

Jornal do hpclub do Brasil

<http://www.hpclub.com.br>

Edição nº 19 - 13/01/2001

Primeiro jornal do milênio

Entramos agora em um novo milênio, uma coisa que poucas pessoas na história da humanidade tiveram a oportunidade de passar. Agora podemos dizer das "coisas antigas" que ocorreram no "milênio passado", como por exemplo o invento das calculadoras HP, ou podemos dizer ainda: "desde o milênio passado que eu mexo com essas máquinas!".

Diferente do que muitas pessoas disseram, na virada do milênio (leia-se falsa virada do milênio de 1999 para 2000) não houve um bug mundial nos computadores fazendo tudo entrar em colapso e na verdadeira virada do milênio (2000 para 2001) houve um pequeno bug na república Tcheca que fez com que os trens parassem de funcionar durante algumas horas, além disso nada de mais grave ocorreu, o mundo não acabou como foi dito por algumas seitas espalhadas pelo mundo nem nenhum messias apareceu na Terra para salvar a todos.

O hpclub do Brasil, como muitas outras coisas, também sobreviveu a entrada de 2001 e como era de se esperar ele continua como sempre foi, tentando sempre trazer novidades sobre o mundo das calculadoras HP, assim como servir como "Ponto de encontro dos usuários das calculadoras HP". O jornal do hpclub do Brasil também sobreviveu e está aqui com essa primeira edição do novo milênio. Seu formato continua o mesmo (só porque é um novo milênio você esperava que tudo mudasse? Você mudou completamente sua vida esses últimos dias???) e sempre esperando trazer novidades e dicas úteis a todos os níveis de usuários, desde os novatos até os programadores avançados.

Nessa primeira edição do milênio, do século e do ano, como todas as outras vocês encontrarão dicas de manuseio, programação básica e avançada além de dicas rápidas e alguns textos referentes à assuntos relacionados à HP.

Uma informação triste é dada nesse jornal como pode ser visto logo abaixo, o anuncio do falecimento de William Hewlett, um dos fundadores da Hewlett-Packard. As calculadoras HP e o hpclub do Brasil não existiriam se não fosse esse grande homem de negócios.

Tacio Philip Sansonovski - hpclub do Brasil

Falecimento de William Hewlett

"Bill Hewlett, HP Co-founder, Dies at 87

PALO ALTO, Calif., Jan. 12, 2001

William R. Hewlett, co-founder and former president of Hewlett-Packard Company, and one of the United States' foremost business leaders, technologists and philanthropists, died at home in his sleep at 8 a.m. PST today of natural causes. He was 87 years old."

Tradução

Bill Hewlett, co-fundador da HP morre aos 87 anos.

Palo Alto-California, 12 de janeiro de 2001

William R. Hewlett co-fundador e antigo presidente da Hewlett-Packard, homem de negócios, técnico e filantropo com posição de destaque entre os mais importantes dos EUA faleceu em sua casa de morte natural, enquanto dormia, as 8 horas da manhã de hoje (hora do Pacífico) aos 87 anos.

Informação enviada por Carlos Marangon

Ajustar funções não lineares

Para ajustar uma função que não seja linear, após entrar os dados em Σ DAT entre em [->] STAT ... *Linear Fit*. No campo *Model* você encontrará as seguintes opções:

LINFIT - ajuste linear
LOGFIT - ajuste logarítmico
EXPFIT - ajuste exponencial
PWRFIT - ajuste potência (seu caso [para a função X^2])
BESTFIT - com essa função a HP verifica qual a função que se adapta melhor aos seus dados.

Após selecionar o ajuste desejado é só clicar em |OK| ou [ENTER], a HP retornará na pilha operacional a função ajustada além do coeficiente de correlação e covariança.

Tacio

Simple programa usando INPUT ou INFORM para calculo de pH

Utilizando INPUT

```
<< "Qual a concentração de H?" "" INPUT 'H' STO 'pH=-log(H)'\>>
```

ou utilizando variáveis locais:

```
<< "Qual a concentração de H?" "" INPUT -> H 'pH=-log(H)' >>
```

Utilizando INFORM

```
<< "CÁLCULO DE pH" {{ "H" "Qual a concentração de H?" 0 }} 1 {{ }} DROP INFORM  
DROP OBJ-> DROP -> H 'pH=-log(H)' >>
```

O primeiro DROP é para deitar fora o 1 que o comando INFORM deixa na pilha e o segundo é porque depois de executar o OBJ-> fica na pilha o objeto e um número de objetos que também não interessa.

José Alberto Novais Machado

Para fazer gráficos de funções não inclusas na HP

Apesar da HP não possuir todas as funções matemáticas existentes você pode fazer gráficos de outras funções entrando no campo EQ da tela plot um programa que resulte na função desejada.

Um exemplo seria plotar um gráfico de logaritmo em uma base diferente de 10 (LOG) ou diferente de e (LN). A HP não possui essa função, mas é só interpretar a propriedade de logaritmo de mudança de base:

$$\log_b a = \frac{\log_c a}{\log_c b}$$

Para fazer um gráfico de log na base .5 por exemplo você pode armazenar em EQ:

```
<< X LOG .5 LOG / >>
```

Note que no programa é entrado o valor de X (variável definida) e retornado um valor que corresponde a Y (variável indefinida) da função.

Desse mesmo modo, fazendo-se alterações em programas você poderá plotar diferentes funções em sua HP, aumentando assim suas possibilidades gráficas.

Tacio

WSLOG - o que significam os códigos

Ao digitar na HP WSLOG é retornado na pilha operacional 4 strings com código, data e hora dos 4 últimos warmstarts (resets) da HP. Os códigos do WSLOG significam o seguinte:

- 0- O 'relatório' foi limpo pressionando as teclas ON+SPC (não funciona na hp49g)
- 1- O sistema de interrupções detectou uma condição de bateria fraca nos contatos da bateria (não é o mesmo que ter a tensão do sistema baixa), e colocou a HP no modo coma (Deep Sleep) (com o relógio sem funcionar). Quando ON é pressionado depois da tensão da bateria ter sido restaurada, o sistema faz um arranque a 'morno' (equivalente a um ON+C) e coloca um 1 no relatório do WSLOG.
- 2- O hardware falhou a transmissão (time-out)
- 3- O programa correu através do endereço 0
- 4- O relógio do sistema está corrompido (no log aparece por exemplo 4-0000000000000)
- 5- A calculadora acorda depois de ter estado no modo coma (Deep Sleep), por exemplo pressionando ON ou um alarme
- 6- Não usado
- 7- Um nibble de 5 bits (palavra de teste CMOS) na RAM está corrompida. (esta palavra é verificada em cada interrupção, mas é somente usada como indicador de RAM corrompida.
- 8- Não usado
- 9- A lista de alarmes está corrompida.
- A- Um programa em system RPL saltou para #0
- B- O módulo do cartão foi removido
- C- Ocorreu um reset do hardware (por exemplo, uma descarga eletrostática ou um reset feito pelo usuário)
- D- Uma rotina de controle de erros esperada em system RPL não foi encontrada.

José Alberto Novais Machado

Converter códigos sysrpl da HP48 para HP49

Um procedimento que em alguns casos funciona para converter um código sysrpl da HP48 para HP49 é o descrito abaixo. Esse procedimento é muito útil na conversão de códigos simples como cálculos simples, em outros casos onde existe uma apresentação com telas de INFORM, INPUT ou outras, além de haver diferentes arquivos do programa esse método muitas vezes não funciona ou torna-se muito trabalhoso.

1. Instale em sua HP48 o Jazz e algum programa como Hackit que serve para converter arquivos em string hexadecimal.
2. Pegue o programa da HP48 que deseja converter para HP49 e usando o Jazz com a HPTabs da HP48 decompile o objeto (ASS)
3. Mude os códigos de endereço no editor (os comandos não é necessário mudar já que logo em seguida será compilado para HP49). Os códigos que aparecerem PTR xxx ou ROMPTR xxx devem ser editados um a um para seu correspondente para HP49.
4. Instale agora na HP48 a hptabs do Jazz com os endereços para HP49
5. NÃO TESTE O PROGRAMA NA HP48!
6. Com o Hackit ou semelhante converta o código ou DIR do programa para string hexadecimal (->H ou ->ASC normalmente)
7. Transfira a string hexadecimal para HP49 em modo ASCII
8. Na HP 49 converta novamente a string em programa (H->) - lib 256
9. Teste o programa.

Como foi dito acima esse procedimento algumas vezes funciona.

Tacio

Como detectar um objeto com o infravermelho da HP48

Eu posso lhe informar como enviar dados (ligar o IR e detectar) pela porta do IR e detecta-los, a parte do programa vocês podem fazer sozinhos (pode ser em RPL mesmo) ...

IRECHO:

```
SAVE
D0= 0011A  <- Inicializa os dados do IR.
D1= 0011C
A=0 A
DAT0=A 1
LC 8
DAT1=C      <- Apenas seta os bits p/ inicializar o IR
C=DAT0 1    <- É aqui que ele lê o valor do IR, 1 p/ recebeu IR, 0 p/ nada.
?C#0 P
GOYES W1
LC 2A2B4    <- Posição do Numero 0 no sistema (0 p/ stack)
GONC W2
*W1
LC 2A2C9    <- Posição do numero 1 no sistema (1 p/ stack)
*W2
RSTK=C
LOAD
C=RSTK
A=C A
PC=(A)
P+1
@
```

Compile usando o Masd, ele retorna 1 ou 0 dependendo do estado do IR (1 = Objeto perto, 0 sem objeto). Abaixo é dado um programa exemplo para usar o programa acima.

```
<<
DO
  IF
    IRECHO
  THEN
    1000 .01 BEEP
  END
UNTIL
  KEY
END
DROP
>>
```

O programa gerará beeps caso algo esteja presente e que seja capaz de refletir a luz do IR. Para que o programa funcione normalmente sem erros de detecção é necessário que seja retirada a capa protetora do IR. Caso você queira fazer um programa 100% ML você deve utilizar contadores para controlar o tempo, desabilitar o interrupt handler p/ maior velocidade, e checar teclas p/ sair do programa.

Daniel

hpbrasil

O local certo para você comprar sua calculadora HP!

Todos os modelos de calculadoras inclusive a **HP49G!**
O melhor preço do mercado com entregas em todo o Brasil

<http://orbita.starmedia.com/~hpbrasil>

Como funcionam os testes de teclado (KBD)

Um dos auto testes das calculadoras HP é o teste de teclado (KBD - KeyBoard), para acessa-lo deve-se pressionar as teclas [ON] e [D] simultaneamente. Irá aparecer uma tela com um risco ao centro e dois nos cantos na HP48 ou uma tela com textos mostrando os testes disponíveis na HP49.

Para iniciar o teste na HP48, após a tela inicial deve-se clicar na tecla [E]. Aparecerá uma tela mostrando no topo KBD1. Para efetuar o teste deve-se iniciar clicando na tela [A], seguindo para direita até a tecla [F], descendo-se em seguida para tecla [MTH] e seguindo-se sempre a direita e abaixo até chegar a tecla [+]. Ao final de todas as teclas deverá aparecer "KBD1 OK" caso não haja nenhum problema com o teclado da calculadora. Existe ainda o teste KBD2, que é utilizado em testes internos na própria Hewlett Packard, Este teste é acessado após a tela inicial com a tecla [F], se alguém souber a seqüência de teclas desse teste por favor nos envie para que a coloquemos na próxima edição do jornal.

Na HP 49, após a tela inicial deve-se clicar em [3] (KBD1) e pressionar as teclas da esquerda para a direita : [F1] até [F6]. [APPS], [MODE] , [TOOL], seta para cima, [VAR], [STO], [NXT], seta para esquerda, seta para baixo, seta para direita [HIST] até backspace [Y^x] até [TAN] e assim sucessivamente até a tecla [ENTER].

Para efetuar o teste KBD2 na HP49, pressionar as teclas: [F1], [F2], [F3], [F4], [F5], [F6], [APPS], [MODE], [TOOL], [VAR], [STO], [NXT], SETA ESQUERDA e [ALPHA].

Tacio Philip
Romualdo Lourençon
José Alberto Novais Machado

Reis longos - sysrpl

Em sysrpl é possível efetuar cálculos com números com uma precisão muito maior do que em USER-RPL, os chamados reais longos - ou estendidos - (Long Real), podendo-se trabalhar com valores no intervalo de 1E-50000 até 9.999999999999999E49999, ou seja, pode-se efetuar cálculos com 15 algarismos significativos e além disso usar potências de 10 até 50000!

A estrutura do número real longo é semelhante a um real simples quando digitado um código sysrpl, com a única diferença que para um real o valor é definido com um caracter porcentagem (%) e para o real longo com dois caracteres (%%) como pode ser visto nos exemplos abaixo:

%% 1 | %% 1E-50000 | %% 9.999999999999999E49999

Para se efetuar operações não se pode usar as funções sysrpl para reais, mas sim comandos para reais estendidos como alguns mostrados abaixo (note que para praticamente todos comandos sysrpl para reais existe uma semelhante para reais longos)

%% - +soma de dois reais longos	%%>= - maior ou igual
%% - subtração	%%SIN - seno
%%* - multiplicação	%%COS - cosseno
%%/ - divisão	%%TAN - tangente
%%< - comparação menor	%%SQRT - raiz quadrada
%%> - comparação maior	%%>%% - converte real longo em real
%%<= - menor ou igual	%%>%% - converte real em real longo

Tacio

Você sabia?

- Um dos modos de se usar o cabo de conexão da HP sem o mouse é configurar o Windows nas opções de acessibilidade para deficientes, assim pode-se usar o ponteiro do mouse através do teclado numérico . É só entrar em ... *Iniciar ... configurações ... painel de controle* e depois *opções de acessibilidade*.
Claudio
- Utilizando-se o comando PDIM você pode criar imagens ou plotar gráficos maiores que 131 x 64 pixels, para isso especifique na pilha operacional os valores x e y no formato de binários inteiros (#nh) e execute o comando PDIM.
Tacio
- comando UBASE converte qualquer valor com unidade para seu valor com unidades no sistema internacional de medidas, com essa função você pode saber por exemplo que 1_atm é correspondente a 101325_kg/(m*s^2) no sistema internacional.
Tacio

Iniciando programação ML - parte IV

Gráficos Continuação

Trabalhando com pixels.

O display da HP funciona por meio de nibbles. Cada nibble trabalha com 4 bits e estes controlam os 4 pontos do display.

Observe: O numero F (hexa) é representado por 1111 em binário. Quando você usa um programa e bota para ele escrever o valor F no display você verá 4 pixels no canto superior do display. É essa a forma que o display trabalha: cada pixel representa um bit dentro de um nibble. Caso você queira testar essa estrutura o programa abaixo mostra bem claramente o que está acontecendo com cada bit quando se seta determinados valores:

```
::                               Observe que os pixels são tratados na forma Binária:
CLEARVDISP                       o número 3 tanto em Hexa como em Decimal é representado
CODE                              por 0011, como o processador utiliza o chamado "little
SAVE                              indian" o ultimo bit se torna o mais significante
GOSBVL ="D0->Row1"              aparecendo dois quadradinhos da esquerda para a direita.
LA 00                             Você deve estar se perguntando: Bom eu já sei disso
LC 0F                             tudo, quero saber é como acender um pixel! Existe uma
*lp1                              forma de acender um pixel que é a mais simples, para isso
DAT0=A B                          basta você saber algumas coisas:
A+1 B                             A posição do bit a ser aceso, a posição do nibble em
D0+ 34                             que o bit do pixel está e a linha, tendo isso você pode
D0+ 34                             acender qualquer pixel. Mas como obter isso tendo apenas
C-1 B                             as coordenadas X e Y? Simples:
GONC lp1                           Para calcular a posição do Bit tem-se a formula:
LOADRPL                            2^(CoordenadaX MOD 4) -> MOD = Resto da divisão de X por 4
SetDAsTemp                          Para a posição do Nibble:
;                                    Floor(CoordenadaX/4) + CoordY*ValordaLinha
@                                    ValordaLinha: 2*(Floor((LarguraDoGrob-1)/8)+1)
```

Cabos Pag's - Cabos de comunicação para HP48

Sua interface com o PC

Conecte sua HP48 ao PC e aproveite todo o seu potencial!

Maiores informações **Cabos Pag's**

<http://www.abati.com.br/CabosPags/>

```

Transformando isso tudo em um programa tem-se isso:
::
CLEARVDISP
CODE
SAVE
GOSBVL ="D0->Row1"
% Esse código pode ser usado como Subrotina para qualquer programa...
% A[A] = Coordenada X
% C[A] = Coordenada Y
% Ele retorna um pixel X,Y aceso no HARDBUFF...
% Caso você esteja trabalhando com um grob independente e que você tenha
% o seu tamanho diferente do display basta modificar os dados abaixo...
% Ele utiliza A[W] B[W] C[W] R0[A] SB

LC 0000F
LA 0000F

B=0 W
B=C A
LC #00056 <- Apenas um teste para que não haja problemas.
?B<C A <- sai caso a posição do pixel seja fora do display Y> Altura.
GOYES YOK <- Y está beleza ... teste o X...
*quit
LOADRPL
*YOK
C=0 W
LC #00131 <- carrega C com 131 decimal (tamanho do display, modifique p/
tamanho de grob diferente)
?A>=C A <- sai caso X >= que a Largura.
GOYES quit

R0=A.F A <- A[A]=R0= Coordenada X
ASRB A <- X DIV 2
ASRB A <- X DIV 4
AR0EX A <- R0=X DIV 4, A[A]=X
ASRC <- Trabalha com X DIV 4 p/ obter X MOD 4
A+A S
A+A S
A=0 A
ASLC
ASRB B
ASRB B <- A[A] = X MOD 4 = Valor da posição do Bit

% Trabalhará agora com o Bit dentro do Nibble a ser aceso...
% Bit = 0, Valor = 1
% Bit = 1, Valor = 2
% Bit = 2, Valor = 4
% Bit = 3, Valor = 8

A=0 S <- o A[S] irá ficar com o valor da posição do Bit...
A+1 S
*maskl

```

```

A-1 A    <- Se o bit = 0
GOC maskd <- Valor encontrado...
A+A S    <- caso não ... multiplique o Valor por 2
GOTO maskl <- ... e continue no looping...

% A[S] = Valor do pixel a ser aceso.
% R0[A] = X pixel.
% B[W] = Y pixel.
% C[A] = Comprimento do Grob
% D1= Endereço do conteúdo do Grob (no caso o display).

*maskd    <- calcula a posição do nibble a partir da coordenada Y
C-1 A
CSRB A
CSRB A
CSRB A
C+1 A
C=C+C A   <- C[A]= Linha a ser modificada.
ABEX W    <- Salva o valor do pixel a ser aceso em B[W] e põe em A[W] a
Coordenada Y.

B=C A     <- esse código é uma modificação do MPY encontrada na ROM (tem a
função de multiplicar...)
C=0 A     <- Multiplica C[A] por A[A]
*mull
SB=0
ASRB A
?SB=0
GOYES muld
C=C+B A
*muld
B+B A
?A#0 A
GOYES mull <- resultado em C[A]

A=R0.F A <- A[A]=Posição X
A+C A
CDOEX
C+A A
CDOEX
C=DAT0 S
C=C!B S
DAT0=C S

LOADRPL
ENDCODE
SetDAsTemp
;
@

```

Esse código pode ser modificado para trabalhar independentemente do HARDBUFF, você pode trabalhar com grobs no stack que quiser, ou mesmo trabalhar como subrotina para um programa maior.

Luis Daniel