Seção Especial: Construindo Seu Próprio Computador

Nos vários problemas que se seguem, nos desviamos temporariamente do mundo da programação em linguagem de alto nível. Vamos "remover a cobertura" de um computador e verificar sua estrutura interna. Apresentamos a programação em linguagem de máquina e escrevemos vários programas em linguagem de máquina. Para fazer com que isso seja uma experiência válida, construímos um computador (por intermédio da técnica de *simulação* baseada em software) no qual você pode executar seus programas em linguagem de máquina.

7.18 *(Programação em Linguagem de Máquina)* Vamos criar um computador a que chamaremos Simpletron. Como o nome já diz, ele é um equipamento simples, mas também, como veremos em breve, poderoso. O Simpletron executa programas escritos apenas na linguagem que entende diretamente, isto é, a Linguagem de Máquina Simpletron, ou, abreviadamente, LMS.

Código da operação	Significado
Operações de entrada/saída:	
#define READ 10	Lê uma palavra do terminal e a coloca em um local específico da memória.
#define WRITE 11	Escreve no terminal uma palavra de um local específico da memória.
Operações de carregamento/armazenamento:	
#define LOAD 20	Carrega no acumulador uma palavra de um local específico da memória.
#define STORE 21	Armazena em um local específico da memória uma palavra do acumulador.
Operações aritméticas:	
#define ADD 30	Adiciona uma palavra de um local específico da memória à palavra no acumulador (o resultado fica no acumulador).
#define SUBTRACT 31	Subtrai da palavra no acumulador, uma palavra em um local específico da memória (o resultado fica no acumulador).
#define DIVIDE 32	Divide uma palavra em um local específico da memória pela palavra no acumulador (o resultado fica no acumulador).
#define MULTIPLY 33	Multiplica uma palavra em um local específico da memória pela palavra no acumulador (o resultado fica no acumulador).
Operações de transferência de controle:	
#define BRANCH 40	Desvia para um local específico da memória.
#define BRANCHNEG 41	Desvia para um local específico da memória se o acumulador for negativo.
#define BRANCHZERO 42	Desvia para um local específico da memória se o acumulador for zero.
#define HALT 43	Término, i.e., o programa completou sua tarefa

Fig. 7.31 Códigos de operação da Linguagem de Máquina Simpletron (LMS).

O Simpletron contém um *acumulador* — um "registro especial" no qual as informações

são colocadas antes que o Simpletron as utilize em cálculos ou as examine de várias maneiras. Todas as informações no Simpletron são manipuladas em termos de *palavras.* Uma palavra é um número decimal de quatro dígitos e com sinal, como +3364, -1293, +0007, -0001 etc. O Simpletron está equipado com uma memória de 100 palavras, e faz-se referência a essas palavras por meio de seus números de localização 00,01, ...,99.

Antes de rodar um programa em LMS, devemos *carregar* ou colocar o programa na memória. A primeira instrução de qualquer programa em LMS é sempre colocada no local 00.

Cada instrução escrita em LMS ocupa uma palavra da memória do Simpletron (e assim as instruções são números decimais de quatro dígitos com sinal). Admitiremos que o sinal de uma instrução LMS é sempre positivo, mas o sinal de uma palavra de dados pode ser tanto positivo como negativo. Cada local da memória do Simpletron pode conter uma instrução, o valor de um dado usado por um programa ou uma área não-utilizada da memória (e portanto indefinida). Os dois primeiros dígitos de cada instrução LMS são o *código da operação,* que especifica a operação a ser realizada. Os códigos de operação da LMS estão resumidos na Fig. 7.31.
Os dois últimos dígitos de uma instrução LMS são o *operando,* que é o endereço do local da memória que contém a palavra à qual a operação se aplica. Agora vamos examinar vários programas simples em LMS.

Exemplo 1 Local	Número	Instrução
00	+1007	(Ler A)
01	+1008	(Ler B)
02	+2007	(Carregar A)
03	+3008	(Adicionar B)
04	+2109	(Armazenar C)
05	+1109	(Escrever C)
06	+4300	(Terminar)
07	+0000	(Variavel A)
08	+0000	(Variavel B)
09	+0000	(Resultado C)

Esse programa em LMS lê dois números do teclado, calcula sua soma e a imprime. A instrução +10 07 lê o primeiro número do teclado e o coloca no local 07 (que foi inicializado com o valor zero). A seguir, + 1008 lê o próximo número e o coloca no local 08. A instrução *carregar (load),* +2 007, coloca o primeiro número no acumulador, e a instrução *adicionar (add),* +3008, soma o segundo número ao número existente no acumulador. *Todas as instruções aritméticas LMS deixam seus resultados no acumulador.* A instrução *armazenar (store),* +2109, coloca o resultado novamente no local de memória 09 do qual a instrução *escrever (write),* +1109, obtém o número e o imprime (como um número inteiro de quatro dígitos e com sinal). A instrução *terminar (halt),* +4300, termina a execução.

Exemplo 2 Local	Número	Instrução
00	+1009	(Ler A)
01	+1010	(Ler B)
02	+200*	(Carregar A)

03	+3110	(Subtrair B)
04	+4107	(Desvio negativo para 07)
05	+1109	(Escrever A)
06	+4300	(Terminar)
07	+1110	(Escrever B)
08	+4300	(Terminar)
09	+0000	(Variavel A)
10	+0000	(Variavel B)

Esse programa em LMS lê dois números do teclado, determina o maior valor e o imprime. Observe o uso da instrução **+4107** como uma transferência condicional de controle, muito parecida com a instrução **if da** linguagem C. Agora escreva programas LMS para realizar as seguintes tarefas.

a) Use um loop controlado por um valor sentinela para ler 10 números positivos e calcular e imprimir sua soma.

b) Use um loop controlado por contador para ler sete números, alguns positivos e outros negativos, e calcule e imprima sua média.

c) Leia uma série de números e determine e imprima o maior deles. O primeiro número lido indica quantos números devem ser processados.

7.19 *(Um Simulador de Computador)* **À** primeira vista pode parecer chocante, mas neste problema você vai construir seu próprio computador. Não, você não estará unindo componentes. Em vez disso, você usará a poderosa técnica de *simulação baseada em software* para criar um *modelo de software* do Simpletron. Você não ficará desapontado. Seu simulador do Simpletron transformará seu computador em um Simpletron e você poderá realmente executar, testar e depurar os programas em LMS escritos no Exercício 7.18. Quando seu simulador Simpletron for executado, ele deve começar imprimindo:

***** Bem vindo ao Simpletron! *****
***** Por favor digite uma instrução (ou palavra *****
***** de dados) de seu programa por vez. Digitarei o *****
***** numero da posição e um ponto de interrogação *****
***** (?). Digite então a palavra para aquela posição *****
***** Digite o valor sentinela -99999 para encerrar a *****
***** digitação de seu programa. *****

Simule a memória do Simpletron com um array unidimensional **memória** com 100 elementos. Agora admita que o simulador está sendo executado, e vamos examinar o diálogo quando entrarmos com o programa do Exemplo 2 do Exercício 7.18:

00 ? +1009
01 ? +1010
02 ? +2009
03 ? +3110
04 ? +4107
05 ? +1109
06 ? +4300
07 ? +1110
08 ? +4300
09 ? +0000
10 ? +0000
11 ? -99999
***** Carregamento do programa concluido *** *** Inicio da execução do programa *****