 280
Guarulhos - São Paulo - Brasil
Com filmes fornecidos pelo Editor

OUTROS LIVROS NA ÁREA

GUIAS DO OPERADOR

Gifford — Apple II — Comandos Básicos

Ingraham — CP/M — Comandos Básicos

Wilson — Visicalc — Comandos Básicos

Wilson — IBM-PC — Comandos Básicos

GUIAS DO USUÁRIO

Baras — Lotus 1-2-3 — Guia do Usuário

Castlewitz — Visicalc — Guia do Usuário

Curtis — WordStar — IBM-PC — Guia do Usuário

Ettlin — WordStar — Guia do Usuário

Hoffman — MS/DOS — Guia do Usuário

Hogan — CP/M — Guia do Usuário

Poole — Apple II — Guia do Usuário

Sachs — IBM-PC — Guia do Usuário

Sanders — Manual do Apple — Macintosh

Townsend — dBASE II — Guia do Usuário

* Com referência a outros títulos, consulte nosso Catálogo de Informática.