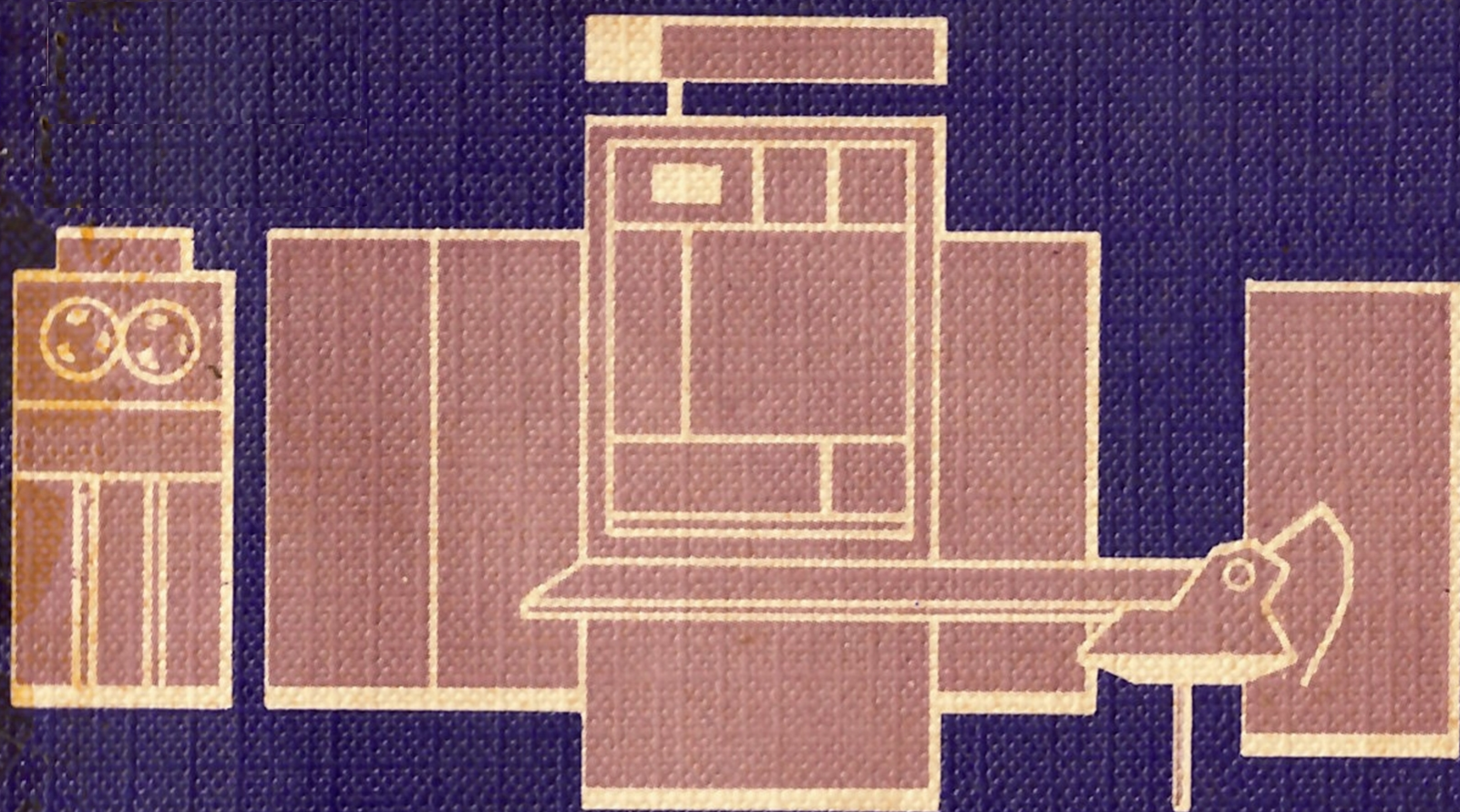


ERNESTO M. HABERKORN



Computador e Processamento de Dados



ATLAS

COMPUTADOR E PROCESSAMENTO DE DADOS

ERNESTO M. HABERKORN

O livro dá uma idéia clara e simples das funções e características dos computadores eletrônicos, sua utilização e a forma de tratar os dados e os procedimentos a serem mecânicos.

Foi escrito para pessoas que nunca tiveram contato com a matéria, não necessitando, para sua compreensão, de qualquer conhecimento prévio de outra disciplina.

Através de uma linguagem acessível, mostra o funcionamento do computador, as formas de armazenamento e transferência das informações, a programação e conceitos de processamento de dados.

Considerando-se que atualmente o computador vem sendo utilizado nas mais variadas atividades, desde as aplicações comerciais e científicas até aquelas que exercem influência diária em nossa vida particular e coletiva, e que brevemente passará a ser um equipamento de uso tão divulgado quanto o telefone ou um aparelho de televisão, é imprescindível que cada um de nós tenha ao menos uma idéia de suas funções e como usá-lo.

RESUMO DO CONTEÚDO: Os tópicos mais importantes dos doze capítulos que compõem esta edição são: conceitos e funções do

(Continua na orelha posterior)



EDITORA ATLAS S.A.
Rua Helvetia, 574/578 — CELis
Caixa Postal 7186 — Tel.: (011) 221-9144
01215 São Paulo (SP)
BRASIL

COMPUTADOR
E PROCESSAMENTO
DE DADOS

ERNESTO M. HABERKORN

COMPUTADOR E PROCESSAMENTO DE DADOS

EDITORA ATLAS S.A.

COMPUTADOR E PROCESSAMENTO DE DADOS

ERNESTO MÁRIO HABERKÖRN

Capa, diagramação e coordenação gráfica de
PAVEL GERENCER

Copyright © 1978
EDITORA ATLAS S. A.

TODOS OS DIREITOS RESERVADOS — Nos termos da Lei que resguarda os direitos autorais, é proibida a reprodução total ou parcial, bem como a produção de apostilas a partir deste livro, de qualquer forma ou por qualquer meio — eletrônico ou mecânico, inclusive através de processos xerográficos, de fotocópia e de gravação — sem permissão, por escrito, do Editor.

1.^a EDIÇÃO
4.^a Tiragem — 1978

Impresso no Brasil
Printed in Brazil

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

H118c Haberkorn, Ernesto Mário, 1943-
Computador e processamento de dados. São Paulo,
Atlas, 1978.
144 p. ilustr.

1. Computadores eletrônicos 2. Processamento e-
letrônico de dados 3. Programação (Computadores e-
letrônicos) I. Título.

76-0304

17. CDD-651.8
18. -001.6
18. -001.64
18. -001.642

Índices para catálogo sistemático:

1. Computadores : Programação : Processamento de dados 651.8(17.) 001.642(18.)
2. Computadores eletrônicos : Processamento de dados 651.8(17.) 001.64(18.)
3. Processamento de dados 651.8(17.) 001.6(18.)
4. Processamento eletrônico de dados 651.8(17.) 001.64(18.)
5. Programação de computadores : Processamento de dados 651.8(17.) 001.642(18.)

PREFÁCIO

É cada vez maior, nos dias de hoje, a necessidade de pessoal com conhecimentos de Processamento de Dados. E cada vez maior é o número de pessoas dispostas a ingressarem neste novo campo de trabalho.

Processamento de Dados é hoje matéria básica. Conhecer seus fundamentos é, além de extremamente interessante, vital para se compreenderem certos mecanismos da vida moderna.

Mas transpor esta barreira — a de saber e não saber o que é, o que faz e como funciona um computador — tem sido tarefa das mais difíceis. A maioria dos livros, cursos e manuais parte do princípio de que o leitor já está amplamente familiarizado com o assunto tratado. Na verdade, raramente isto acontece.

Processamento de Dados é matéria simples, sustentada por conceitos rigorosamente lógicos, de fácil compreensão, interligados entre si de tal forma que desprezar uma regra poderá ser fatal no processo de assimilação.

Este livro é o começo. O desenvolvimento vem com a prática e a leitura de obras mais avançadas. O estudo de uma linguagem de programação, não importa qual, é o próximo passo. A seguir, a análise de sistemas. Ou, para os mais minuciosos, a dissecação de sistemas operacionais e técnicas avançadas de processamento. Um estágio como operador de computadores permite ainda um contato mais íntimo com a máquina, sentindo-se de perto sua fantástica velocidade e extraordinária capacidade.

A multiplicidade de tipos de equipamentos, as infinitas áreas de aplicações e as várias linguagens desenvolvidas constituem tal volume de informação, que somente a especialização em determinada área poderá dar condições a um conhecimento mais profundo.

A Editora Atlas possui hoje uma seleção especial de livros — escritos por autores nacionais ou traduzidos de obras estrangeiras — que contribuem para a formação de profissionais do setor, suprindo as necessidades de nosso mercado e que são adotados em grande parte das faculdades e cursos de Processamento de Dados existentes no país.

ERNESTO HABERKORN

SUMÁRIO

1 — O CONCEITO DE PROCESSAMENTO DE DADOS

Informações iniciais	16
Transformações	16
Resultados finais	17
Processamento manual, semi-automático e automático	17
Sistemas de processamento de dados	18
Tipos de computadores eletrônicos	18
Computadores analógicos e computadores digitais	19
Computadores de primeira, segunda e terceira gerações	19
Computadores de pequeno, médio e grande portes	20
Computadores científicos e comerciais	21
História do desenvolvimento do processamento de dados	22
Herman Hollerith e as máquinas de cartão perfurado	23
As calculadoras de relés, nas décadas de 30 e 40	24
ENIAC, o primeiro computador eletrônico	24
Generalização do uso de computadores	25
Terceira geração de computadores	25

2 — FUNÇÕES DE UM SISTEMA DE PROCESSAMENTO DE DADOS

Entrada de dados	26
Armazenagem de dados	26
Processamento aritmético e lógico	27
Saída de dados	27
Controle	27
Algumas atividades típicas de um centro de processamento de dados	28

Áreas de aplicação dos computadores eletrônicos	29
Administração pública	30
Bancos	30
Companhias de seguros	30
Empresas comerciais	30
Empresas de transportes e comunicações	30
Escolas e universidades	31
Indústrias	31
Serviços de utilidade pública	31

3 — ENTRADA E SAÍDA DE DADOS

Como se origina a informação — o conceito de documento original	32
O conceito de portador de dados ou veículos de informação ..	33

4 — A ARMAZENAGEM DE INFORMAÇÕES NO COMPUTADOR

O conceito de Bit — Representação física de um bit — sistemas bi-estáveis	34
Quantidade de informações que podem ser processadas por um computador eletrônico digital	36
Dígitos, bytes e palavras	36
Blocos	36
Portador de dados	37
Registros	37
Arquivos	37
Bancos de dados	38
Alguns sistemas de memória utilizados pelos computadores eletrônicos	38
Memória de núcleos magnéticos	38
Memória de película delgada	40
Memória de tambor magnético	40
Memória de discos magnéticos	42
Disposição dos dados no disco magnético	43
O conceito de cilindro	44
Formas de organização de dados em memória de discos magnéticos	44
Modalidades de processamento de arquivos de disco magnético	45
Tempos de acesso	45
Memória de trabalho e memória de massa	45
Memória de acesso aleatória e de acesso não-aleatório	46
Alguns conceitos importantes relacionados com a armazenagem de informações nos computadores eletrônicos	47
Unidade de informação	47
Posição de memória	47

Endereço — endereçamento	48
Conteúdo de uma posição de memória	49
Áreas de memória	50
Símbolos usados para a formação dos endereços	50
Tipos de endereços	51
Endereços relativos	52
Ciclo de memória	52
Tempo de acesso	53
Memória permanente e volátil	54
Memória primária e secundária	54
Memória virtual	55

5 — SISTEMAS DE NUMERAÇÃO

Base de um sistema de numeração	56
Princípios básicos de um sistema de numeração	57
Primeiro princípio	57
Segundo princípio	57
Polinômio referente a um sistema de numeração	58
O sistema de numeração de base 10	58
Polinômio referente ao sistema de numeração decimal	59
O sistema de numeração de base 2	59
O sistema de numeração de base 16	60
Mudança de base	62
Passagem de uma base qualquer b para a base 10	62
Passagem da base 10 para uma base qualquer b	63
Aritmética de complementos	65
O conceito de complemento aritmético	65
Complemento aritmético decimal — regra prática	65
Complemento aritmético binário — regra prática	66
Adição complementada	67
Adição complementada de números decimais	67
Adição complementada de números binários	68

6 — PARTES DE UM COMPUTADOR ELETRÔNICO DIGITAL

Unidade central de processamento	69
Canais de entrada e saída	69
Dispositivos periféricos	70
Dispositivos que são somente de entrada de dados	70
Dispositivos que são somente de saída de dados	70
Dispositivos que servem tanto para a entrada como para a saída	70

7 — O CONCEITO DE PROGRAMA E O CONCEITO DE INSTRUÇÃO

Partes de uma Instrução — Tipos de Instruções	73
Código de operação	73
Endereços de operandos	73
Especificações adicionais	74
Instruções de um, dois e três endereços	74
Instrução de três endereços	74
Instrução de dois endereços	74
Instrução de um endereço	75
Categorias de Instruções	77
Seqüência de execução das instruções	79

8 — OS PRINCIPAIS CÓDIGOS UTILIZADOS PELOS COMPUTADORES ELETRÔNICOS

O Código Binário puro e suas Variantes	81
O Código Binário puro	82
O Código Binário-decimal	82
Por que bytes em vez de dígitos?	85
Códigos de 8 níveis de Bit	86
O Código EBCDIC	87
O Código ASCII	89
O Código Hexadecimal	90
Indicação de sinal em dados numéricos	92

9 — A ENTRADA E SAÍDA DE DADOS — VEÍCULOS DE ENTRADA E SAÍDA

O Cartão Perfurado	93
Descrição do cartão de 80 colunas	94
O código de cartão perfurado	95
Campos de um cartão	95
Cartões-mestre, cartões-detelhe, cartões-resumo	96
Máquinas de cartão perfurado	97
Desenho de um cartão perfurado	97
O Formulário contínuo	98
Descrição do formato de uma linha	99
A Fita de Papel Perfurada	100
Leitura de fita	102
Canal de paridade	102
A Fita Magnética	102
Canais e frames	103
Disposição dos dados na fita magnética	104
Subdivisão dos blocos em registros	105
Densidade de gravação — Velocidade de transferência	106

Velocidade instantânea e velocidade efetiva de transferência ..	106
Assinalamento da fita magnética	107
Assinalamento feito por grupos de caracteres gravados	107
Assinalamento feito por marcas físicas	107
Exemplo de formato de registro de fita	108
Documento original como veículo de entrada	109

10 — ETAPAS DA ELABORAÇÃO DE UM PROGRAMA

Levantamento da tarefa	111
Análise do problema	111
Resolução lógica do problema	112
Exemplos de fluxograma de programação	114
Programação	114
Depuração	116
Documentação	116
Linguagens simbólicas de programação	116
Linguagem de alto nível e de baixo nível	119
Principais características de uma linguagem de alto nível	119
Principais características de uma linguagem de baixo nível ..	119
Assembler	120
Cobol	121
Fortran	122
Algol	122
PL/1	122
Geradores de relatório (RPG — Report Program Generators)	123

11 — O CONCEITO DE SOFTWARE — OS PRINCIPAIS RECURSOS DE SOFTWARE

Rotinas de controle	125
Rotinas utilitárias	125
Programas de aplicação	126
A segurança do processamento eletrônico de dados	126
Exatidão de resultados	127
Verificação dos dados iniciais	127
Proteção de dados	129
Proteção dos dados gravados em fita magnética	129
Proteção dos dados gravados em disco magnético	130
Alguns conceitos de processamento de dados relacionados com o grau de simultaneidade operacional	131
Introdução	131
Funcionamento simultâneo do processador e das unidades de entrada e saída	131
O conceito de buffer	131
Interrupção de unidade periférica	132
Outros tipos de interrupção	133

Multiprogramação	133
Multiprocessamento	134
Processamento em lotes e processamento em tempo real	134
Teleprocessamento	135
Multiacesso simultâneo (time Sharing)	135
O uso das telecomunicações no processamento de dados	136

12 — O COMPUTADOR DENTRO DA EMPRESA

Vantagens que o processamento eletrônico de dados pode trazer à empresa	139
Vantagens diretas	139
Vantagens indiretas	139
Modificações que a introdução do computador produz na empresa	139
Problemas materiais da implantação do processamento eletrônico de dados numa empresa	140
Problemas humanos	140
Situação do centro de processamento de dados dentro da empresa	141
A participação humana no processamento eletrônico de dados ..	141
O trabalho do analista de sistemas	142
O trabalho do programador	142
O operador e suas responsabilidades	143
A manutenção	143
Serviços auxiliares	144
O gerente do centro de processamento de dados	144

I

O CONCEITO DE PROCESSAMENTO DE DADOS

O termo "Processamento de Dados" designa uma grande variedade de atividades que ocorrem tanto nas organizações industriais, comerciais, bancárias etc. como na vida diária de cada um de nós. Por exemplo, as seguintes atividades bastante corriqueiras podem ser consideradas como casos particulares de processamento de dados:

- Classificar várias contas a pagar, em ordem cronológica de data de vencimento.
- Fazer uma relação destas contas e somar os seus valores para obter a importância total a ser paga.
- Comparar vários orçamentos para um dado serviço, a fim de verificar qual deles é o mais vantajoso.
- Dar baixa, em nosso talão de cheques, de um cheque emitido.
- Procurar um número de telefone na lista telefônica e anotar este número em uma caderneta.

Para tentarmos definir o que é processamento de dados (e isto não é fácil) temos de ver o que existe de comum a todas estas atividades. Fazendo um estudo atento destes exemplos, constatamos que em todos eles são dadas certas informações iniciais (as contas a pagar, os orçamentos, o saldo anterior e o valor do cheque emitido, a lista telefônica) e estas informações foram sujeitas a certas transformações, com o que foram obtidas informações, ou as mesmas informações, sob uma outra forma, adequada para alguma finalidade prática.

Poderíamos então apresentar a seguinte definição:

Processamento de Dados é a atividade que consiste em transformar determinadas informações, a fim de obter outras informações ou as mesmas informações sob uma outra forma, para alguma finalidade prática.
--

Esta definição envolve três elementos essenciais do processamento de dados:

- 1 — As informações iniciais, que constituem o ponto de partida do processo.
- 2 — As transformações, que constituem o processamento propriamente dito.
- 3 — Os resultados finais, que são o objetivo do processamento.

Isto pode ser representado graficamente da seguinte maneira:



Figura 1.1.

Vamos analisar com maiores detalhes estes três elementos do processamento de dados.

INFORMAÇÕES INICIAIS

As informações iniciais são aquelas que serão sujeitas a certas transformações. Estas informações podem estar sob a forma de documentos originais: um conjunto de pedidos de mercadoria, os cheques e talões de depósito de uma agência bancária, os cartões de ponto de uma indústria etc. Outras vezes, as informações iniciais são anotações feitas em cadernos, como no caso da leitura de consumo de água ou de luz. Frequentemente, as informações iniciais se encontram sob uma forma prática para serem introduzidas rapidamente no computador, por exemplo, gravadas em fita magnética.

TRANSFORMAÇÕES

As transformações realizadas nas informações são as mais variadas possíveis. Algumas delas visam obter novas informações, enquanto outras se limitam a apresentar as mesmas informações com uma outra forma.

Exemplificando, quando somamos certo número de parcelas, obtemos uma informação nova, que é a soma destas parcelas. Por outro lado, quando pegamos algumas fichas numeradas e as colocamos em ordem numérica, estamos apenas obtendo as mesmas informações em outra ordem, ou seja, sob uma outra forma.

Um tipo importante de transformação da forma é a atividade chamada transcrição, que consiste em reproduzir a informação na mesma ordem, porém, de uma outra forma, como, por exemplo, quando fazemos uma cópia manuscrita de um parágrafo de um livro. Uma grande parte das atividades realizadas num centro de processamento de dados são atividades de transcrição.

Finalmente, a consulta de uma lista telefônica é um tipo especial de processamento, chamado busca de informações, que consiste em localizar uma informação particular no meio de uma grande quantidade de informações.

RESULTADOS FINAIS

Os resultados finais do processamento podem ser os mais variados possíveis. Em alguns casos, queremos como resultado um único valor, como a soma de numerosas parcelas. Outras vezes, desejamos uma lista ou uma tabela, como, por exemplo, uma relação das faturas emitidas por uma indústria, em determinado período. Outro tipo de resultado são mapas estatísticos: uma relação das vendas efetuadas em determinado período, classificadas por zona, por vendedor, por tipo de cliente, por tipo de produto etc.

Muitas vezes, o resultado desejado é um documento: uma conta de água, de luz ou telefone, um aviso de pagamento de imposto, uma fatura, um cheque de pagamento etc.

PROCESSAMENTO MANUAL, SEMI-AUTOMÁTICO E AUTOMÁTICO

As numerosas e variadas atividades de processamento de dados podem tanto ser realizadas manualmente como com o auxílio de máquinas.

Em muitas pequenas organizações, os trabalhos de emissão de notas fiscais, emissão de recibos, preparação de folhas de pagamento, controle de estoque etc. são feitos manualmente, usando fichas, cartões, lápis ou canetas; quando muito, certos documentos são datilografados e os cálculos são feitos ou verificados com pequenas máquinas de calcular.

Esta forma de processamento recebe o nome de processamento *manual*.

Em organizações um pouco maiores, usam-se máquinas de contabilidade para a execução de determinadas tarefas. Trata-se de um tipo de processamento que podemos chamar *semi-automático*, uma vez que ele tem algumas características do processamento manual e

outras características do processamento automático. Com efeito, na máquina de contabilidade um operador deve inserir as fichas, uma a uma, o que constitui um processo manual; por outro lado, depois de receber as fichas e os dados iniciais, a máquina de contabilidade realiza numerosas operações consecutivas sem intervenção do operador, o que constitui o aspecto automático do processo.

Finalmente, existem organizações em que o volume de trabalho é tão grande que a execução dos serviços é impraticável se ela depender de intervenção de um operador em cada ciclo de processamento (por exemplo, para a emissão de cada fatura ou para o cálculo do salário de cada funcionário). Nesse caso, utilizam-se máquinas que, uma vez condicionadas para executarem uma determinada seqüência de operações, podem repetir essa seqüência de operações inúmeras vezes, sem que haja necessidade de intervenção humana entre um ciclo e o seguinte. Esta forma de processamento chama-se processamento *automático*.

SISTEMAS DE PROCESSAMENTO DE DADOS

Qualquer máquina ou conjunto de máquinas capaz de executar o processamento de dados da maneira que acabamos de definir como processamento de dados automático recebe o nome de Sistema de Processamento de Dados.

Os primeiros sistemas de processamento de dados datam do fim do século passado, tendo sido construídos por Herman Hollerith, inventor do cartão perfurado. Tais sistemas tinham o seu funcionamento inteiramente baseado em princípios mecânicos. Até hoje são usadas versões aperfeiçoadas destes sistemas de funcionamento eletromecânico e dotados de relés eletromagnéticos. Estes sistemas recebem geralmente o nome de Sistemas Convencionais.

Desde 1951, são fabricados sistemas de processamento de dados cujo funcionamento interno se baseia em circuitos eletrônicos, infinitamente mais velozes do que os antigos sistemas eletromecânicos. Um sistema de processamento de dados cujas operações internas são realizadas por circuitos eletrônicos recebe o nome de Sistema Eletrônico de Processamento de Dados, ou, simplesmente, computador eletrônico.

TIPOS DE COMPUTADORES ELETRÔNICOS

Os computadores eletrônicos podem ser classificados em diversos tipos, conforme as características de operação, de construção e de utilização.

Com relação às características de operação, existem dois tipos básicos de computadores, os analógicos e os digitais. Quanto às características de construção, dividimos os computadores em sistemas de primeira geração, de segunda geração e de terceira geração. Falamos ainda em sistemas de porte pequeno, médio e grande. Finalmente, com relação às características de utilização, os computadores dividem-se em científicos e comerciais.

Computadores analógicos e computadores digitais

Os computadores analógicos e digitais constituem duas classes fundamentalmente diferentes quanto ao princípio de operação.

Um computador analógico representa as quantidades por meio de grandezas físicas, como, por exemplo, a intensidade de uma corrente elétrica ou o ângulo de giro de uma engrenagem; tal computador realiza as operações por meio de fenômenos físicos, e dá resultados sob forma de números, que são sempre aproximados e não exatos. Por essas razões, os computadores analógicos não podem ser utilizados em organizações bancárias, comerciais e industriais, que necessitam de valores numéricos exatos. Os computadores analógicos são usados em laboratórios de pesquisa e para aplicações científicas e tecnológicas, como, por exemplo, o estudo de redes de distribuição de energia elétrica.

Os computadores digitais representam as quantidades por meio de símbolos e executam as operações por meio de um formalismo matemático. Dão resultados exatos, apresentando-os sob a forma de caracteres (algarismos e letras). A diferença entre esses dois tipos de computadores pode ser resumida na seguinte sentença:

O Computador Analógico MEDE e o Computador Digital CONTA.

Os computadores digitais são utilizados em organizações bancárias, comerciais, industriais, governamentais etc. O presente livro trata exclusivamente dos computadores digitais.

Computadores de primeira, segunda e terceira gerações

Os computadores eletrônicos estão sendo aperfeiçoados constantemente; a cada ano aparecem novos modelos, sempre com a preocupação de aumentar a velocidade e a capacidade de processamento, reduzindo ao mesmo tempo as dimensões físicas e, principalmente, o custo de fabricação.

Distinguimos três gerações de computadores, que representam três estágios da tecnologia de fabricação.

Os sistemas de primeira geração, construídos aproximadamente entre 1951 e 1958, têm circuitos com válvulas eletrônicas e medem-se os seus tempos de operação interna em milésimos de segundos (milissegundos). (Entendemos por tempos de operação interna os tempos gastos nas operações aritméticas e lógicas.)

Os sistemas de segunda geração começam a ser construídos por volta de 1958, e sua produção vem sendo reduzida gradualmente desde 1965, à medida que vão entrando no mercado os sistemas de terceira geração. Os sistemas de segunda geração têm circuitos eletrônicos transistorizados e quase todos eles têm memória de núcleos magnéticos. Seus tempos de operação interna medem-se em milionésimos de segundo (microsegundos).

Os sistemas de terceira geração possuem microcircuitos integrados, onde todos os componentes estão fundidos num único cristal de dimensões microscópicas. Podem ter memória de núcleos magnéticos ou memórias baseadas em películas delgadas, sendo que seus tempos de operação interna são medidos em bilionésimos de segundo (nanossegundos). Entretanto, as características de construção, por si só, não bastam para caracterizar um sistema de terceira geração.

Outro aspecto muito importante é o conceito de utilização. Sob esse ponto de vista, os sistemas de terceira geração caracterizam-se pelo uso de multiprogramação em larga escala. (Chama-se multiprogramação à execução simultânea de dois ou mais programas.) Esta modalidade de utilização, por sua vez, tornou necessário o desenvolvimento de complexos conjuntos de rotinas de controle, denominados Sistemas Operacionais.

Podemos dizer, pois, que num computador de terceira geração todas as fases do trabalho são controladas por um conjunto de rotinas, e a existência de tais rotinas possibilitou o trabalho em regime de multiprogramação.

Computadores de pequeno, médio e grande portes

Nos primeiros tempos da computação eletrônica, costumava-se fazer uma classificação dos computadores em sistemas de pequeno, médio e grande portes, sendo que o "porte" se referia mais à capacidade de processamento do que propriamente às dimensões físicas do sistema.

Atualmente, essa classificação não tem mais razão de ser, porquanto um dado modelo de computador pode apresentar-se com diversas configurações, isto é, pode ser constituído de um número maior ou menor de unidades, que podem ter diferentes capacidades. Por essa razão, falamos hoje em configurações pequenas, médias e grandes.

Esta classificação é, de certo modo, arbitrária, pois quem a faz precisa tomar por base determinados parâmetros e definir o limite entre “pequeno”, “médio” e “grande”.

Podemos tomar vários parâmetros mas, para que a escolha seja válida, o parâmetro escolhido deve relacionar-se com a capacidade de processamento do sistema. Desse ponto de vista, parece-nos razoável caracterizar o porte de uma configuração pela capacidade de armazenar de sua memória interna.

Por exemplo, podemos considerar como “pequena” uma configuração com memória inferior a 32 mil bytes, como “média” uma configuração com memória entre 32 mil e 216 mil bytes, e como “grande” uma configuração com memória superior a 216 mil bytes. (Em quase todos os computadores atualmente em uso, a capacidade da memória interna é um múltiplo inteiro de 32 mil bytes, ou, mais precisamente, um múltiplo inteiro de 32 768 bytes.)

A existência de diversas configurações de um mesmo modelo de computador tem como consequência o conceito de MODULARIDADE.

A modularidade pode ser definida como a capacidade de ampliação de um sistema já instalado, pela agregação de novas unidades ou pelo aumento da capacidade das unidades existentes.

Graças a este conceito, o usuário poderá instalar, inicialmente, uma configuração correspondente às suas necessidades do momento, e depois ampliar a instalação (por exemplo, aumentar a capacidade de fita ou de disco), à medida que for aumentando o volume e/ou a complexidade do serviço.

Assim sendo, um usuário poderá começar com uma configuração pequena de determinado modelo de computador, e depois expandi-la, de modo a obter uma configuração média ou até grande.

Computadores científicos e comerciais

Não existe diferença essencial entre processamento de dados para fins científicos e comerciais. A diferença que existe entre os dois tipos de aplicações é a seguinte:

As aplicações científicas têm geralmente um pequeno volume de dados de entrada e dados de saída e cálculos numerosos e de grande complexidade; já as aplicações comerciais têm geralmente um grande volume de dados de saída e um número relativamente pequeno de cálculos.

Tendo em vista estas características gerais, um sistema destinado a aplicações científicas caracteriza-se por ter a parte de processa-

mento muito desenvolvida e uma capacidade relativamente pequena de entrada e saída, enquanto um sistema destinado a aplicações comerciais tem uma grande capacidade de entrada e saída, e a parte de processamento é mais simples do que a de um sistema científico.

Apesar disso, um computador comercial é perfeitamente capaz de executar um cálculo científico, e um computador científico pode perfeitamente executar um serviço comercial. É óbvio, entretanto, que cada um desses tipos de computador funcionará com maior eficiência quando executar a espécie de serviço para a qual ele está mais bem equipado.

Notemos, finalmente, que há casos de computadores executando ambos os tipos de serviço. Por exemplo, numa indústria, um computador normalmente usado para folha de pagamento, faturamento etc. poderá ocasionalmente colaborar com o departamento de engenharia; por outro lado, um computador científico instalado numa universidade pode proporcionar à mesma uma renda, executando serviços comerciais.

HISTÓRIA DO DESENVOLVIMENTO DO PROCESSAMENTO DE DADOS

O processamento de dados, em seu sentido mais amplo, é tão antigo quanto a humanidade. Tanto o pastor grego que contava as suas ovelhas, como o agrimensurador egípcio que demarcava anualmente as propriedades, após as cheias do Nilo, estavam realizando processamento de dados. No que diz respeito a recursos auxiliares, podemos citar o ábaco, utilizado pelos romanos, há cerca de 2000 anos. Nesse instrumento, os números eram representados por pequenas pedrinhas de calcário denominadas "calculi", vindo daí a palavra "cálculo".

Nos tempos modernos, podemos considerar como precursor das máquinas de calcular o matemático e filósofo francês Blaise *Pascal*, que construiu em 1645 uma máquina capaz de somar e subtrair números de 8 algarismos, baseado em engrenagens. Também o matemático e filósofo alemão Gottfried von *Leibniz* construiu uma máquina de calcular em 1677. Por volta de 1830, começa na Europa a fabricação em série de máquinas de calcular mecânicas. Em 1801, ocorre um fato que seria de grande importância para o futuro desenvolvimento do processamento de dados; o tecelão francês Joseph-Marie *Jacquard* constrói um tipo de tear comandado por um conjunto de cartões perfurados enfileirados. É a primeira utilização prática de cartões perfurados.

Como precursor do processamento automático de dados podemos citar o matemático inglês Charles *Babbage*, nascido em 1792. Foi ele

o primeiro a propor a construção de uma “Calculadora Automática Seqüencial”, isto é, uma máquina que fosse capaz de executar automaticamente uma seqüência predeterminada de operações. Em 1834, ele construiu um modelo da sua máquina, superando as dificuldades financeiras, bem como os problemas técnicos decorrentes do estágio primitivo da tecnologia de sua época.

É importante notar que, embora Babbage não tenha conseguido levar avante seus trabalhos, devido a problemas técnicos e financeiros, ele já teve a visão dos princípios gerais em que se baseiam os modernos sistemas de processamento de dados, a saber:

- a) A máquina deveria ser comandada por um programa elaborado previamente.
- b) A máquina deve possuir memória para armazenar resultados intermediários.
- c) Os cálculos devem ser realizados em ciclos, com utilização de resultados intermediários obtidos anteriormente.
- d) A saída deve ter uma forma adaptada às características da máquina.

Passemos agora a historiar o desenvolvimento do processamento de dados em tempos mais recentes ou, mais precisamente, de 1885 para cá.

Herman Hollerith e as Máquinas de Cartão Perfurado

Um importante desenvolvimento na técnica de processamento de dados é devida a Herman Hollerith (1860-1929), engenheiro americano, filho de imigrantes alemães. Hollerith era funcionário do Departamento Federal de Estatística dos EUA, sendo esta repartição encarregada de fazer um Censo Demográfico do país, de 10 em 10 anos. O Recenseamento de 1880 havia acusado uma população de aproximadamente 55 milhões de habitantes, sendo que a apuração ocupou 500 funcionários durante 7 anos. Impressionado com a magnitude da tarefa do Censo seguinte, Hollerith procurou meios de mecanizar os trabalhos de apuração, utilizando a idéia do cartão perfurado, proposta por Jacquard. A construção das máquinas de cartão perfurado foi iniciada em 1885, e foram utilizadas no Recenseamento de 1890, com grande sucesso. Basta dizer que, graças a elas, os trabalhos de apuração foram concluídos em cerca de um ano, com apenas 43 funcionários.

Não demorou muito e as máquinas de cartão perfurado começaram a ser utilizadas para as mais diversas aplicações, não só em repartições, mas também em organizações comerciais, industriais, bancárias etc.

Nas primeiras décadas do século XX, as máquinas de cartão foram muito aperfeiçoadas, sendo criados modelos de máquinas capazes de realizar operações aritméticas elementares. Essas calculadoras eram eletromecânicas e, portanto, bastante lentas.

As calculadoras de relés, nas décadas de 30 e 40

A partir de 1930, foram feitos vários esforços para construir calculadoras mais rápidas e mais versáteis, com funcionamento baseado em relés eletromagnéticos.

Em 1935, Konrad Zuse começa a desenvolver calculadoras automáticas de relés, na Alemanha. Foram construídos os modelos Z1, Z2, Z3 e Z4. O modelo Z1, construído em 1941, foi a primeira calculadora automática a trabalhar sob o controle de um programa perfurado numa fita de papel.

Em 1937, nos Estados Unidos, Howard Aiken inicia o desenvolvimento de uma calculadora de relés, na Universidade de Harvard. A primeira calculadora automática, controlada por programa, construída por Aiken, ficou pronta em 1944 e recebeu o nome de Mark I.

ENIAC, o primeiro computador eletrônico

Em 1943, John P. Eckert e John W. Mauchly iniciaram os trabalhos de construção de um computador com funcionamento baseado em circuitos eletrônicos. Este trabalho, realizado na Universidade da Pennsylvania, foi concluído em 1946. O primeiro computador eletrônico recebeu o nome de ENIAC (Electronic Numeric Integrator And Calculator). Essa máquina tinha 18 000 válvulas eletrônicas, e utilizava o sistema de numeração binário para a realização dos cálculos, enquanto o Mark I trabalhava com o sistema de numeração decimal.

O computador eletrônico ENIAC representava um imenso progresso em relação aos computadores de relés que o precederam. Basta dizer que o Mark I gastava 6 segundos para multiplicar dois fatores de 10 algarismos, enquanto o ENIAC realizava o mesmo cálculo em 0,003 segundos.

Logo depois do ENIAC, foi construído outro computador eletrônico chamado EDVAC, no qual foi utilizada pela primeira vez a idéia da programação interna, proposta por John von Neumann. Esta idéia (que é adotada hoje em dia em quase todos os computadores) consiste em armazenar o programa, em forma codificada, dentro da memória do computador, e não num veículo externo como uma fita perfurada. Graças ao conceito de programação interna, o programa pode executar instruções ou grupos de instruções em qualquer ordem e repeti-las quantas vezes for necessário.

Generalização do uso de computadores

O ENIAC e o EDVAC haviam sido modelos experimentais, e não foram explorados comercialmente. A partir da década de 1950, várias empresas começaram a fabricar computadores em série. Em 1951, foi anunciado o computador UNIVAC I, iniciando a primeira geração de computadores. O primeiro computador UNIVAC I foi entregue ao Bureau of Census, a mesma repartição em que havia surgido o cartão perfurado, 60 anos antes. Em 1951, a International Business Machines (IBM) lança o seu primeiro computador eletrônico, chamado IBM 701.

Em meados da década de 1950, aparecem os primeiros computadores inteiramente transistorizados, que assinalam o início da segunda geração de computadores. Em 1958, a Siemens, na Alemanha, lança o seu sistema Siemens 2002. O representante mais típico da segunda geração de computadores é o IBM 1401.

O ano de 1960 marca um importante desenvolvimento da técnica de utilização de computadores. Neste ano, é criada nos Estados Unidos uma linguagem universal de programação para fins comerciais, por um grupo de trabalho formado de representantes do governo dos E.U.A., de fabricantes e de usuários de computadores. Esta linguagem universal recebeu o nome de COBOL (COmmon Business Oriented Language).

Terceira geração de computadores

Em meados da década de 1960, inicia-se a terceira geração de computadores, caracterizada pelo uso de circuitos monolíticos integrados e por avançados sistemas operacionais.

2

FUNÇÕES DE UM SISTEMA DE PROCESSAMENTO DE DADOS

Qualquer sistema eletrônico de processamentos de dados, isto é, qualquer computador eletrônico, apresenta as mesmas cinco funções básicas, que são:

- 1 — entrada de dados;
- 2 — armazenagem de dados;
- 3 — processamento aritmético e lógico;
- 4 — saída de dados;
- 5 — controle.

ENTRADA DE DADOS

A primeira etapa do processamento eletrônico de dados consiste em introduzir no sistema as informações que ele deverá processar.

Esta é a função de ENTRADA, que é executada por alguma das unidades periféricas agregadas ao processador: leitoras de cartões, leitoras de fita de papel, unidades de fita magnética, unidades de disco etc.

Na maioria dos casos, as informações são introduzidas nessas unidades periféricas sob forma codificada. Por exemplo, no cartão perfurado, cada caráter é representado por perfurações feitas em determinadas posições. As unidades de entrada lêem as informações codificadas, interpretam tais informações e as encaminham à memória do sistema.

ARMAZENAGEM DE DADOS

A armazenagem ou memorização de dados é a função que consiste em reter os dados dentro do sistema, para que o computador possa

utilizá-los quando precisar deles. Essa memorização pode ser feita na memória principal, constituída, geralmente, de núcleos magnéticos ou sistemas de película delgada, e na memória auxiliar de grande capacidade, constituída, geralmente, de discos ou tambores magnéticos.

O tempo de permanência de uma informação dentro da memória pode variar desde frações de segundo até horas ou mesmo dias.

PROCESSAMENTO ARITMÉTICO E LÓGICO

O processamento aritmético e lógico é o conjunto das operações necessárias para converter os dados iniciais nos resultados finais desejados. Esta função é executada pelos circuitos eletrônicos (circuitos aritméticos e lógicos) da Unidade Central de Processamento, sendo a seqüência das operações determinada pelo programa.

O processamento envolve não apenas cálculos matemáticos, mas também operações de classificação, alteração, inclusão e retirada de dados, comparações, verificações etc.

SAÍDA DE DADOS

O sistema de processamento de dados guarda na memória os resultados do processamento. Evidentemente, não podemos ver o que se passa dentro da memória. Por conseguinte, é preciso fornecer ao usuário, de algum modo, os resultados obtidos.

Esta é a função de SAÍDA, executada por algumas das unidades periféricas agregadas ao computador: impressoras, perfuradoras de cartões, unidades de fita magnética, unidades de disco.

Na impressora, as informações são apresentadas de maneira visível; nas demais unidades citadas acima, as informações de saída estão em forma codificada, a ser interpretada posteriormente.

Existem ainda unidades de apresentação visual (terminais de vídeo), que projetam as informações num tubo de raios catódicos.

CONTROLE

O controle é a coordenação de todas as funções do sistema; graças a ela, as várias unidades do sistema operam harmonicamente e realizam as diversas operações na seqüência correta. A função de controle é efetuada pelos circuitos de controle, sob o comando do programa.

Note-se que a essas cinco funções não correspondem, necessariamente, cinco móveis ou unidades. Em certos casos, uma unidade abriga os circuitos e dispositivos pertencentes a mais de uma função.

Por exemplo, a maioria dos computadores possui uma unidade chamada unidade central de processamento ou processador central, que executa as funções de processamento e controle, podendo também alojar a memória do sistema. Por outro lado, na maioria dos sistemas, existe mais de uma unidade para a função de entrada e mais de uma unidade para a função de saída de dados.

As relações entre as funções de um sistema de processamento de dados podem ser representadas esquematicamente pelo desenho que segue.

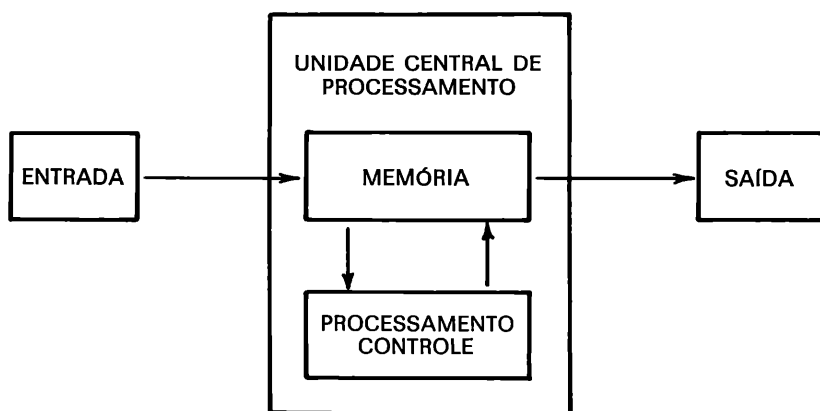


Figura 2.1.

ALGUMAS ATIVIDADES TÍPICAS DE UM CENTRO DE PROCESSAMENTO DE DADOS

Um centro de processamento de dados é uma organização dotada de equipamento convencional e/ou eletrônico de processamento de dados, podendo ser um departamento de uma empresa ou de um órgão governamental, ou, então, uma organização independente, criada para prestar serviços a outras empresas.

As atividades de tal centro são as mais variadas possíveis. Abaixo, daremos uma relação de algumas das atividades mais comuns, dividindo-as em dois grupos: as atividades de transcrição, que não alteram a natureza nem a ordem das informações, e as atividades de processamento propriamente ditas, que podem alterar a natureza ou a ordem, ou ambas.

Entre as atividades de transcrição, citaremos as seguintes:

- Perfurar em cartões os dados tirados de algum documento original (um pedido, um cheque, um talão de depósito, um cartão de ponto, uma requisição de material etc.).
- Perfurar em fita de papel os dados provenientes de um documento original.
- Gravar em fita magnética os dados perfurados em cartões.
- Gravar em fita magnética os dados perfurados em fita de papel.
- Imprimir num formulário os dados perfurados em cartões. (Listagem de cartões.)
- Imprimir num formulário os dados gravados em fita magnética. (Listagem de fita magnética.)

Agora, apresentaremos alguns exemplos de atividades de processamento:

- Emissão de faturas, contas e avisos de pagamento.
- Emissão de folhas de pagamento e de cheques de pagamento.
- Emissão de extratos de conta corrente.
- Atualização de saldos de conta corrente.
- Preparação de relações de duplicatas vencidas e não pagas.
- Preparação de borderô (relação de duplicatas que serão entregues a um banco para desconto ou cobrança).
- Elaboração de estatísticas (estatísticas de venda, de movimento de passageiros, de consumo de energia elétrica etc.).
- Classificação de dados, em ordem numérica ou alfabética.

Estas são atividades de execução de serviços já implantados, isto é, serviços cujas dificuldades iniciais já foram superadas, e que são executados regularmente. Além disso, uma parte do tempo do centro é dedicada à implantação de novos serviços: teste de novos programas, estudos de tempo etc.

ÁREAS DE APLICAÇÃO DOS COMPUTADORES ELETRÔNICOS

Os computadores eletrônicos digitais podem ser aplicados nos mais diversos setores da atividade humana. Diariamente, descobrem-se novas aplicações, e, hoje em dia, praticamente não existe um campo de atividade que não tenha sido influenciado, direta ou indiretamente, pela existência dos computadores eletrônicos.

Damos, em seguida, uma relação dos tipos de organizações que fazem uso de computadores com maior frequência, e de alguns tipos de serviços que os computadores executam dentro de tais organizações.

Administração Pública

- Emissão de avisos de taxas e de impostos
- Controle de arrecadação
- Cadastro de contribuintes
- Folha de pagamento
- Planejamento econômico
- Recenseamento e estudos estatísticos

Bancos

- Controle de contas correntes
- Cálculo de juros
- Controle de carteiras
- Estudos econômicos
- Compensação de cheques

Companhias de Seguros

- Emissão de apólices
- Controles financeiros
- Cálculos atuariais (Estatística de seguros)

Empresas Comerciais

- Emissão de faturas
- Controle de cobrança
- Estatísticas de vendas
- Planejamento de compras
- Controles financeiros

Empresas de Transportes e Comunicações

- Reserva de passagens
- Organização de horários
- Controle do tráfego de veículos
- Cálculo de tarifas
- Contabilidade de tráfego mútuo
- Produção de mapas estatísticos
- Controle do tráfego de mensagens
- Controle da manutenção dos veículos
- Controle do estoque de peças

Escolas e Universidades

- Controle de frequência
- Controle de notas
- Correção de testes
- Organização de horários
- Pesquisas científicas
- Instrução auxiliada pelo computador (C.A.I.)

Indústrias

- Controle de estoque
- Folha de pagamento
- Planejamento de produção
- Controle de qualidade
- Previsão de vendas
- Cálculo de custo
- Faturamento

Serviços de Utilidade Pública

- Emissão de contas
- Controle de arrecadação
- Estatísticas de consumo
- Controles econômicos
- Planos de expansão
- Folha de pagamento
- Almoxarifado
- Simulação de redes de distribuição

3

ENTRADA E SAÍDA DE DADOS

COMO SE ORIGINA A INFORMAÇÃO — O CONCEITO DE DOCUMENTO ORIGINAL

Podemos dizer que uma informação é um registro de um fato dinâmico ou estático. Por exemplo, uma contagem física do estoque de um documento antigo nos dá uma informação sobre um fato estático. Por outro lado, a saída de uma certa quantidade de artigos do estoque é uma informação do tipo dinâmico, pois corresponde à alteração de um valor.

Na maioria dos casos, a informação é registrada, inicialmente, em um pedaço de papel ou cartolina, que vai constituir o documento original, ponto de partida da cadeia de operações chamada processamento de dados. Como exemplos de documentos originais podemos citar: cheques, talões de depósitos, pedidos de mercadoria, cartões de ponto, declarações de renda etc.

Em alguns casos especiais, o documento original pode ser diretamente introduzido no sistema de processamento de dados, como ocorre, por exemplo, com um cheque dotado de caracteres magnéticos. Na maioria das vezes, entretanto, os dados do documento original devem ser transcritos num outro meio (veículo de informação) para, então, serem introduzidos no computador.

Nos últimos anos, vem ganhando importância cada vez maior um processo de trabalho que dispensa inteiramente o documento original: é o dos terminais remotos diretamente ligados a um computador, localizado numa outra parte da cidade ou mesmo em uma outra cidade. Isso é muito usado para o serviço de reservas de passagens aéreas. Neste caso, a agência de viagens tem um terminal de comunicações, dotado de um teclado e ligado ao Centro de Processamento de Dados

da Empresa Aérea, através de algum tipo de linha de comunicação. O funcionário registra no teclado os dados referentes ao lugar desejado, que são transmitidos diretamente ao computador, sem a intervenção de um documento. (Uma vez concluída a operação, o terminal de comunicação geralmente imprime numa folha de papel a confirmação de reserva.)

O CONCEITO DE PORTADOR DE DADOS OU VEÍCULO DE INFORMAÇÃO

Normalmente, uma informação está registrada em algum meio físico: uma folha de papel, um quadro-negro, um letreiro de madeira ou de plástico etc. Esse meio material portador da informação chama-se veículo de informação ou portador de dados.

No caso dos computadores eletrônicos, chama-se veículo de entrada aquele que serve para introduzir informações no sistema, e veículo de saída aquele que recebe as informações produzidas pelo sistema. Em alguns casos, o mesmo veículo serve tanto para a entrada como para a saída de dados. Por exemplo, existem unidades capazes de ler informações em determinados campos de um cartão perfurado e posteriormente perfurar outras informações em outros campos do mesmo cartão.

Muitas vezes, o mesmo tipo de veículo (mas não o mesmo veículo) serve tanto para a entrada como para a saída de dados. Por exemplo, um sistema pode ler dados gravados num carretel de fita magnética, e gravar resultados num outro carretel. Entretanto, nunca se pode gravar no carretel de fita que está sendo lido.

Finalmente, existem veículos que ou se prestam apenas para a entrada ou apenas para a saída de dados. A seguir, apresentamos uma tabela com os veículos de informação mais utilizados pelos computadores eletrônicos.

TABELA DOS PRINCIPAIS VEÍCULOS DE INFORMAÇÃO

NOME DO VEÍCULO	SERVE PARA A ENTRADA?	SERVE PARA A SAÍDA?
CARTÃO PERFURADO	SIM	SIM
FITA DE PAPEL PERFURADA	SIM	SIM
FITA MAGNÉTICA	SIM	SIM
DISCO MAGNÉTICO	SIM	SIM
FORMULÁRIO CONTÍNUO	NÃO	SIM
DOCUMENTO ORIGINAL (LEITURA ÓTICA OU CARACTERES MAGNÉTICOS)	SIM	NÃO

4

A ARMAZENAGEM DE INFORMAÇÕES NO COMPUTADOR

Em geral, as informações armazenadas nos computadores digitais são constituídas de caracteres. Em processamento de dados, um “caráter” pode ser um algarismo, uma letra, um sinal de pontuação ou um símbolo matemático, sendo que essas últimas duas categorias são englobadas na denominação “Símbolos Especiais”. Portanto, um computador pode armazenar letras, símbolos especiais e algarismos.

Entretanto, tais caracteres, para poderem ser armazenados na memória do computador, devem ser submetidos a um processo de codificação, que reduz cada caráter a uma combinação de dois símbolos fundamentais.

Com isto, a construção do computador simplifica-se bastante, pois, em vez de ter que distinguir várias dezenas de símbolos, o computador terá apenas que reconhecer dois símbolos fundamentais. Um exemplo conhecido de um sistema de codificação baseado em dois símbolos fundamentais é o Código Morse, usado nos primeiros tempos do telégrafo e que representa as letras e os algarismos por uma combinação de pontos e traços. (Um ponto era um sinal breve e um traço era um sinal longo.)

A codificação de informações nos computadores envolve o importante conceito de bit, sobre o qual falaremos a seguir.

O CONCEITO DE BIT — REPRESENTAÇÃO FÍSICA DE UM BIT — SISTEMAS BI-ESTÁVEIS

Nos computadores eletrônicos digitais, os dois símbolos fundamentais, com os quais se formam os demais símbolos, são geralmente designados pelo algarismo 0 e 1. Esta escolha foi feita porque os algarismos 0 e 1 apresentam a importante propriedade de serem os

únicos algarismos de um sistema de numeração de base 2. Por essa razão, os símbolos fundamentais 0 e 1 recebem o nome de “Dígitos Binários”. Sua designação mais usual é a palavra BIT, contração do termo inglês “binary digit”. Assim, os dois símbolos fundamentais são normalmente designados pelos nomes “Bit 0” e “Bit 1”.

O bit é a menor quantidade de informações que se pode armazenar no computador. A reunião de um certo número de bits forma um dígito ou uma palavra.

A cada bit armazenado na memória corresponde um sistema físico dentro do computador. Uma vez que a codificação mencionada anteriormente reduziu a dois o número de símbolos a serem reconhecidos, o sistema físico em questão precisa representar somente esses dois símbolos.

Para este fim, usam-se os sistemas bi-estáveis, que podem ser definidos da seguinte maneira:

“Chama-se sistema bi-estável um sistema físico que pode encontrar-se em dois estados opostos e mutuamente exclusivos, e tal que a passagem de um estado ao outro seja um fenômeno reversível.”

Um exemplo muito simples de sistema bi-estável é uma lâmpada incandescente, que pode estar acesa ou apagada. Estes são dois estados opostos e mutuamente exclusivos; além disso, a transformação é reversível, pois podemos acender uma lâmpada que está apagada e apagar uma lâmpada que se encontra acesa. Um outro exemplo de sistema bi-estável é um interruptor elétrico, que pode encontrar-se aberto ou fechado.

Em todos os computadores eletrônicos digitais, a armazenagem de informações está baseada em sistemas bi-estáveis.

A um dos dois estados de um sistema bi-estável associa-se arbitrariamente o bit 0, e ao outro estado o bit 1. Por exemplo, uma lâmpada incandescente poderia representar um bit 0 quando apagada e um bit 1 quando acesa.

Na prática, os atuais computadores eletrônicos digitais armazenam as informações usando sistemas bi-estáveis, baseados em fenômenos magnéticos e eletromagnéticos.

Os sistemas usados com maior frequência para esta finalidade são os seguintes:

- Um pequeno trecho de uma superfície metálica que pode ou não estar imantado.
- Uma pequena argola metálica, que pode estar imantada em sentido igual ao do movimento dos ponteiros de um relógio (sentido horário) ou no sentido oposto (sentido anti-horário).
- Uma delgadíssima película metálica, plana ou cilíndrica, que pode estar imantada em dois sentidos opostos.

QUANTIDADE DE INFORMAÇÕES QUE PODEM SER PROCESSADAS POR UM COMPUTADOR ELETRÔNICO DIGITAL

A menor quantidade de informação, do ponto de vista físico, bem como do ponto de vista lógico, é o BIT. Com efeito, o bit é a menor quantidade de informação que pode ser armazenada na memória de um computador; por exemplo, um núcleo magnético. Por outro lado, o bit é também a menor quantidade de informação que pode ser processada; por exemplo, podemos transformar um bit 0 em bit 1 ou vice-versa.

Do ponto de vista físico, temos acima do bit o dígito, o byte e a palavra (que ocupam o mesmo nível de hierarquia), o bloco e o portador de dados. Temos, pois, a seguinte HIERARQUIA FÍSICA:

- BIT
- DÍGITO, BYTE, PALAVRA
- BLOCO
- PORTADOR DE DADOS

Dígitos, bytes e palavras

Normalmente, em dado instante, um computador tem acesso a mais de um bit. O conjunto de bits a que o computador tem acesso simultaneamente pode ser considerado como a unidade básica de informação, do ponto de vista do funcionamento interno do sistema.

Nos computadores de construção mais antiga (computadores de segunda geração), esta unidade básica é o DÍGITO ou CARÁTER, formado de 6 bits, acrescidos ou não de um bit de verificação. Nos sistemas de terceira geração, apareceu uma nova unidade de informação, chamada BYTE, constituída de 8 bits, acrescidos ou não de um bit de verificação. Nestes mesmos sistemas, usa-se também o conceito de PALAVRA, sendo uma palavra formada de 32 bits. Tanto o DÍGITO como o BYTE e a PALAVRA têm uma coisa em comum: tratam-se de certo número de bits aos quais o sistema tem acesso no mesmo instante.

No quadro acima, estes três termos foram escritos na mesma linha, por representarem quantidades de informação conceitualmente equivalentes, que diferem entre si apenas pelo número de bits envolvido.

Blocos

Um bloco é a quantidade de informações que uma unidade de entrada introduz na memória em decorrência de uma única instrução de entrada, ou que a memória transmite a uma unidade de saída em decorrência de uma única instrução de saída. Por exemplo, as informações de um cartão perfurado constituem um bloco, assim como as

informações de uma linha produzida pela impressora. No trabalho com fitas magnéticas, um bloco é o conjunto de caracteres lidos ou gravados em decorrência de uma única instrução dada a uma unidade de fita magnética.

O bloco também recebe o nome de registro físico.

Portador de dados

Um portador de dados é algum objeto físico que contenha um conjunto de blocos, como, por exemplo, um carretel de fita magnética, um conjunto de discos magnéticos (disc-pack), um estojo de fichas magnéticas etc.

Do ponto de vista lógico, acima de bit existe o registro, o arquivo e o banco de dados. Temos, portanto, a seguinte HIERARQUIA LÓGICA:

- BIT
- REGISTRO
- ARQUIVO
- BANCO DE DADOS

Registros

Um REGISTRO ou ITEM (do inglês, *record*) pode ser definido como um conjunto de informações relacionadas com um identificador comum. Por exemplo, o conjunto dos dados referentes a dado cliente (nome, endereço, número de inscrição) constitui um registro. Outro exemplo de registro é o conjunto das informações referentes a uma transação bancária: número da conta, data da transação, valor e código de transação.

O REGISTRO também recebe o nome de REGISTRO LÓGICO.

Note-se que um bloco, ou registro físico, pode conter mais de um registro lógico. Reciprocamente, pode ocorrer que sejam necessários vários registros físicos (por exemplo, vários cartões perfurados) para conter as informações correspondentes a um único registro lógico.

Arquivos

Um AROUIVO DE DADOS (do inglês, DATA FILE) é uma certa quantidade de registros a serem processados ou que já foram processados. Quando o arquivo é constituído de dados iniciais, ele se chama ARQUIVO DE ENTRADA (do inglês, INPUT FILE), e, quando o arquivo é constituído de dados já processados, ou seja, de resultados do processamento, ele se chama ARQUIVO DE SAÍDA (do inglês, OUTPUT FILE).

Por exemplo, num serviço de atualização de saldos de contas correntes, o conjunto dos saldos anteriores constitui um arquivo de entrada; o conjunto dos dados sobre as transações ocorridas (retira-

das e depósitos) constitui um outro arquivo de entrada; e o conjunto dos saldos atualizados constitui um arquivo de saída.

O conceito de arquivo está associado ao conceito de unidade de entrada e saída. Um arquivo de entrada é um conjunto de registros introduzidos no sistema através de determinada unidade de entrada, e um arquivo de saída é um conjunto de registros que o sistema produz em determinada unidade de saída.

Bancos de dados

Um certo número de arquivos referentes a determinada área de aplicação constitui um BANCO DE DADOS. Por exemplo, o Ministério da Fazenda pode formar um banco de dados, constituído de informações sobre os contribuintes, os impostos lançados e pagos etc.

ALGUNS SISTEMAS DE MEMÓRIA UTILIZADOS PELOS COMPUTADORES ELETRÔNICOS

Como vimos, as informações dentro do computador estão codificadas em forma de grupos de bits (portanto, dígitos, bytes ou palavras), sendo cada bit representado fisicamente por um sistema bi-estável. Os sistemas de memória utilizados com mais frequência são os seguintes: núcleo magnético, película delgada, tambor magnético e disco magnético.

Memória de núcleos magnéticos

A memória de núcleos magnéticos é atualmente o tipo de memória mais utilizado. Um núcleo magnético é uma pequena argola de uma liga metálica denominada ferrite. Esta argola pode estar imantada em dois sentidos opostos, como se vê na figura que segue.

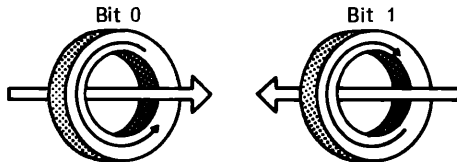


Figura 4.1.

Quando o núcleo está imantado em determinado sentido, ele representa um bit 0; quando imantado no sentido oposto, representa um bit 1.

Os núcleos magnéticos estão montados num reticulado de fios, denominado *plano de memória*.

A figura a seguir mostra um trecho de um plano de memória.

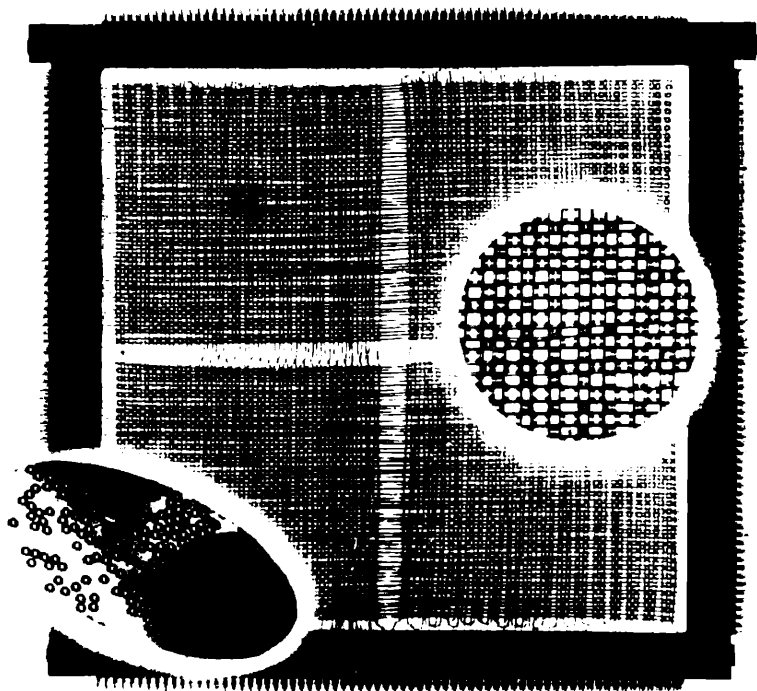
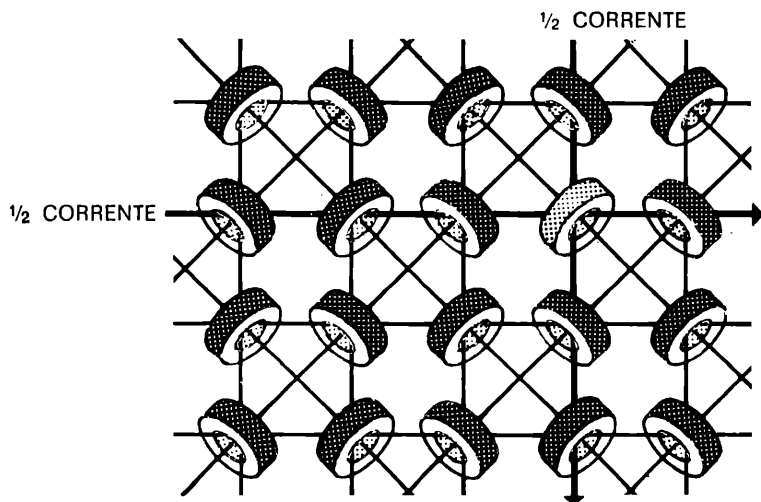


Figura 4.2.

Vários planos de memória superpostos formam um *banco de memória*, também chamado *módulo de memória*. A quantidade de dígitos ou palavras que podemos armazenar em um bloco de memória varia conforme o modelo de computador. Geralmente, o número que exprime esta capacidade é uma potência de 2, sendo $2^{12} = 4096$ um valor típico para a capacidade de armazenagem de um banco de memória. Muitos computadores têm memória do tipo *modular*, isto é, a capacidade de memória de um computador já instalado pode ser aumentada mediante a agregação de novos módulos de memória.

Memória de película delgada

Outro sistema de memorização de dados é o que utiliza delgadíssimas camadas metálicas vaporizadas ou eletrodepositadas sobre alguma base. Estas camadas estão divididas em pequeníssimas regiões, sendo que cada uma dessas regiões pode estar imantada em dois sentidos opostos. Cada região armazena um bit 0 ou um bit 1, conforme o seu sentido de imantação.

Existem várias formas de memória de película delgada: memória de película delgada plana, de haste magnética e de fio magnético.

A memória de película delgada plana é constituída de pequenos pontos metálicos vaporizados sobre uma base plana.

A memória de haste magnética é constituída de pequenas hastes metálicas envolvidas por uma película delgada.

A memória de fio magnético é constituída de uma fina camada metálica eletrodepositada sobre um fio cilíndrico.

Memória de tambor magnético

A memória de tambor magnético é constituída de um cilindro metálico que gira com alta velocidade, e cuja superfície lateral é revestida de um material capaz de reter informações sob a forma de marcas magnéticas.

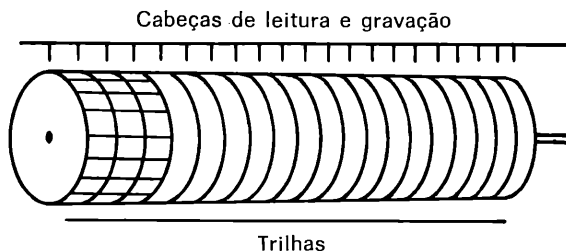


Figura 4.3 — A memória de tambor magnético

A superfície lateral do cilindro é dividida idealmente em faixas denominadas *trilhas*, que, por sua vez, estão divididas em pequenos trechinhos retangulares, como se vê na figura anterior.

Cada um destes trechinhos armazena um bit; a presença de uma marca magnética num trechinho de superfície do tambor é interpretada como bit 1, e a ausência de marca magnética no trecho da superfície representa um bit 0.

Cada trilha armazena um grande número de caracteres e cada tambor magnético apresenta um grande número de trilhas. Assim sendo, a capacidade de armazenagem de um tambor é enorme, chegando a milhões de caracteres ou mesmo a dezenas de milhões de caracteres.

A leitura e gravação de informações na superfície lateral do tambor é feita pelas cabeças de leitura e gravação. Em alguns modelos de tambores magnéticos, existe uma cabeça de leitura e gravação para cada trilha; em outros modelos, cada cabeça de leitura e gravação deve atender a várias trilhas.

Nesse último caso, antes de cada leitura ou gravação de informações deve haver movimentação da cabeça de leitura e gravação, para a seleção da trilha desejada.

Na realidade, o acesso a dada informação gravada num tambor magnético envolve quase sempre dois movimentos: um movimento de posicionamento da cabeça em dada trilha, e depois um movimento de rotação do tambor, para que o trecho desejado da trilha fique abaixo da cabeça. Esse movimento de rotação é ilustrado pela figura abaixo:

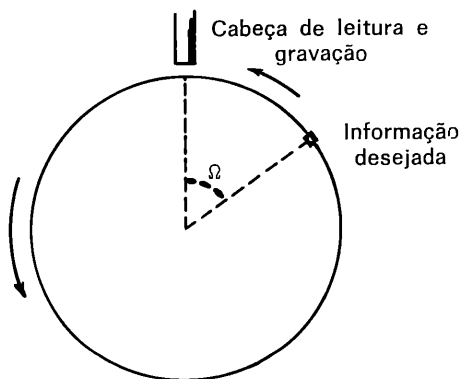


Figura 4.4.

O tempo de espera devido a movimentos mecânicos recebe o nome de *latência*. O problema da latência existe na memória de tambor, bem como na memória de disco, da qual falaremos a seguir.

Memória de discos magnéticos

A memória de discos magnéticos é constituída de um conjunto de discos metálicos giratórios, denominado "disc-pack". As faces dos discos estão revestidas de uma substância capaz de armazenar informações sob a forma de marcas magnéticas permanentes. Cada face de disco está dividida em trilhas concêntricas e as trilhas, por sua vez, estão divididas em registros. Esta divisão é ilustrada pela figura seguinte:

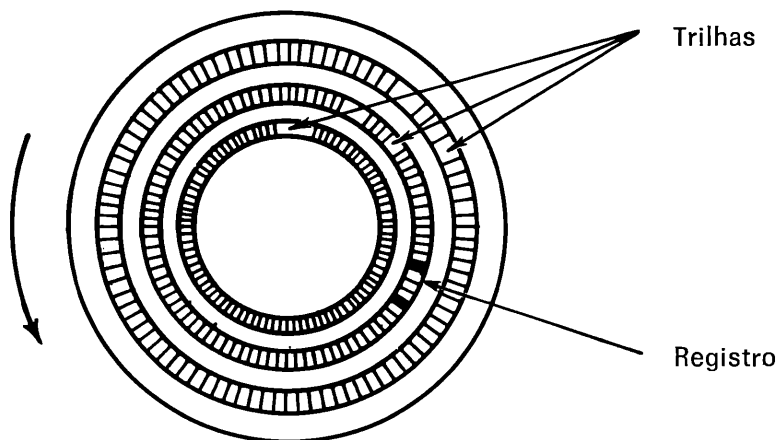


Figura 4.5.

A cabeça de leitura e gravação está presa a um braço que se movimenta na direção radial para selecionar determinada trilha. Esse movimento é ilustrado pela figura 4.6.

Um disc-pack é, geralmente, um conjunto de 6 discos, com 10 faces de gravação, ou um conjunto de 11 discos, com 20 faces de gravação. (As duas faces externas da pilha não aceitam gravação de dados.) Uma característica muito importante do disc-pack é o fato de ser removível. Desta maneira, o usuário pode armazenar em discos uma quantidade praticamente ilimitada de informações, usando o número de "disc-packs" que quiser, e introduzindo na unidade de discos, em determinado momento, aquele disc-pack que tiver as informações desejadas.

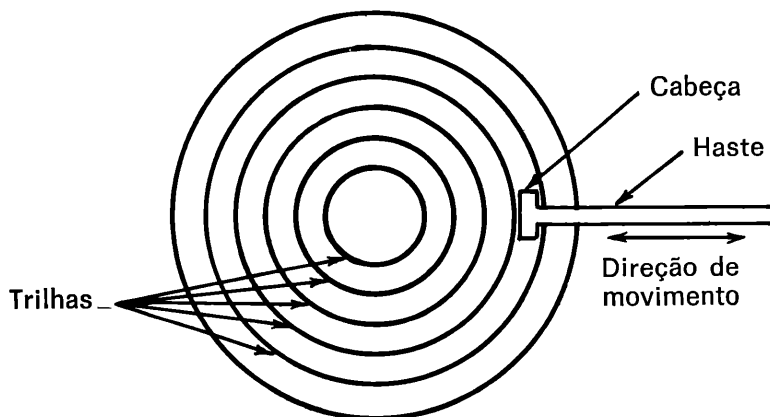


Figura 4.6 — Disco magnético.

A figura abaixo mostra, de perfil, a estrutura de um “disc-pack” e o pente com as hastes que seguram as cabeças de leitura e gravação.

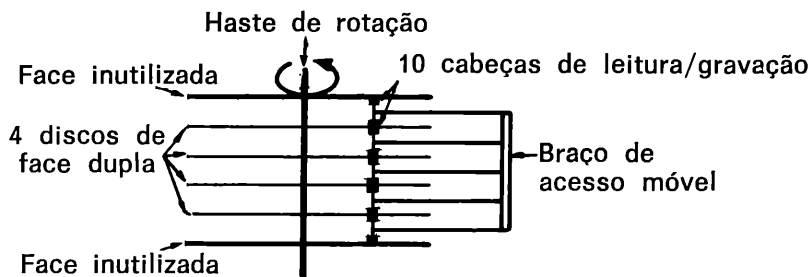


Figura 4.7 — Estrutura de um disc-pack de 10 discos

DISPOSIÇÃO DOS DADOS NO DISCO MAGNÉTICO

Já dissemos que a superfície de cada disco está dividida em certo número de trilhas concêntricas. O número de trilhas por superfície varia, mas um modelo típico pode ter 200 trilhas para cada superfície de disco. A trilha é dividida em registros de extensão variável. O número de bytes que podem ser armazenados numa trilha depende do número de registros em que está dividida a capacidade de armazenar. No caso especial em que toda a trilha é considerada como um único registro, ela pode, nos modelos mais comuns, armazenar 3 625 bytes.

O CONCEITO DE CILINDRO

Numa unidade de discos magnéticos que trabalha com “disc-packs”, existem 10 ou 20 cabeças de leitura e gravação, que se acham em hastes presas a uma barra comum, como se vê na figura desenhada acima, e têm um movimento conjunto. Portanto, quando uma das cabeças está posicionada numa trilha, cada uma das restantes está posicionada na trilha correspondente de uma outra superfície de disco.

Em outras palavras, em cada instante existem 10 (ou 20) cabeças de leitura e gravação, posicionadas em 10 (ou 20) trilhas, pertencentes a diferentes superfícies de gravação. É verdade que apenas uma dessas cabeças pode estar ativada de cada vez, de modo que a unidade de discos só tem acesso a uma trilha em cada instante, mas a unidade de discos pode selecionar uma outra trilha deste grupo de 10 ou 20 trilhas sem ter que reposicionar as cabeças, bastando desativar uma cabeça e ativar outra.

O conjunto de trilhas que ficam acessíveis à unidade de discos para dado posicionamento das cabeças recebe o nome de *cilindro ideal* ou, simplesmente, *cilindro*. Esse nome é justificado do ponto de vista geométrico, pois essas trilhas estão situadas sobre a superfície de um mesmo cilindro.

Para maior eficiência de processamento, os dados são sempre armazenados segundo o conceito de cilindro. Por exemplo, se um arquivo de dados ocupa 5 trilhas, ele será armazenado em 5 trilhas pertencentes ao mesmo cilindro (e, portanto, a faces de disco diferentes), e *não* em 5 trilhas consecutivas da mesma face de disco. Desta forma, o acesso ao arquivo exige apenas um movimento de posicionamento das cabeças.

FORMAS DE ORGANIZAÇÃO DE DADOS EM MEMÓRIA DE DISCOS MAGNÉTICOS

Quando se armazena um arquivo de dados em memória de discos magnéticos, esse arquivo pode ser organizado de três maneiras diferentes. Existe a organização seqüencial, a organização seqüencial indexada e a organização aleatória.

Na *organização seqüencial*, os dados estão classificados em determinada ordem, e o arquivo contém somente os registros com os dados.

Na *organização seqüencial indexada*, os dados também estão classificados em determinada ordem; nesse caso, entretanto, o arquivo contém não somente os registros com os dados, mas também registros que formam um índice, a fim de facilitar a rápida localização dos dados.

Na *organização aleatória*, cada registro tem um número de identificação, mas os registros *não* estão classificados de acordo com esse número. Em vez disto, cada registro está armazenado num local da memória, cujo endereço pode ser calculado a partir do número de identificação, por meio de uma fórmula matemática.

MODALIDADES DE PROCESSAMENTO DE ARQUIVOS DE DISCO MAGNÉTICO

Um arquivo de dados armazenado em memória de discos pode ser processado de duas maneiras: de modo seqüencial e de modo aleatório. No processamento seqüencial, são processados todos os registros do arquivo ou de uma parte do arquivo, na ordem em que estão armazenados na memória. No processamento aleatório, são processados apenas alguns registros, que podem estar localizados em qualquer parte do arquivo.

Quando um arquivo tem organização seqüencial, ele só pode ser processado de maneira seqüencial. Quando a organização de um arquivo é seqüencial indexada ou aleatória, esse arquivo pode ser processado de maneira seqüencial ou de maneira aleatória.

Tempos de acesso

Memória de trabalho e memória de massa

Uma característica importante de um sistema de memória é o seu tempo de acesso, isto é, o tempo necessário para localizar na memória a informação desejada e transferir esta informação para o local desejado.

No caso da memória de núcleos magnéticos e de película delgada, nos computadores de construção mais recente, o tempo de acesso é da ordem de um microssegundo (um milionésimo de segundo). No caso das memórias de tambor magnético e de disco magnético, o tempo de acesso é da ordem de algumas dezenas de milissegundos (milésimos de segundo).

Portanto, nas memórias de tambor ou de disco, o acesso à informação é milhares de vezes mais lento do que nas memórias de núcleos e de película delgada.

Devido a estas características, dividimos os sistemas de memória em duas categorias: memória de trabalho e memória de massa. A memória de trabalho deve armazenar o programa (ou os programas)

que o computador está executando em determinado momento, bem como os dados a que o computador deve recorrer com grande frequência. É evidente que a memória de trabalho deve ser de acesso rápido; assim, usa-se para este fim a memória de núcleos magnéticos ou de película delgada.

Esta memória tem, no entanto, uma limitação: sendo de fabricação muito dispendiosa, o seu tamanho é limitado pelo fator custo.

Assim, a memória de trabalho pode armazenar algumas dezenas de milhares ou centenas de milhares de caracteres, chegando em alguns casos excepcionais a um milhão de caracteres.

Já a memória de massa ou memória de grande capacidade, representada pelos discos e tambores, pode armazenar milhões ou dezenas de milhões de caracteres; em compensação, neste caso a velocidade de acesso é muito menor. Por isto, usa-se a memória de massa para armazenar programas com os quais não se trabalha no momento, ou dados a que o computador recorre com menos frequência.

Memória de acesso aleatório e de acesso não-aleatório

O termo "aleatório" significa ao acaso, sem nenhuma ordem predeterminada.

Um sistema de memória é de acesso aleatório quando todas as suas posições de memória têm o mesmo tempo de acesso, ou seja, quando o tempo de acesso de dada posição de memória não depende de sua localização.

Os sistemas que não têm latência (núcleos magnéticos, película delgada) são sistemas de acesso aleatório. Os sistemas em que existe latência (disco, tambor) não são, a rigor, de acesso aleatório, mas podem ser considerados de acesso quase aleatório.

Com efeito, embora o tempo de acesso das várias posições de memória não seja o mesmo, ele pode ser considerado da mesma ordem de grandeza.

Por outro lado, se considerarmos um rolo de fita magnética como um sistema de memória externa, esse sistema será de acesso não-aleatório.

Com efeito, o tempo gasto para ler e analisar as informações de um rolo de fita é da ordem de 5 minutos. Portanto, se dada informação estiver localizada logo no início da fita, o seu tempo de acesso poderá ser de alguns milissegundos, e, para as informações gravadas no meio da fita, de alguns minutos.

(Note-se que este fato não contra-indica o uso da fita magnética como sistema de memória externa; ele nos obriga, apenas, a usar um método de trabalho chamado processamento seqüencial, onde os dados são utilizados na ordem em que se encontram gravados na fita.)

ALGUNS CONCEITOS IMPORTANTES RELACIONADOS COM A ARMAZENAGEM DE INFORMAÇÕES NOS COMPUTADORES ELETRÔNICOS

Um dos aspectos mais importantes do processamento de dados é a armazenagem de informações e o acesso às informações armazenadas.

A seguir, examinaremos sucintamente alguns conceitos relacionados com a armazenagem de informações e o acesso a elas.

Unidade de informação

A unidade de informação de um computador é a quantidade de informações a que o processador tem acesso, em determinado instante. Como já vimos, em alguns computadores, esta unidade básica é o dígito de 6 bits, em outros é o byte de 7 bits, e em outros ainda é uma palavra constituída de um número de bits, que varia conforme o modelo de computador.

Esta unidade de informação tem uma propriedade importante: pode ser individualizada por meio de um símbolo que é o seu endereço.

Posição de memória

Qualquer que seja a unidade de informação de dado computador, esta unidade ocupará certo lugar físico em sua memória. Esse lugar físico recebe o nome de *posição de memória* ou *local de memória*.

Podemos, pois, apresentar a seguinte definição:

"Uma posição de memória ou local de memória é o lugar físico ocupado na memória de um computador pela unidade de informação desse computador."

Exemplificando, num computador dotado de memória de núcleos magnéticos, no qual a unidade de informação seja um byte de 8 bits, uma posição de memória será constituída de 8 núcleos magnéticos, já que cada núcleo magnético armazena um bit.

Em certos casos, há necessidade de se fazer uma representação gráfica da memória de um computador. Esta representação gráfica recebe o nome de *mapa de memória*. Num mapa de memória, cada posição de memória é geralmente representada por um quadro ou por um retângulo.

O desenho abaixo dá idéia de um mapa de memória:

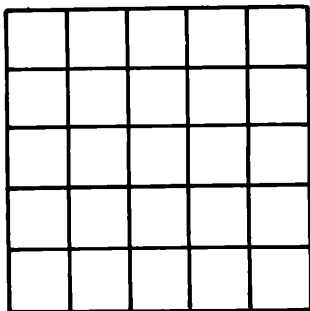


Figura 4.8.

Endereço — Endereçamento

O desenho feito não pode ser considerado um mapa de memória completo. Embora tenha representado por um quadrado cada uma das posições de memória, ele não individualizou, isto é, não forneceu um meio de designar especificamente determinada posição de memória.

Esta individualização é de importância fundamental nos computadores eletrônicos, pois de nada nos adianta estar uma informação armazenada na memória se não tivermos um meio de identificá-la para a buscarmos quando necessário.

Num computador eletrônico digital, a cada posição de memória está associado, de maneira permanente, um símbolo que a designa especificamente, e que tem o nome de *endereço* da posição de memória. Este fato dá origem à seguinte definição:

“O endereço de uma posição de memória é um símbolo permanentemente associado a esta posição de memória que a individualiza em meio às demais.”

Dizer que a associação é permanente significa dizer que ela vem da fábrica e não pode ser alterada pelo programador. Em outras palavras, o endereço de uma posição de memória é uma característica daquela posição de memória.

Se considerarmos, para exemplificar, a memória de um computador hipotético, dotado de 25 posições de memória, existem várias maneiras de individualizar estas posições. A mais simples consistiria em numerá-las de 1 a 25, ou, antes, de 0 a 24. (Em processamento eletrônico de dados, a numeração começa geralmente por zero, e não por 1.)

Neste caso, o mapa de memória, com endereços, teria o seguinte aspecto:

00	01	02	03	04
05	06	07	08	09
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

Figura 4.9.

Em vez dos números de 0 a 24, também poderíamos ter escolhido as letras de A até X, ou qualquer outro conjunto ordenado de símbolos para constituírem os endereços das posições de memória.

O ato de designar uma posição de memória pelo seu endereço recebe o nome de *endereçamento*.

Conteúdo de uma posição de memória

O conteúdo de uma posição de memória é a informação armazenada nessa posição de memória. Esta informação *não* é permanente e imutável. Pelo contrário, ela geralmente muda a cada instante, à medida que o processamento vai-se desenvolvendo. Imaginemos, por exemplo, um conjunto de posições de memória consecutivas, numeradas de 55 a 60; suponhamos que, em dado momento, elas contenham o número 172 350, e que este número foi somado com 111 222, passando assim a ser 283 572.

Isto pode ser representado graficamente da seguinte maneira:

	55	56	57	58	59	60
CONTEÚDO INICIAL	1	7	2	3	5	0
	55	56	57	58	59	60
CONTEÚDO FINAL	2	8	3	5	7	2

Figura 4.10.

Note-se que o conteúdo das várias posições de memória foi modificado, mas o endereço de cada uma permaneceu inalterado.

Se quisermos fazer referência escrita ao conteúdo de uma posição de memória, basta escrevermos seu endereço entre parênteses. Assim, por exemplo:

2851 significa: a posição de memória de endereço 2851.
(2851) significa: o conteúdo da posição de memória de endereço 2851.

Áreas de memória

Em geral, um programa não processa posições de memória isoladas, e sim um certo número de posições de memória consecutivas, que recebe o nome de *área de memória* ou *campo*. Em muitos computadores, a área de memória também recebe o nome de operando, isto é, a área de memória cujo conteúdo vai ser submetido a uma operação.

Símbolos usados para a formação dos endereços

O método mais usual de endereçamento consiste em numerar as posições de memória, consecutivamente, desde 0 até N-1, sendo N o número de posições de memória do computador. Então, por exemplo, os endereços das posições de uma memória de 5 000 posições seriam números desde 0 000 até 4 999.

O fato de iniciarmos a contagem por zero, e não por um, tem um motivo prático. Imagine-se um computador com 10 000 posições de memória. Numerando-as de 0 000 a 9 999, todos os endereços podem ser escritos com quatro algarismos, ao passo que, numerando-as 0 001 a 10 000, existiria um endereço "excepcional" exigindo 5 algarismos, e, por causa deste endereço único, teríamos que reservar sempre uma casa a mais, nos lugares destinados a escrever endereços.

Quando se usa o processo de numeração consecutiva, o endereço de uma posição de memória é um número, mas não necessariamente um número do sistema decimal de numeração. Demonstra-se que há um melhor aproveitamento da memória quando o sistema interno de endereçamento é feito num sistema de numeração, cuja base seja uma potência de 2.

Por esta razão, na maioria dos computadores atualmente em uso o sistema interno de endereçamento utiliza a base 16 (sistema de numeração hexadecimal). À primeira vista, pode parecer que esta modalidade de endereçamento dificulta bastante o trabalho do programador, mas na realidade isto não ocorre, porquanto, geralmente, o

programa não é escrito em linguagem de máquina, e sim em linguagem simbólica. Nessa linguagem, ele poderá escrever os endereços como se fossem números decimais, e um programa especial de conversão transformará cada número decimal no seu equivalente hexadecimal.

Existem computadores em que os endereços não são números, mas combinação de algarismos, letras e símbolos especiais.

Tipos de endereços

Existem três tipos de endereços, a saber: endereços absolutos, endereços simbólicos e endereços relativos.

O *endereço absoluto* é um endereço que designa uma posição de memória bem determinada. Um endereço absoluto é escrito com os símbolos usados pelo computador em questão, para a formação de endereços. Por exemplo, se os endereços de um computador forem números decimais e se esse computador tiver 8 192 posições de memória (um valor bastante freqüente), os endereços absolutos serão números entre 0000 e 8191.

O endereço absoluto é o que é diretamente “compreendido” pela máquina, sem necessidade de um processo de conversão.

Um *endereço simbólico* é um nome inventado pelo programador que designa uma posição de memória que não está determinada no momento em que se escreve o programa. O uso de endereços simbólicos pressupõe a existência de uma linguagem simbólica de programação e de um programa de conversão (elaborado pelo fabricante do computador), e se encarrega de converter o endereço simbólico em absoluto, isto é, de escolher determinada posição da memória e associar o endereço absoluto desta posição ao endereço simbólico proposto pelo programador.

Vamos dar um exemplo. Em dado programa, há necessidade de uma posição de memória para armazenar um dígito.

O programador escolheu para esta posição de memória o nome arbitrário de K2 (que não representa nenhuma posição de memória específica). O programa de conversão “resolve” que ao nome simbólico K2 deverá corresponder o endereço absoluto 2715. A partir deste momento, K2 passa a designar o endereço absoluto 2715, ou seja, passa a designar uma posição de memória bem definida.

Então, todas as vezes que, no programa em linguagem simbólica, aparecer o endereço simbólico K2, o programa de conversão irá substituí-lo pelo endereço absoluto 2715.

O conceito de endereço simbólico é extremamente importante, pois nos modernos computadores eletrônicos quase toda a programação é feita em linguagem simbólica.

Em inglês, um endereço simbólico é chamado de TAG ou LABEL, o que pode ser traduzido por *rótulo* ou *etiqueta*. Muitas vezes não se traduz o termo, usando a própria palavra LABEL.

ENDEREÇOS RELATIVOS

Um endereço simbólico é, como vimos, o endereço de uma posição de memória que não está definida no momento em que se escreve o programa, ficando definida após um processo em que determinado programa de conversão “escolhe” um endereço absoluto que ele irá associar ao endereço simbólico.

Ora, há casos em que nós queremos designar uma posição de memória que fique “tantas posições antes” ou “tantas posições depois” de uma posição de memória designada por um label. Para isto, usamos os *endereços relativos*, que são endereços da forma LABEL + N.

Trata-se de endereços que são somas algébricas de dois termos. O primeiro deles é, geralmente, um endereço simbólico, e constitui uma origem a ser determinada, e o segundo constitui um deslocamento.

Seja, por exemplo, o endereço simbólico XYZ. Se quisermos referir-nos à posição de memória que fica imediatamente à direita da posição designada pelo label XYZ, escreveremos XYZ + 1.

Note-se que, enquanto a localização absoluta da posição XYZ estiver indefinida, também não estará definida a de XYZ + 1. Mas, no momento em que a XYZ estiver associado um endereço absoluto, estará também definida a localização de XYZ + 1: será a posição de memória cujo endereço é uma unidade a mais do que o endereço absoluto associado ao label XYZ. Se, por exemplo, a XYZ foi associado o endereço absoluto 1350, o endereço relativo XYZ + 1 designará a posição de memória 1351.

Ciclo de memória

O ciclo de memória, também chamado ciclo de máquina, é um valor característico para cada computador, que nos dá uma idéia da velocidade de operação deste computador.

Se estudarmos os tempos gastos na execução das várias instruções de um computador, constataremos que todos esses tempos são múltiplos de determinado valor; este é o valor denominado ciclo de memória.

Portanto, podemos defini-lo como sendo um intervalo de tempo fundamental, do qual todos os tempos internos de operação de um computador são múltiplos inteiros.

O valor exato do ciclo de memória varia conforme o modelo de computador; em geral, num computador de segunda geração, este valor é próximo de 0,000 005 segundo (5 microssegundos), enquanto nos sistemas de terceira geração ele é próximo de 0,000 001 segundo (1 microssegundo), chegando mesmo a ser inferior a um microssegundo.

Tempo de acesso

O tempo de acesso da memória pode ser definido da seguinte maneira:

“Chama-se tempo de acesso de um sistema de memória o tempo necessário para localizar nesse sistema uma posição de memória e transferir uma informação de ou para esta posição de memória.”

Esta definição mostra que o tempo de acesso pode ser considerado como a soma de duas parcelas: o tempo de localização da posição de memória e o tempo de transferência da informação.

$$T_{ac} = T_{loc} + T_{tr}$$

O tempo de localização (latência) existe somente nos sistemas em que a localização da informação depende de movimento mecânico (memória de tambor e disco). Nos demais sistemas de memória, esta parcela é nula, e o tempo de acesso é somente o tempo de transferência da unidade de informação de ou para uma posição de memória.

Em sistemas de memória que não têm latência, como a memória de núcleos magnéticos e os sistemas de película delgada, o tempo de acesso está relacionado com o ciclo de memória, embora não se confunda com ele. Nestes sistemas, o tempo de acesso vale tantos ciclos de memória quantos forem necessários para a transferência da unidade de informação.

Nas memórias de tambor e disco, o tempo de acesso não depende do ciclo da memória principal do computador, e sim das características mecânicas do tambor ou disco. Neste caso, geralmente não existe um valor fixo para o tempo de acesso, porquanto este varia conforme a amplitude do movimento mecânico necessário para localizar dada informação. Então, para caracterizar quantitativamente o tempo de acesso do sistema de memória, cita-se um valor mínimo, que representará o caso mais favorável, e um valor máximo, que representará o caso mais desfavorável. A experiência mostra que, para um

grande número de consultas à memória (supondo que a localização das sucessivas posições procuradas seja completamente ao acaso), é obtido um tempo médio de acesso, próximo da metade do tempo máximo.

Em geral, o tempo médio de acesso de um sistema de tambor e de disco varia desde algumas dezenas até algumas centenas de milissegundos.

Memória permanente e volátil

Dizemos que um sistema de memorização de dados é permanente quando as informações armazenadas se mantêm por um tempo indefinido, independentemente de fatores externos, sendo destruídas somente quando o próprio sistema efetua a sua desgravação, a fim de armazenar no mesmo local uma nova informação.

Pertencem a esta categoria de sistema de armazenagem a fita magnética e o disco magnético. (Note-se que, mesmo nestes casos, a permanência "indefinida" da informação é relativa; ao fim de alguns meses ou anos, a gravação magnética tende a enfraquecer e exige uma regravação.)

Um sistema de memorização volátil é aquele em que, por influência de fatores externos, a informação armazenada pode perder-se de um momento para outro. Pertence a este grupo a memória de núcleos magnéticos cujo conteúdo está sujeito a destruição se ocorrer uma interrupção no fornecimento de corrente elétrica.

Sendo a memória de núcleos magnéticos o sistema de uso mais difundido, existem métodos de proteção destinados a impedir a perda das informações, devido à interrupção de corrente. Trata-se de geradores que entram em ação no momento em que se verifica uma queda de voltagem da linha, acoplados a sistemas mecânicos que impedem uma variação brusca da voltagem no computador, durante o período de transição, que dura alguns segundos.

Memória primária e secundária

Os sistemas de memória podem ser classificados, conforme a maneira de utilização, em dois grupos: sistemas primários e sistemas secundários.

A memória primária, também chamada memória principal ou memória de trabalho, é o sistema de memória ao qual a Unidade Central de Processamento tem acesso direto e instantâneo.

Em outras palavras, a U.C.P. pode, a qualquer momento, chamar qualquer informação primária, dando o seu endereço, e obterá a informação desejada instantaneamente.

A memória secundária é aquela a que a U.C.P. tem acesso indireto, ou seja, através de um canal seletor ou canal multiplex. A memória secundária complementa a memória principal, sendo geralmente constituída de sistemas de memória de massa (discos magnéticos, tambóres magnéticos ou cartões magnéticos).

Memória virtual

A memória virtual é um conceito recente, adotado por alguns tipos de equipamentos. Trata-se de memória de tambor magnético, de acesso ultra-rápido, que constitui realmente um prolongamento da memória principal (e não apenas um complemento desta), graças a um sistema especial de endereçamento, que utiliza o conceito de "páginas" de memória.

Exemplificando, um equipamento pode apresentar uma memória real de mais de 2 milhões de bytes (núcleos magnéticos), complementada por uma memória virtual (em tambor magnético) de mais de 8 milhões de bytes.

A memória virtual pode ser armazenada em disco magnético. No entanto, a constante troca de páginas traz uma degradação no desempenho da máquina. A vantagem da memória virtual é que os programas podem ter um tamanho quase que ilimitado, e a multiprogramação pode ser executada independentemente da capacidade da memória principal.

5

SISTEMAS DE NUMERAÇÃO

Quando o homem, há milhares de anos atrás, descobriu as noções de quantidade e de número, suas conversas e transações envolviam apenas números pequenos: uma ovelha, duas árvores, cinco pães etc. Para cada um desses números foi inventado um nome e, posteriormente, quando apareceu a numeração escrita, um símbolo.

Com o desenvolvimento da civilização e o aumento das transações comerciais, apareceu a necessidade de lidar com números maiores e, como não era possível inventar um nome particular para cada número, foram elaborados sistemas que reaproveitam alguns nomes básicos para formar qualquer número desejado.

Surgiram, assim, os sistemas de numeração. No caso da numeração escrita, este reaproveitamento é de uma eficiência fabulosa: basta dizer que, com os nossos dez algarismos, podemos escrever qualquer número, por maior que seja, isto é, infinitos números, no sentido exato da palavra.

BASE DE UM SISTEMA DE NUMERAÇÃO

Em um sistema de numeração, os números agrupam-se em ordens, sendo que certa quantidade de unidades de dada ordem forma uma unidade da ordem imediatamente superior.

Exemplificando, no sistema decimal, que usamos em nossa vida diária, temos a ordem das unidades simples, a ordem das dezenas, a ordem das centenas, a ordem das unidades de milhar, a ordem das dezenas de milhar e assim por diante. Neste caso, 10 unidades de uma ordem formam uma unidade da ordem imediatamente superior.

Em dado sistema de numeração, o número de unidades de uma ordem, que forma uma unidade da ordem imediatamente superior,

constitui a BASE desse sistema de numeração. Historicamente, era bastante usada como base o número dez, que se adota até hoje, em nossa vida diária. Entretanto, houve civilizações que adotaram sistemas de numeração com bases diferentes de dez, e dos quais existem vestígios até hoje. Por exemplo, a língua francesa chama o número 80 de “quatre vingts”, o que é vestígio de um sistema de numeração de base vinte; os babilônios adotavam um sistema de numeração de base 60, cujo uso se conservou até hoje nas medidas de ângulos e de tempo.

Qualquer número inteiro maior que 1 pode ser usado como base de um sistema de numeração. No trabalho com computadores eletrônicos são muito usados os sistemas de base 2 e de base 16, dos quais falaremos mais adiante.

PRINCÍPIOS BÁSICOS DE UM SISTEMA DE NUMERAÇÃO

Daremos abaixo os dois princípios básicos que regem o nosso sistema de numeração decimal, e qualquer sistema de base b , semelhante ao nosso.

Note-se que o segundo princípio se aplica à numeração escrita e pressupõe a adoção da assim chamada “numeração de posição”. Desta forma, o sistema de numeração dos romanos, embora utilize a base 10 e obedeça ao primeiro princípio, é completamente diferente do nosso quanto à numeração escrita, por não obedecer ao segundo princípio.

Primeiro Princípio

“Num sistema de numeração de base b , são necessárias b unidades de determinada ordem para formar uma unidade da ordem imediatamente superior.”

Segundo Princípio

“Num sistema de numeração de base b , um algarismo escrito imediatamente à esquerda de outro vale b vezes mais do que se estivesse no lugar deste outro.”

O segundo princípio equivale a dizer que cada algarismo dentro de um número tem dois valores, a saber: o seu *valor intrínseco*, que ele tem por ser o algarismo que é, e o seu *valor de posição*, que ele tem por ocupar dentro do número o lugar que ocupa.

Obtém-se o valor de posição multiplicando-se o valor intrínseco pelo *peso* da casa ocupada pelo algarismo, sendo que os pesos das sucessivas casas da direita para a esquerda são as sucessivas potências da base: b^0 , b^1 , b^2 ,, b^n .

Exemplificando, consideremos o número decimal 777. Os três algarismos deste número têm o mesmo valor intrínseco, que é sete.

Entretanto, eles têm valores de posição diferentes: o algarismo da esquerda tem o valor de posição 700 (obtido multiplicando-se o valor intrínseco 7 pelo peso 100); o algarismo do meio tem o valor de posição 70; e o algarismo da direita apresenta o valor de posição igual a 7. Neste último caso, o valor de posição é igual ao valor intrínseco, porque o peso da casa das unidades simples é 1.

POLINÔMIO REFERENTE A UM SISTEMA DE NUMERAÇÃO

Em conseqüência do segundo princípio enunciado atrás, podemos afirmar que um número escrito num sistema de numeração de base b pode ser considerado como uma representação abreviada do seguinte polinômio:

$$N = a_n b^n + a_{n-1} b^{n-1} + \dots + a_2 b^2 + a_1 b^1 + a_0 b^0$$

sendo que, em conseqüência do primeiro princípio, os coeficientes a_0 , a_1 , \dots , a_n , têm que ser menores do que a base b . (Com efeito, se tivéssemos um termo $a_k \cdot b^k$, com um coeficiente a_k maior ou igual à base b , então as a_k unidades de ordem k formariam, pelo menos, uma unidade de ordem $k+1$.)

Isto nos leva a concluir que, em qualquer sistema de numeração, o maior algarismo é o que vale uma unidade a menos do que a base. Em qualquer sistema de numeração, a base deste sistema é representada pelo número 10.

O SISTEMA DE NUMERAÇÃO DE BASE 10

O sistema de numeração de base 10, comumente chamado sistema de numeração decimal, usa os algarismos 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. Nesse sistema, dez unidades de dada ordem constituem uma unidade da ordem imediatamente superior, e qualquer algarismo escrito imediatamente à esquerda de um outro vale dez vezes mais do que se estivesse no lugar deste outro.

O sistema decimal é o sistema de numeração usado em nossas transações diárias e é o sistema em que estamos habituados a executar as operações aritméticas. Por isso, tomaremos sempre esse sistema como referência, mesmo para o estudo dos demais sistemas de numeração.

Quando se escreve um número, sem qualquer indicação da base, pressupomos que este número esteja escrito no sistema de numeração

de base 10. Quando se escreve um número num sistema de numeração cuja base é diferente de 10, convém indicar a base, escrevendo-a à direita do número, meio espaço abaixo:

251_{16} (o número 251, escrito no sistema de base 16).

10110_2 (o número 10110, escrito no sistema de base 2).

A indicação da base é dispensável nos casos em que, por alguma razão, não houver dúvida sobre a base em que foi escrito o número.

POLINÔMIO REFERENTE AO SISTEMA DE NUMERAÇÃO DECIMAL

Um número qualquer, escrito no sistema de numeração de base 10, pode ser considerado como uma representação abreviada do seguinte polinômio:

$$N = a_n 10^n + a_{n-1} 10^{n-1} + \dots + a_1 10^1 + a_0 10^0$$

Pode-se verificar isto praticamente, por meio de um exemplo. Seja, por exemplo, o número 718 205 escrito no sistema de numeração de base 10. É óbvio que este número pode ser decomposto na seguinte soma:

$$718\ 205 = 700\ 000 + 10\ 000 + 8\ 000 + 200 + 0 + 5$$

Esta soma, por sua vez, pode ser escrita sob a forma:

$$7 \times 100\ 000 + 1 \times 10\ 000 + 8 \times 1\ 000 + 2 \times 100 + 0 \times 10 + 5 \times 1,$$

que equivale ao seguinte:

$$7 \times 10^5 + 1 \times 10^4 + 8 \times 10^3 + 2 \times 10^2 + 0 \times 10^1 + 5 \times 10^0,$$

como queríamos demonstrar.

Note-se que nenhum dos algarismos do número (isto é, nenhum dos coeficientes do polinômio) pode ser maior ou igual a 10, pois se fosse ele formaria, pelo menos, uma unidade da ordem imediatamente superior.

O SISTEMA DE NUMERAÇÃO DE BASE 2

O sistema de numeração de base 2, também chamado sistema binário, tem apenas os algarismos 0 e 1, sendo que um número escrito nesse sistema de numeração pode ser considerado como uma representação abreviada do seguinte polinômio:

$$N = a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_1 2^1 + a_0 2^0$$

isto é, os “pesos” das casas, da direita para a esquerda, são as sucessivas potências de 2.

Um número escrito no sistema de numeração de base 2 chama-se número binário.

A título de ilustração, mostraremos como se escrevem, no sistema de numeração de base 2, os números desde 0 até 10 000₂.

Número Binário	Equivalente Decimal
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15
10000	16

Muitos computadores eletrônicos fazem as operações aritméticas no sistema de numeração de base 2, o que apresenta duas importantes vantagens. Em primeiro lugar, a armazenagem de um número escrito no sistema de numeração de base 2 requer um número menor de bits do que a armazenagem do valor correspondente no sistema decimal; além disso, as operações aritméticas binárias são de execução mais rápida do que as operações decimais.

O SISTEMA DE NUMERAÇÃO DE BASE 16

No sistema de numeração de base 16, também chamado sistema hexadecimal, o maior algarismo corresponde ao número que, no sistema decimal, se escreveria como 15. Os dez primeiros algarismos do sistema hexadecimal são os mesmos do sistema decimal; para os seis algarismos restantes temos que inventar um conjunto de símbolos; na prática, usam-se, para este fim, as seis primeiras letras do alfabeto. Portanto, os algarismos hexadecimais são os seguintes:

Algarismo Hexadecimal	Equivalente Decimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Um número escrito no sistema de numeração de base 16 pode ser considerado uma representação abreviada do seguinte polinômio:

$$N = a_n 16^n + a_{n-1} 16^{n-1} + \dots + a_1 16^1 + a_0 16^0$$

Um número escrito no sistema de numeração de base 16 chama-se número hexadecimal.

A título de ilustração, escreveremos os números hexadecimais desde 10_{16} até 20_{16} :

Número Hexadecimal	Equivalente Decimal
10	16
11	17
12	18
13	19
14	20
15	21
16	22
17	23
18	24
19	25
1A	26
1B	27
1C	28
1D	29
1E	30
1F	31
20	32

MUDANÇA DE BASE

O problema da mudança de base pode ser enunciado da seguinte maneira: "Dado um número qualquer, escrito em certa base b_1 , achar o número equivalente a ele, em outra base b_2 ." A seguir, apresentaremos dois métodos gerais de mudança de base: a passagem de uma base b qualquer para a base 10, e a passagem da base 10 para uma base qualquer b . Note-se que, com estes dois processos, podemos também efetuar a passagem de uma base b_1 para uma base b_2 , ambas diferentes de 10. Por exemplo, para passar da base 3 para a base 16, passaríamos, primeiramente, da base 3 para a base 10, e, numa segunda transformação, da base 10 para a base 16.

Passagem de uma base qualquer b para a base 10

O problema desta passagem pode ser assim enunciado: Dado um número, escrito no sistema de numeração de base b , achar o número que lhe corresponde no sistema de numeração de base 10. A resolução deste problema é muito simples: basta escrever o polinômio referente a este número e, a seguir, calcular o valor numérico do referido polinômio.

Vamos ilustrar esta regra com dois exemplos, nos quais passaremos, respectivamente, das bases 2 e 16 para a base 10.

1.º EXEMPLO

Achar o número decimal correspondente ao número 110010, escrito no sistema de numeração de base 2.

O número binário 110010 corresponde ao seguinte polinômio:

$$1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

O valor numérico deste polinômio é:

$$1 \times 32 + 1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 32 + 16 + 2 = 50$$

Portanto, podemos escrever:

$$110010_2 = 50_{10}$$

2.º EXEMPLO

Achar o número decimal equivalente ao número 2AB3, escrito no sistema de numeração de base 16.

O número dado 2AB3 corresponde ao seguinte polinômio:

$$2 \times 16^3 + A \times 16^2 + B \times 16^1 + 3 \times 16^0$$

Lembrando que o algarismo A vale 10 e que o algarismo B vale 11, obtemos para este polinômio o seguinte valor numérico:

$$2 \times 4\,096 + 10 \times 256 + 11 \times 16 + 3 \times 1 = \\ 8\,192 + 2\,560 + 176 + 3 = 10\,931$$

Portanto, podemos escrever:

$$2AB3_{16} = 10\,931_{10}$$

OBSERVAÇÃO

No caso particular da passagem da base 2 para a base 10, em vez de formar o polinômio e calcular o seu valor numérico, podemos usar um método mais simples, que consiste no seguinte: escreve-se o número binário e, por cima de cada algarismo deste, escreve-se o respectivo valor de posição; a seguir, somam-se os valores de posição daquelas casas em que exista o algarismo binário 1.

Seja, por exemplo, converter para a base 10 o número binário 11100101. Aplicando a regra, temos:

128	64	32	16	8	4	2	1
1	1	1	0	0	1	0	1

Portanto, o número decimal procurado será:

$$128 + 64 + 32 + 4 + 1 = 229$$

Passagem da base 10 para uma base qualquer b

O problema desta passagem pode ser assim enunciado: Dado um número decimal qualquer, achar o número que lhe corresponde em certa base b . A resolução desse problema envolve uma sucessão de divisões. Começamos dividindo o número decimal dado pela base b ; se o quociente obtido for maior ou igual a b , tornamos a dividir b , e assim sucessivamente, até obtermos um quociente menor que b .

Neste momento, encerramos a série, e podemos escrever o número procurado, que será formado com o último quociente e com os sucessivos restos, da direita para a esquerda.

Vamos ilustrar esta regra com dois exemplos, nos quais passaremos da base 10 para as bases 2 e 16, respectivamente.

A, e o valor 13 pela letra D. Portanto, o número hexadecimal procurado é A5D, e podemos escrever:

$$2653_{10} = A5D_{16}$$

Sugerimos ao leitor que tire a prova, procurando converter para a base 10 o número hexadecimal A5D.

ARITMÉTICA DE COMPLEMENTOS

O conceito de complemento aritmético

Dado um número inteiro e positivo, num sistema de numeração qualquer, este número poderá ser uma potência exata da base. Por exemplo, no sistema de base 10 os números 1 000 e 100 000 são potências exatas da base 10. Entretanto, se pegarmos um número ao acaso, o mais provável é que ele não seja uma potência exata da base.

Por outro lado, qualquer número inteiro e positivo que não seja uma potência exata da base de seu sistema de numeração estará sempre situado entre duas potências consecutivas dessa base. Por exemplo, no sistema de numeração de base 10, o número 750 está situado entre 100 e 1 000.

Dado um número que não seja potência exata da base de seu sistema de numeração, chama-se **COMPLEMENTO ARITMÉTICO** do número o valor que se obtém subtraindo este número da potência da base imediatamente superior a ele.

Em processamento eletrônico de dados, o conceito de complemento aritmético é importante, porque muitos computadores substituem a subtração pela “adição complementada”, isto é, ao invés de subtraírem do número A o número B, somam ao número A o complemento aritmético do número B.

Complemento aritmético decimal — Regra prática

Dado um número inteiro e positivo, escrito no sistema de numeração de base 10, seu complemento aritmético é o número que se obtém subtraindo o número dado da potência de 10 imediatamente superior a ele. Por exemplo, dado o número 4 300, situado entre 1 000 e 10 000, seu complemento aritmético é o número que obtemos subtraindo 4 300 de 10 000, ou seja, 5 700. Para determinar rapidamente o complemento aritmético de qualquer número decimal, basta

seguir uma regra prática, em que o cálculo se faz em duas etapas, a saber:

- a) Substituir cada algarismo do número dado pelo seu complemento em relação a 9, ou seja, pela diferença entre 9 e o algarismo em questão.
 - b) Somar uma unidade ao número obtido na etapa anterior.
- Dado, por exemplo, o número 4 300, obtemos:
 Subtraindo os algarismos de 9: 5 699
 Somando uma unidade : 5 700

Complemento aritmético binário — Regra prática

O complemento aritmético de um número inteiro e positivo, escrito no sistema de numeração de base 2, é o número que se obtém subtraindo o número dado da potência de 2 imediatamente superior a ele.

Seja, por exemplo, o número binário 10001 (equivalente ao número decimal 17); a potência de 2 imediatamente superior a ele é o número binário 10000, equivalente ao número decimal 32. O complemento aritmético do número dado será o número binário 001111, equivalente ao número decimal 15.

A subtração tem o seguinte aspecto:

$$\begin{array}{r} 10000 \\ 10001 \text{ --} \\ \hline 001111 \end{array}$$

Para o cálculo do complemento binário, podemos utilizar a seguinte regra prática:

- a) Substituir cada algarismo do número dado pela diferença entre 1 e este algarismo; isto equivale, na prática, a escrever 0 onde havia 1, e 1 onde havia 0.
- b) Somar uma unidade ao número binário obtido pela transformação descrita no item anterior.

Aplicando essa regra prática ao número 10001, que nos serviu de exemplo inicial, obteríamos:

- a) Trocando 1 por 0 e vice-versa: 01110
- b) Somando com uma unidade : 01111

sendo que o zero à esquerda pode ser suprimido sem alterar o valor do número.

Aplicando essa mesma regra prática ao número binário 111000 (equivalente ao número decimal 56, cujo complemento binário deve ser o equivalente binário de 8), temos:

Número dado:	1 1 1 0 0 0
Trocando 1 por	
0 e 0 por 1:	0 0 0 1 1 1
Somando uma	
unidade:	0 0 1 0 0 0

Adição complementada

Uma das aplicações práticas do conceito de complemento aritmético é a assim chamada adição complementada, usada por muitos computadores que, em vez de subtrair, somam o complemento aritmético.

Vamos apresentar alguns exemplos deste processo, nos sistemas de numeração de base 10 e de base 2. O processo de adição complementada pode ser usado com números positivos e negativos, mas, em nossos exemplos, vamos limitar-nos ao caso em que todos os números são positivos e o resultado também é positivo, isto é, o minuendo é maior que o subtraendo.

ADIÇÃO COMPLEMENTADA DE NÚMEROS DECIMAIS

Consideremos a subtração: $620 - 200 = 420$. Ora, em vez de subtrair 200 de 620, podemos somar a 620 o complemento aritmético de 200, que é 800. Obtemos, então:

$$\begin{array}{r} 620 \\ 800 + \\ \hline (1)420 \end{array}$$

Desprezando a unidade transportada, obtemos o resultado correto: 420.

Seja agora a subtração: $4\ 600 - 300 = 4\ 300$. Neste caso, se quisermos substituir a subtração pela adição complementada, devemos completar o subtraendo com zeros à esquerda, para que fique com a mesma extensão do minuendo. Portanto, a subtração será: $4\ 600 - 0300$. Em vez de subtrair 0300, somamos o complemento aritmético de 0300, que é 9700. Obtemos, assim:

$$\begin{array}{r} 4\ 600 \\ 9\ 700 + \\ \hline (1)4300 \end{array}$$

Desprezando a unidade transportada, obtemos o resultado correto, 4 300.

ADIÇÃO COMPLEMENTADA DE NÚMEROS BINÁRIOS

No parágrafo em que tratamos da subtração de números binários, havíamos apresentado três exemplos. Aqui, vamos repetir esses mesmos cálculos, utilizando, porém, o método da adição complementada.

Exemplo (a) $1111 - 1000 = ?$

O complemento aritmético de 1000 é o próprio número binário 1000. Somando 1111 com 1000, obtemos:

$$\begin{array}{r} 1111 \\ 1000 + \\ \hline (1)0111 \end{array}$$

Desprezando a unidade transportada, obtemos o resultado correto 0111.

Exemplo (b) $10100 - 11 = ?$

Neste caso, o subtraendo tem menos dígitos que o minuendo, e deve, portanto, ser completado com zeros à esquerda para que passe a ter a mesma extensão do minuendo: assim, a subtração a efetuar é: $10100 - 00011 = ?$

O complemento aritmético do subtraendo é 11101. Somando 10100 com 11101, obtemos:

$$\begin{array}{r} 10100 \\ 11101 + \\ \hline (1)10001 \end{array}$$

Desprezando a unidade transportadora, obtemos o resultado correto, 10001.

Exemplo (c) $1110 - 1011 = ?$

O complemento aritmético do subtraendo é 0101. Somando 1110 com 0101, obtemos:

$$\begin{array}{r} 1110 \\ 0101 + \\ \hline (1)0011 \end{array}$$

Desprezando a unidade transportadora, obtemos o resultado correto 0011.

6

PARTES DE UM COMPUTADOR ELETRÔNICO DIGITAL

Um computador eletrônico é constituído de várias unidades, interligadas entre si por meio de cabos especiais.

Podemos classificar as unidades que constituem o computador da seguinte maneira:

- UNIDADE CENTRAL DE PROCESSAMENTO (U.C.P.)
- CANAIS DE ENTRADA E SAÍDA
- DISPOSITIVOS PERIFÉRICOS

UNIDADE CENTRAL DE PROCESSAMENTO

Esta é a parte básica do computador. Ela contém os circuitos eletrônicos que realizam as operações aritméticas e lógicas, bem como os circuitos de controle que comandam e coordenam o funcionamento de todo o sistema. Além disso, a U.C.P. contém geralmente a memória principal.

CANAIS DE ENTRADA E SAÍDA

Um Canal de Entrada e Saída é um conjunto de cabos de ligação e circuitos eletrônicos que controla o intercâmbio de dados entre a U.C.P. e os dispositivos periféricos.

Existem dois tipos de canais de entrada e saída: canal multiplex e canal seletor.

Um canal multiplex interliga a U.C.P. com os dispositivos de baixa e média velocidades, como impressoras, unidades de cartão perfurado, unidades de fita perfurada. Pode atender a várias unidades ao

mesmo tempo, numa transmissão intermitente: um byte para uma unidade, depois um byte para uma segunda unidade, um byte para uma terceira e assim por diante. Portanto, neste caso, as unidades são atendidas alternadamente.

Um canal seletor é um canal de alta velocidade, que pode transmitir centenas de milhares de bytes *por segundo*. O canal seletor interliga a U.C.P. às unidades periféricas de alta velocidade: unidades de fita magnética, unidades de disco magnético, unidades de tambor magnético.

Pode haver vários dispositivos periféricos ligados ao mesmo canal seletor, mas o canal só pode atender a um destes dispositivos de cada vez. Isto ocorre porque, no canal seletor, a transmissão de dados não é intermitente, e sim contínua: todos os bytes de um registro são transmitidos em seguida, sem interrupção. Esta modalidade de transmissão chama-se “modalidade de rajadas” (em inglês, burst mode).

DISPOSITIVOS PERIFÉRICOS

Qualquer sistema de processamento de dados necessita de unidades de entrada e saída de dados. Estas unidades recebem o nome genérico de dispositivos periféricos.

A seguir, vamos citar alguns tipos de dispositivos de entrada e saída que podem ser utilizados num sistema de processamento de dados.

Dispositivos que são somente de Entrada de Dados

- Leitora de Cartões Perfurados
- Leitora de Fita Perfurada
- Leitora de Caracteres Magnéticos
- Leitora Ótica

Dispositivos que são somente de Saída de Dados

- Impressora
- Perfuradora de Cartões
- Perfuradora de Fita de Papel

Dispositivos que servem tanto para a Entrada como para a Saída

- Unidade de Fita Magnética
- Console do Operador

- Terminal de Telecomunicações
- Terminal de Vídeo
- Unidades de Disco Magnético (*)
- Unidades de Tambor Magnético (*)
- Unidades de Cartões Magnéticos (*)

Observação

A característica de modularidade permite que o usuário forme uma ampla variedade de dado modelo de computador, tanto no que se refere à capacidade da U.C.P. como no que se refere à escolha dos dispositivos periféricos.

A enumeração precedente mostra o grande número de tipos de dispositivos de entrada e saída. Se a isto acrescentarmos o fato de que geralmente existe mais de um modelo para cada tipo de dispositivo, vemos que o número de diferentes configurações de dado modelo de computador é enorme, para não dizer ilimitado.

(*) Estes dispositivos são unidades de memória externa, que podem ser consideradas como dispositivos de entrada e saída.

7

O CONCEITO DE PROGRAMA E O CONCEITO DE INSTRUÇÃO

O computador eletrônico é uma ferramenta de trabalho capaz de executar as tarefas mais complexas de processamento de dados que se possam imaginar; entretanto, para isto é preciso que estas tarefas sejam decompostas em um grande número de operações elementares; outrossim, cada uma destas tarefas elementares deverá ser “explicada” ao computador, numa linguagem que este entenda.

Eis as duas tarefas básicas do programador:

- 1) Decompor uma operação complexa numa seqüência de operações simples.
- 2) Expressar cada uma dessas operações simples sob forma de um comando que o computador possa entender e executar.

Com isto, podemos definir os conceitos fundamentais da programação: o conceito de instrução e o de programa.

Chama-se INSTRUÇÃO um comando referente a uma operação elementar que o computador seja capaz de executar. Cada modelo de computador dispõe de certo conjunto de instruções, que constituem o seu *repertório* de instruções. O número de instruções que constituem o repertório varia desde algumas dezenas até mais de uma centena.

Chama-se PROGRAMA o conjunto das instruções necessárias para a execução de determinada tarefa. O programa é o roteiro que orienta o computador, mostrando-lhe a seqüência das operações necessárias para executar uma tarefa de processamento de dados. Conforme a complexidade desta tarefa, o programa pode ter centenas ou até milhares de instruções. Note-se, entretanto, que nem todas as instruções de um programa são executadas em todos os casos; por outro lado, um mesmo grupo de instruções dentro de um programa poderá

ser executado repetidas vezes. De modo geral, embora se saiba o número total de instruções que constituem determinado programa, é impossível prever o número de instruções que serão efetivamente executadas em determinado caso.

Quando se deseja que o computador execute determinada tarefa, é preciso introduzir na memória o programa correspondente a essa tarefa.

O nome desta operação é “carregamento do programa”. Normalmente, cada instalação tem uma “biblioteca de programas”, isto é, um disco, ou uma fita magnética, que contém todos os programas utilizados naquela instalação. Cada programa é identificado por um nome, e, quando se deseja executar dado programa, basta “chamar” o programa pelo seu nome.

Uma rotina especial encarrega-se, então, de localizar o programa na fita ou no disco e de carregá-lo na memória.

Em alguns computadores de primeira geração, o programa não ficava armazenado na memória. Em vez disso, usava-se um painel de ligações para montar o programa por meio de fios que uniam determinados pontos do mesmo. Este conceito, denominado programação externa, está hoje inteiramente superado.

PARTES DE UMA INSTRUÇÃO — TIPOS DE INSTRUÇÕES

Uma instrução é um conjunto de caracteres, numéricos ou alfanuméricos. O número de caracteres e sua significação variam conforme o modelo de computador. Entretanto, qualquer instrução é constituída basicamente dos mesmos elementos, que são:

- 1) código de operação;
- 2) endereços de operandos;
- 3) especificações adicionais.

Código de operação

O código de operação é um caráter ou um conjunto de caracteres que especifica a natureza da operação a ser executada.

Endereços de operandos

Os endereços de operandos geralmente especificam as áreas da memória onde se encontram os dados com os quais deverá ser executada a operação especificada.

Uma instrução pode ter um, dois ou três endereços, conforme a lógica de funcionamento do computador.

Especificações adicionais

Em certos casos, o código de operação e os endereços dos operandos são suficientes para determinar completamente a operação a ser executada. Entretanto, em outros casos, estas informações devem ser complementadas com especificações adicionais.

Nos sistemas de terceira geração, um esquema de instrução bastante comum é aquele que apresenta, além do código de operação e dos endereços de operandos, a especificação da extensão dos operandos.

INSTRUÇÕES DE UM, DOIS E TRÊS ENDEREÇOS

Existem diversos formatos de instrução, sendo que uma instrução pode conter o endereço de uma área de memória, os endereços de duas áreas de memória ou os endereços de três áreas de memória.

Vamos apresentar sucintamente as características desses três tipos de instrução.

Instrução de três endereços

Uma instrução de três endereços é aquela em que aparecem os endereços de três áreas de memória. Dois desses endereços referem-se às áreas que contêm os valores que entram em determinada operação e o terceiro endereço designa a área na qual deverá ser armazenado o resultado da operação. Por exemplo, desejamos fazer dada operação OP com dois valores armazenados, respectivamente, nas áreas de memória M1 e M2, e desejamos armazenar o resultado dessa operação na área de memória M3.

A instrução teria a seguinte estrutura:

OP	M1	M2	M3
----	----	----	----

Esta é uma instrução de três endereços, pois aparecem nela os endereços M1, M2 e M3. Quando um computador usa instruções deste tipo, diz-se que o computador tem uma *lógica de três endereços*. Essa lógica é pouco usada, porque torna a palavra da instrução muito longa, com o que o programa ocupará muita memória.

Instrução de dois endereços

Numa instrução de dois endereços aparecem os endereços de duas áreas de memória, que são as que contêm os dados que entram em

determinada operação. Quanto ao resultado, ele ficará sempre em uma das duas áreas que continham os valores iniciais.

Portanto, a execução de uma instrução de dois endereços destrói sempre o conteúdo inicial de uma das áreas de memória, cujos endereços aparecem na palavra de instrução. Por exemplo, a instrução que realiza determinada operação OP com os conteúdos das áreas de memória M1 e M2 teria a seguinte estrutura:

OP M1 M2

A localização do resultado varia conforme o modelo de computador. Em alguns modelos, o resultado fica sempre em M1; em outros, o resultado fica sempre em M2; outros ainda permitem ao programador especificar se ele deseja o resultado em M1 ou em M2.

Quando um computador usa instruções de dois endereços, diz-se que ele tem uma lógica de *dois endereços*.

Se quisermos fazer a operação citada anteriormente (fazer a operação OP com os conteúdos das áreas M1 e M2, armazenando o resultado na área M3) num computador de lógica de dois endereços, precisamos dar duas instruções:

OP M1 M2
TR Mn M3

O símbolo TR designa uma operação de transferência, que irá movimentar o conteúdo de Mn para M3. Mn poderá ser M1 ou M2, dependendo das características do computador em questão, isto é, se ele deixou o resultado da operação OP em M1 ou em M2.

Instrução de um endereço

Uma instrução de um endereço contém somente o endereço de uma área de memória. Um computador que utiliza este tipo de instrução tem uma *lógica de um endereço*.

A estrutura de uma instrução de um endereço é a seguinte:

OP M

Na realidade, a maioria das instruções (sejam elas aritméticas, lógicas ou de movimentação de dados) envolve duas áreas de memória. Por isso, os computadores que têm lógica de um endereço utilizam dispositivos auxiliares de memória denominados *registradores*. Dessa maneira, a instrução faz menção à área que contém um dos dois valores, pressupondo que o outro já se encontre num registrador.

Para realizar a operação OP com os conteúdos das áreas de memória M1 e M2, armazenando o resultado na área M3, com um computador que usa lógica de um endereço, teríamos que dar três instruções:

CR	M1
OP	M2
AR	M3

A primeira dessas instruções armazena o conteúdo da área M1 em dado registrador. (Esta operação denomina-se “Carregar o Registrador”).

A segunda instrução efetua a operação OP com o conteúdo do registrador e o conteúdo da área de memória M2. Finalmente, a terceira instrução transfere o resultado (que se encontra no registrador) para a área de memória M3. (Esta operação denomina-se “Armazenar o Registrador”).

Note-se que o uso de registradores não está limitado aos computadores que trabalham com instruções de um endereço.

Na verdade, a maioria deles tem instruções de 2 endereços e usa registradores. Entretanto, no caso de instruções de dois endereços, elas podem usar ou deixar de usar registradores, enquanto uma instrução de um endereço tem que recorrer obrigatoriamente a um registrador.

EXEMPLO

Para exemplificar, apresentaremos o formato de uma das instruções de um computador com lógica de 2 endereços. Trata-se da instrução “Transferência de Caracteres”, cujo código de operação é a configuração de bits 11010010, correspondente aos dígitos hexadecimais D2. Esta instrução tem uma extensão de 6 bytes (48 bits), podendo ser representada com 12 dígitos hexadecimais.

1.º Byte

É o código de operação.

2.º Byte

É a extensão dos operandos. Na realidade, este byte contém um número hexadecimal, que vale uma unidade a menos do que a extensão dos operandos.

3.º e 4.º Bytes

Contêm o endereço do Operando 1, ou seja, o endereço de destino da informação.

5.º e 6.º Bytes

Contêm o endereço do Operando 2, ou seja, o endereço de origem da informação.

Assim sendo, para transferir o conteúdo do campo de endereço hexadecimal 2A $\emptyset\emptyset$ para o campo de endereço hexadecimal 3B $\emptyset\emptyset$, poderemos dar a seguinte instrução:

D 2	\emptyset 4	3 B \emptyset \emptyset	2 A \emptyset \emptyset
CÓDIGO DE OPERAÇÃO	ENDEREÇO DE OP1	ENDEREÇO DE OP2	

Figura 7.1.

Observação

Se aparecer na instrução a especificação \emptyset 4 para a extensão, a máquina tomará como extensão verdadeira dos operandos uma unidade a mais, ou seja, 5 bytes.

CATEGORIAS DE INSTRUÇÕES

As instruções que constituem o repertório de determinado computador podem ser divididas em várias categorias, conforme a finalidade a que se destinam.

Vamos enunciar e descrever sucintamente algumas das categorias encontradas com maior freqüência. É óbvio que se trata apenas de uma enumeração genérica, uma vez que a divisão em categorias e a nomenclatura variam conforme o modelo de computador. Entretanto, em quase todos os computadores, encontramos as seguintes categorias de instruções:

1 — Instrução de entrada e saída

São aquelas que comandam os processos de entrada e saída (leitura de cartões, leitura de fita magnética, impressão etc.) ou funções relacionadas com os processos de entrada e saída (avanço de formulário na impressora, reenrolamento de fita magnética etc.).

2 — *Instruções de movimentação de dados* (Transferência)

São aquelas que provocam a movimentação de dados de uma área de memória para outra, ou de um registrador para outro, ou de memória para registrador, ou vice-versa. A instrução poderá ser de movimentação acompanhada de modificação da informação que está sendo transferida.

3 — *Instruções aritméticas*

São aquelas que comandam as operações aritméticas elementares: adição, subtração, multiplicação e divisão. Os modernos computadores têm geralmente dois conjuntos de instruções aritméticas: as instruções aritméticas binárias, que efetuam as operações no sistema de numeração de base 2, e as instruções aritméticas decimais, que efetuam as operações no sistema de numeração de base 10.

Em muitos computadores de médio e grande portes existe, ainda, um terceiro grupo de instruções aritméticas de vírgula flutuante. A aritmética de vírgula flutuante é usada principalmente em aplicações científicas, e constitui um modo conveniente de representar números muito grandes ou muito pequenos. Consideremos, por exemplo, um número que, em notação comum, é representado pelo algarismo 4, seguido de 32 zeros. Em notação de vírgula flutuante, este mesmo número se escreve: 4×10^{32} . É evidente a economia de memória e o ganho em eficiência que se obtêm com esta notação.

Obs.: A “vírgula flutuante” é conhecida também pelo nome “ponto flutuante”.

4 — *Instruções de comparação*

São aquelas que efetuam a comparação entre os conteúdos de dois campos, ou entre o conteúdo de um campo e uma constante.

5 — *Instruções lógicas*

São certas instruções especiais do repertório de um computador. O adjetivo “lógico”, aplicado ao nome de uma instrução, tem dois significados. Por um lado, existem certas operações de manipulação de bits, que recebem o nome de lógicas: operação lógica “E”, operação lógica “OU” etc. Por outro lado, o nome “lógica”, aplicado a uma instrução aritmética binária, indica que, nesta operação, os dados são tomados *sem* sinal, isto é, em valor absoluto.

6 — *Instruções de controle de seqüência*

São aquelas que controlam a seqüência de execução das instruções.

7 — Instruções de edição

São aquelas que modificam o formato dos dados, conforme a conveniência do processamento. Por exemplo, usam-se instruções de edição para dar aos dados de saída o formato apropriado para serem impressos, suprimindo zeros desnecessários, inserindo símbolos etc. Um caso particular de edição é a *tradução*, que consiste em converter os dados de determinado código para outro. Essa tradução é necessária quando o código de determinada unidade de entrada ou de saída é diferente do código interno de trabalho do computador.

SEQÜÊNCIA DE EXECUÇÃO DAS INSTRUÇÕES

Quase todos os modernos computadores digitais são do tipo seqüencial, isto é, são computadores em que existe uma ordem predeterminada para a execução das instruções. Esta ordem é, geralmente, a ordem em que as instruções estão armazenadas na memória. Isto não quer dizer, entretanto, que o computador seqüencial execute as instruções *necessariamente* nesta ordem. O programador tem meios de alterar a ordem de execução sempre que a lógica do processamento o exigir. Para isto, usam-se as instruções de controle de seqüência, que provocam uma alteração na seqüência das operações. Essa alteração de seqüência recebe várias designações. Em inglês, ela se denomina *BRANCHING* ou *JUMP*. Em português, ela recebe os nomes: *ramificação*, *desvio*, *salto* ou *quebra de seqüência*. Adotaremos o termo *salto*, que pode ser *condicional* ou *incondicional*.

Uma instrução de *salto incondicional* será sempre executada independentemente de qualquer teste ou verificação.

Uma instrução de *salto condicional* poderá ser executada ou não, conforme o resultado de algum teste. A instrução especifica sempre uma condição, e o computador verifica se a condição está satisfeita. Em caso afirmativo, ocorrerá o salto; em caso negativo (isto é, se a condição especificada não estiver satisfeita), o programa não salta, continuando na seqüência normal.

A condição a ser testada para a realização ou não de um salto poderá ser o sinal do resultado de uma operação aritmética, o resultado de uma comparação, a presença ou a ausência de certos bits em determinada posição de memória etc.

8

OS PRINCIPAIS CÓDIGOS UTILIZADOS PELOS COMPUTADORES ELETRÔNICOS

Como vimos, o componente básico das informações armazenadas na memória do computador é o dígito binário, designado pelo nome técnico BIT. Nos computadores de segunda geração, os bits eram agrupados em conjuntos de 6 bits, chamados DÍGITOS; nos computadores de hoje, são, geralmente, agrupados em conjuntos de 8 bits, chamados BYTES.

Existem diversas maneiras de agrupar os bits para formar as informações, e cada uma destas maneiras constitui um código.

Neste parágrafo, analisaremos alguns dos *códigos* utilizados nos computadores eletrônicos digitais.

Antes de apresentarmos as características específicas dos diversos códigos, faremos uma indagação de ordem geral:

Quantos bits são necessários para formar um dígito?

A resposta a esta pergunta dependerá do número de diferentes caracteres que desejamos representar.

Demonstra-se, matematicamente, que, usando n bits, conseguimos formar 2^n combinações diferentes. (A rigor, não se trata de combinações, e sim de arranjos com repetição, de dois elementos, tomados n a n .)

Ora, se quisermos representar os dez algarismos de 0 a 9, precisamos de um número de bits que nos forneça pelo menos dez combinações diferentes. Portanto, se tomarmos $n = 4$, obteremos $2^4 = 16$ combinações diferentes, o que é perfeitamente suficiente, e até demais.

Assim, se quisermos representar somente caracteres numéricos, basta que os dígitos sejam de 4 bits.

Se quisermos representar caracteres numéricos e alfabéticos, precisamos de, pelo menos, 36 combinações diferentes (26 letras e 10 algarismos); 5 bits são insuficientes, pois $2^5 = 32$, mas 6 bits nos dão $2^6 = 64$ combinações diferentes, que permitem representar as letras e os algarismos, sobrando, assim, 28 combinações para os símbolos especiais. Um desses símbolos especiais é sempre o "espaço", isto é, a ausência de informação, que se representa, graficamente, pela notação \emptyset . Portanto, as 64 combinações, que podem ser obtidas com 6 bits, dividem-se da seguinte maneira:

- 10 caracteres numéricos;
- 26 caracteres alfabéticos;
- 28 caracteres especiais (incluindo o espaço).

Como já mencionamos, os sistemas de terceira geração têm, geralmente, como unidade básica de informação o BYTE, constituído de 8 bits. Estes 8 bits permitem formar 256 configurações diferentes, mas apenas algumas destas configurações representam caracteres que podem ser impressos pela impressora. Mais adiante, mostraremos a vantagem do uso do byte como unidade de armazenagem.

O CÓDIGO BINÁRIO PURO E SUAS VARIANTES

Todos os códigos são variantes de um código básico, chamado código binário puro. Esse código binário puro é usado, principalmente, para informações numéricas, e independe do número de bits que constitui a unidade básica de informação. Os demais códigos, usados tanto para informações numéricas como alfabéticas, estão associados a certa unidade básica de informação. A seguir, apresentamos uma relação dos principais códigos atualmente em uso e da extensão da quantidade básica de informações associada a cada código.

<i>Nome do Código</i>	<i>Extensão da Quantidade Básica de Informação</i>
Binário	Variável
Binário-Decimal (BCD)	6 bits
EBCDIC	8 bits
ASCII	8 bits

O código binário puro

O código binário puro nada mais é do que o sistema de numeração de base 2, descrito no capítulo anterior. Portanto, trata-se de um código que usa somente os algarismos 0 e 1, e os “pesos” das posições, da direita para a esquerda, são as sucessivas potências de dois:

..., 128, 64, 32, 16, 8, 4, 2, 1

Nesse código, por exemplo, o número decimal 370 será representado da seguinte maneira: 101110010.

As operações aritméticas, feitas com números que estão em código binário puro, chamam-se operações binárias. O trabalho em aritmética binária é mais encontrado em computadores científicos, mas os modernos computadores comerciais são, geralmente, capazes de executar algumas operações em aritmética binária, embora utilizem predominantemente a aritmética decimal.

A armazenagem de números binários geralmente não se limita a posições de memória de 6 ou 8 bits; o computador reúne duas ou mais posições de memória para formar um campo de maior capacidade de armazenagem. Por exemplo, a reunião de 2 bytes forma um campo de 16 bits, onde podemos armazenar números binários desde 0 até 32 767.

O CÓDIGO BINÁRIO-DECIMAL

O código binário-decimal, geralmente denominado código BCD (do inglês, Binary-Coded Decimal, isto é, Decimal Codificado em Binário), é um código derivado do binário puro, e oferece mais facilidade de manejo do que este último. Como se verá a seguir, a codificação em BCD é muito menos trabalhosa do que a codificação em código binário puro, e a interpretação de números codificados em BCD também é mais fácil.

O princípio do código binário-decimal é muito simples: escreve-se o número sob forma decimal e, a seguir, codifica-se cada um dos algarismos em código binário puro. Por exemplo, a codificação do número 370 em código BCD seria o seguinte:

0011	0111	0000
3	7	0

Neste exemplo, cada algarismo foi representado com 4 bits, porque, como já foi dito, 4 bits são suficientes para a representação de

todos os algarismos. Entretanto, na memória dos computadores que trabalham com o código BCD, cada posição de memória é, geralmente, constituída por 6 bits. Os quatro bits que, por si só, seriam suficientes para representar os algarismos constituem a PARTE NUMÉRICA do dígito; a estes acrescentam-se mais dois bits usados para a representação de caracteres alfabéticos e especiais, que constituem a PARTE DE ZONA (ou simplesmente a ZONA) do dígito.

Cada posição de bit dentro do dígito recebe uma designação própria, o que facilita a referência a determinado bit.

Os quatro bits da parte numérica são designados, da direita para a esquerda, pelos algarismos 1-2-4-8, que constituem os valores de posição destas quatro posições de bit (vide capítulo precedente); os dois bits de zona são designados, respectivamente, pelas letras A e B, também da direita para a esquerda.

Portanto, a estrutura de um dígito pode ser representada da seguinte maneira:

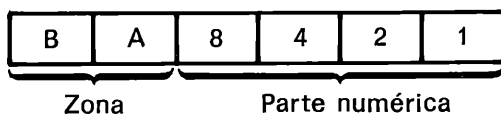


Figura 8.1.

Para qualquer dígito numérico, a parte de zona contém os bits 00. Portanto, a representação do número 370, com todos os bits, é a seguinte:

000011	000111	000000
3	7	0

(Acabamos de dizer que todos os caracteres numéricos têm 00 na parte de zona; entretanto, a recíproca desta afirmação não é verdadeira. Existem caracteres que têm 00 na parte de zona e não são numéricos, como, por exemplo, o caráter “:”, que é representado em código BCD pela configuração 001101.)

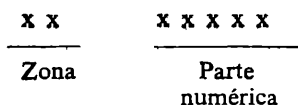
A seguir, apresentamos, a título de ilustração, uma tabela das configurações de bit correspondentes aos algarismos de 0 a 9 e às letras de A até Z, em código BCD.

<i>Caracteres Numéricos</i>		<i>Perfurações</i>
<i>Caráter</i>	<i>Configuração em BCD</i>	<i>no cartão</i>
	B A 8 4 2 1	
0	0 0 0 0 0 0	0
1	0 0 0 0 0 1	1
2	0 0 0 0 1 0	2
3	0 0 0 0 1 1	3
4	0 0 0 1 0 0	4
5	0 0 0 1 0 1	5
6	0 0 0 1 1 0	6
7	0 0 0 1 1 1	7
8	0 0 1 0 0 0	8
9	0 0 1 0 0 1	9
	B A 8 4 2 1	
A	1 1 0 0 0 1	12-1
B	1 1 0 0 1 0	12-2
C	1 1 0 0 1 1	12-3
D	1 1 0 1 0 0	12-4
E	1 1 0 1 0 1	12-5
F	1 1 0 1 1 0	12-6
G	1 1 0 1 1 1	12-7
H	1 1 1 0 0 0	12-8
I	1 1 1 0 0 1	12-9
J	1 0 0 0 0 1	11-1
K	1 0 0 0 1 0	11-2
L	1 0 0 0 1 1	11-3
M	1 0 0 1 0 0	11-4
N	1 0 0 1 0 1	11-5
O	1 0 0 1 1 0	11-6
P	1 0 0 1 1 1	11-7
Q	1 0 1 0 0 0	11-8
R	1 0 1 0 0 1	11-9
S	0 1 0 0 1 0	0-2
T	0 1 0 0 1 1	0-3
U	0 1 0 1 0 0	0-4
V	0 1 0 1 0 1	0-5
W	0 1 0 1 1 0	0-6
X	0 1 0 1 1 1	0-7
Y	0 1 1 0 0 0	0-8
Z	0 1 1 0 0 1	0-9

Por que bytes em vez de dígitos?

A nova unidade "byte" foi introduzida porque oferece uma importante vantagem em relação ao dígito, que é um melhor aproveitamento da memória.

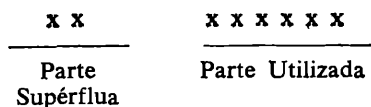
Com efeito, um dígito utiliza 6 bits para armazenar um carácter, que pode ser numérico ou não-numérico:



Ora, no caso de informações numéricas, que constituem, em geral, a maior parte das informações armazenadas na memória, bastam 4 bits para representar o algarismo, sendo desnecessários os 2 bits de zona.

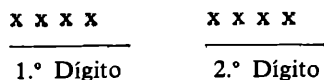
Assim, de 6 bits são utilizados apenas 4, o que corresponde a um índice de aproveitamento de 66%.

Vejamos, agora, o que acontece quando se utiliza o byte como unidade de armazenagem. Se a informação a armazenar é não-numérica, ela exige 6 bits dos 8 que constituem o byte:



Neste caso, de 8 bits são utilizados 6, o que corresponde a um índice de aproveitamento de 75%.

Entretanto, se a informação for numérica, podemos dividir o byte em duas partes, armazenando um dígito numérico em cada parte. Portanto, neste caso, um byte armazena dois algarismos, apresentando um índice de aproveitamento de 100%.



A estrutura de um byte é a seguinte:

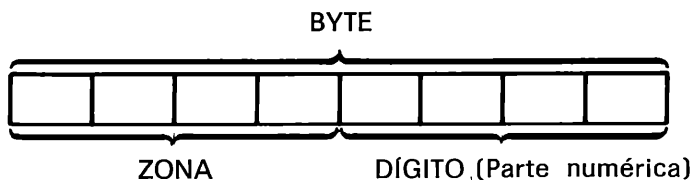


Figura 8.2.

Um byte pode ser utilizado de duas maneiras, que constituem, respectivamente, o **FORMATO DESCOMPACTADO** (Unpacked Format) e o **FORMATO COMPACTADO** (Packed Format).

No formato **DESCOMPACTADO**, o byte armazena **UM CARÁTER ALFANUMÉRICO**:

x x x x x x x x

 8 bits

No formato **COMPACTADO**, cada byte armazena **DOIS CARACTERES NUMÉRICOS**:

x x x x	x x x x
<hr style="width: 50%; margin: auto;"/>	<hr style="width: 50%; margin: auto;"/>
1.º Carácter 4 bits	2.º Carácter 4 bits

A adoção do **BYTE** como unidade de informação é muito vantajosa e permite um melhor aproveitamento da memória, uma vez que as informações numéricas, que constituem, geralmente, a maior parte das informações armazenadas num computador, ocupa somente a metade de uma posição de memória. Na pratica isto equivale a dizer que, para as informações numéricas, a capacidade de armazenar fica duplicada. Por exemplo, 10 bytes permitem armazenar 20 algarismos hexadecimais, enquanto, em 10 posições de memória convencionais, poderíamos armazenar 10 algarismos.

CÓDIGOS DE 8 NÍVEIS DE BIT

Em vista do aparecimento dos computadores de terceira geração, cuja unidade básica de armazenagem de informações (o byte) é constituída de 8 bits, foi necessário criar códigos em que cada carácter fosse representado por 8 bits.

Um desses códigos é uma ampliação do código BCD, e recebeu o nome de **Extended Binary-Coded Decimal Interchange Code** (Código BCD — Ampliado para a Troca de Informação), sendo geralmente conhecido pelas iniciais **EBCDIC**. Um outro código de 8 bits denomina-se **American Standard Code for Information Interchange** (Código Padronizado Americano para a Troca de Informações), sendo geralmente conhecido pelas iniciais **ASCII**.

Note-se que, com 8 bits, podemos formar 256 configurações diferentes, muito mais do que o número de caracteres usuais.

Por esta razão, nos códigos **EBCDIC** e **ASCII**, apenas uma parte de configurações corresponde efetivamente a um carácter de impressão.

Aliás, devido ao grande número de configurações disponíveis, o código prevê letras maiúsculas e minúsculas.

Esta falta de correspondência total entre as configurações de bit e os caracteres acarreta um outro problema, que é o de fazer um “despejo de memória”, isto é, imprimir num formulário o conteúdo de determinado trecho da memória. Esta dificuldade, entretanto, pode ser contornada facilmente, mediante a “representação hexadecimal”, que consiste no seguinte: em vez de procurar representar cada configuração de 8 bits por um caráter, a configuração é desdobrada em dois grupos de 4 bits, cada um dos quais representado por um algarismo hexadecimal. Por exemplo, a configuração 10111000 seria representada pelos algarismos hexadecimais B8.

O CÓDIGO EBCDIC

No código EBCDIC, os bits dentro do byte são numerados de 0 a 7, da esquerda para a direita; os bits de 0 a 3 constituem a zona; e os bits de 4 a 7 constituem a parte numérica do byte. Os caracteres numéricos têm na parte de zona a configuração 1111; nos caracteres alfabéticos, a parte de zona é 1100, 1101 ou 1110.

A seguir, apresentamos uma tabela das configurações que representam os algarismos, as letras e alguns caracteres especiais em código EBCDIC, assim como a representação hexadecimal de cada configuração de bits.

Note-se que, para gerar certo caráter em código EBCDIC, por meio de cartões de 80 colunas, usa-se a mesma combinação de perfurações, que gera esse caráter em código BCD. Exemplificando, a combinação de furos 12-1 gera a letra A em código BCD (110001), quando lida por um sistema que trabalha com 6 níveis de bit, e gera a letra A em código EBCDIC (11000001), quando lida por um sistema que trabalha com 8 níveis de bit.

<i>Algarismos</i>	<i>Configuração em EBCDIC</i>		<i>Representação Hexadecimal</i>
	0123	4567	
0	1111	0000	F0
1	1111	0001	F1
2	1111	0010	F2
3	1111	0011	F3
4	1111	0100	F4
5	1111	0101	F5
6	1111	0110	F6
7	1111	0111	F7
8	1111	1000	F8
9	1111	1001	F9

<i>Letras</i>	<i>Configuração em EBCDIC</i>		<i>Representação Hexadecimal</i>
	0123	4567	
A	1100	0001	C1
B	1100	0010	C2
C	1100	0011	C3
D	1100	0100	C4
E	1100	0101	C5
F	1100	0110	C6
G	1100	0111	C7
H	1100	1000	C8
I	1100	1001	C9
J	1101	0001	D1
K	1101	0010	D2
L	1101	0011	D3
M	1101	0100	D4
N	1101	0101	D5
O	1101	0110	D6
P	1101	0111	D7
Q	1101	1000	D8
R	1101	1001	D9
S	1110	0010	E2
T	1110	0011	E3
U	1110	0100	E4
V	1110	0101	E5
W	1110	0110	E6
X	1110	0111	E7
Y	1110	1000	E8
Z	1110	1001	E9

<i>Caráter Especial</i>	<i>Configuração em EBCDIC</i>		<i>Representação Hexadecimal</i>
	0123	4567	
+	0101	0000	50
-	0110	0000	60
?	1100	0000	C0
!	1101	0000	D0
.	0100	1011	4B
,	0110	1011	6B
:	0101	1110	5E
:	0111	1101	7D
*	0101	1100	5C
\$	0101	1011	5B
/	0110	0001	61
Espaço	0100	0000	40

O CÓDIGO ASCII

No código ASCII, os bits dentro do byte são numerados da seguinte maneira, da esquerda para a direita:

7 6 × 5 4 3 2 1

Portanto, os bits 76×5 formam a parte de zona e os bits 4321 formam a parte numérica do byte.

No código ASCII, os caracteres numéricos têm na parte de zona a configuração 0101, enquanto a parte de zona das configurações alfabéticas é uma das configurações 1010 ou 1011.

A seguir, apresentamos uma tabela das configurações que representam os algarismos e as letras em código ASCII, dando também a representação hexadecimal de cada configuração de bits.

<i>Algarismo</i>	<i>Configuração em ASCII</i>		<i>Representação Hexadecimal</i>
	<i>76x5</i>	<i>4321</i>	
0	0101	0000	50
1	0101	0001	51
2	0101	0010	52
3	0101	0011	53
4	0101	0100	54
5	0101	0101	55
6	0101	0110	56
7	0101	0111	57
8	0101	1000	58
9	0101	1001	59

<i>Letra</i>	<i>Configuração em ASCII</i>		<i>Representação Hexadecimal</i>
	<i>76x5</i>	<i>4321</i>	
A	1010	0001	A1
B	1010	0010	A2
C	1010	0011	A3
D	1010	0100	A4
E	1010	0101	A5
F	1010	0110	A6
G	1010	0111	A7
H	1010	1000	A8
I	1010	1001	A9
J	1010	1010	AA
K	1010	1011	AB
L	1010	1100	AC

<i>Letra</i>	<i>Configuração em ASCII</i>		<i>Representação Hexadecimal</i>
M	1010	1101	AD
N	1010	1110	AE
O	1010	1111	AF
P	1011	0000	BO
Q	1011	0001	B1
R	1011	0010	B2
S	1011	0011	B3
T	1011	0100	B4
U	1011	0101	B5
V	1011	0110	B6
W	1011	0111	B7
X	1011	1000	B8
Y	1011	1001	B9
Z	1011	1010	BA

O CÓDIGO HEXADECIMAL

O código hexadecimal nada mais é do que o sistema de numeração de base 16. Não se trata propriamente de um código interno de armazenagem, mas, antes, de um modo de representar dados binários.

Com efeito, quando um número hexadecimal está armazenado na memória, é impossível distingui-lo do seu equivalente binário, uma vez que a configuração de bits de ambos é a mesma.

No sistema de numeração de base 16, existem 16 algarismos. Para os primeiros dez algarismos do sistema, usamos os mesmos algarismos do sistema decimal, ou seja, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Para os seis algarismos restantes, utilizamos as seis primeiras letras do alfabeto: A, B, C, D, E, F.

A seguir, apresentamos a tabela dos algarismos hexadecimais, com seus equivalentes decimais, e as configurações de bits que os representam, em código binário puro.

<i>Algarismo Hexadecimal</i>	<i>Equivalente Decimal</i>	<i>Configuração de Bits</i>
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101

<i>Algarismo</i> <i>Hexadecimal</i>	<i>Equivalente</i> <i>Decimal</i>	<i>Configuração</i> <i>de Bits</i>
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

O sistema de numeração hexadecimal constitui um modo conveniente para a representação de números binários.

Com efeito, consideremos um número binário qualquer, como por exemplo:

1 0 1 1 0 0 1 1 1 0 1 0 0 1 1 1 (A)

Escrito desta maneira, o número é difícil de ler e, se tivermos que transcrevê-lo manualmente, há uma grande probabilidade de erro.

Vamos separar este conjunto de bits em grupos de quatro bits, a partir da direita. (Se o último grupo da esquerda ficar com menos de quatro bits, completamo-lo com bits zero, à esquerda.)

Obtemos, então:

1 0 1 1 0 0 1 1 1 0 1 0 0 1 1 1 (B)

Basta agora substituir cada configuração de 4 bits pelo seu equivalente hexadecimal, dado na tabela da página precedente, e obtemos um número hexadecimal equivalente ao número binário dado. O número que se obtém é:

B 3 A 7 (C)

Como vemos, a forma (C) é uma representação bem condensada do número binário inicial. Por isso, podemos dizer que o código hexadecimal é um modo abreviado de representação de um número binário.

Existem formalismos matemáticos simples para converter números decimais em hexadecimais, e vice-versa. Na prática, usam-se para este fim tabelas de conversão. A conversão de um número decimal em hexadecimal constitui o meio mais fácil e rápido para codificar esse número em código binário puro.

INDICAÇÃO DE SINAL EM DADOS NUMÉRICOS

Para muitas operações, o computador considera os números armazenados na memória como números algébricos, isto é, números que podem ser positivos ou negativos. Assim sendo, a memória deverá conter alguma indicação de sinal para os valores numéricos armazenados.

A forma da indicação de sinal depende do formato em que estão armazenados os dados. Se os dados estão armazenados em código binário puro, a indicação de sinal é um único bit, geralmente o bit situado mais à esquerda. Normalmente, usa-se o bit 0 para indicar o sinal positivo, e o bit 1 para indicar o sinal negativo.

Nos computadores que trabalham com bytes, os dados numéricos que serão submetidos a operações aritméticas decimais deverão estar num formato especial, chamado formato decimal compactado. Nesse formato, a indicação de sinal é uma configuração de 4 bits, que ocupa a parte da direita (parte numérica) do byte, situado mais à direita (byte menos significativo). No código EBCDIC, a indicação de sinal positivo é a configuração 1100, e a indicação de sinal negativo é a configuração 1101.

A título de ilustração, vamos apresentar as configurações de bit que representam, respectivamente, os números + 12345 e - 12345, em formato decimal compactado.

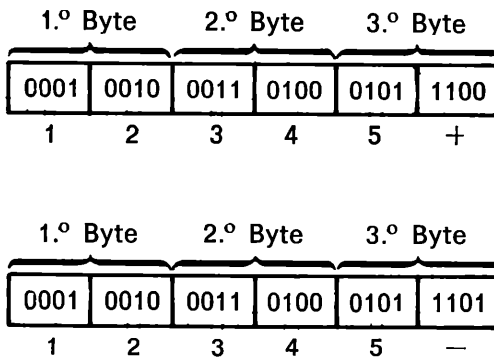


Figura 8.3.

9

A ENTRADA E SAÍDA DE DADOS — VEÍCULOS DE ENTRADA E SAÍDA

Existem modalidades de entrada e saída que dispensam um suporte material para os dados. Isto ocorre, por exemplo, no teleprocessamento, sistema em que os dados são recebidos de ou enviados a um terminal remoto, através de linhas telefônicas, telegráficas, de telex etc.

Normalmente, entretanto, a entrada e saída de dados exige um meio material no qual estejam registradas as informações: esse meio material recebe o nome de portador de dados ou veículo de entrada e saída.

Os principais veículos de entrada e saída são:

- cartão perfurado;
- formulário contínuo;
- fita magnética;
- fita de papel perfurada;
- documento original.

A seguir, examinaremos as principais características desses veículos.

O CARTÃO PERFURADO

Um importante veículo de entrada e saída é o cartão perfurado. Trata-se de um cartão retangular, onde se registram informações sob forma de perfurações retangulares, feitas em posições predeterminadas. É denominado cartão de 80 colunas, devido ao fato de poder armazenar 80 caracteres. Também recebe o nome de cartão Hollerith, em homenagem ao engenheiro Herman Hollerith, a primeira pessoa a usar cartões perfurados para fins estatísticos.

O Código de cartão perfurado

Nos cartões de 80 colunas, usa-se o *Código Hollerith*, baseado em princípios extremamente simples, os quais passamos a expor.

- a) Um carácter numérico é representado por uma única perfuração, na altura correspondente ao algarismo que se quer representar. Assim, uma perfuração na altura 1 representa o algarismo 1, e assim por diante.
- b) Um carácter alfabético é representado por duas perfurações. Para formar um carácter alfabético, usa-se a combinação de uma perfuração 12, 11 ou 0 com uma das perfurações de 1 a 9, da seguinte maneira:

As letras de A até I são representadas pela combinação da perfuração 12 com as perfurações de 1 a 9; a letra A é a combinação 12-1; a letra B, a combinação 12-2; a letra C, a combinação 12-3; e assim por diante.

As letras de J até R são representadas pela combinação da perfuração 11 com as perfurações de 1 a 9; a letra J é a combinação 11-1; a letra K, a combinação 11-2; e assim por diante.

Estas duas séries perfazem 18 letras, faltando, pois, 8 letras para completar as 26 que formam o alfabeto. Assim, as letras de S até Z são formadas pela combinação de perfuração 0 com uma das perfurações de 2 até 9; a letra S é a combinação 0-2; a letra T é a combinação 0-3; e assim por diante, até o Z, que é a combinação 0-9.

- c) Um carácter especial (símbolo matemático ou sinal de pontuação) é representado por uma combinação de duas ou mais perfurações. (Excetua-se o sinal $\&$, que é a perfuração 12, e o sinal —, que é a perfuração 11.) Para os símbolos especiais, não existe uma lei de formação regular como para as letras e os algarismos.

Campos de um cartão

Geralmente, para cada aplicação, as informações de cada espécie são perfuradas nos mesmos lugares, em todos os cartões. Por exemplo, as primeiras seis colunas de cada cartão poderiam ser reservadas para perfurar a data de uma transação. Certo número de colunas consecutivas, destinadas à perfuração de informações de determinada espécie, recebe o nome de *campo*.

Na maioria das aplicações, um cartão tem diversos campos. Por exemplo, num serviço de faturamento, um cartão destinado a registrar um item de um pedido teria um campo para identificar o cliente, outro para a quantidade pedida, o preço unitário e o preço total. Poderia, ainda, ter campos para a descrição do produto, informações

fiscais (alínea, inciso) etc. O modelo que apresentamos atrás é o de um "Cartão Universal", que se presta para qualquer aplicação.

Entretanto, na maioria das instalações, a este clichê fundamental são sobrepostos traços e dizeres que dividem o cartão em campos, e dão a cada campo um título indicativo da informação a ser perfurada nesse campo. É comum a existência, em dada instalação, de dezenas de diferentes tipos de cartão, cada um para determinada finalidade.

Cartões-mestre, cartões-detalle, cartões-resumo

Os cartões perfurados podem ser divididos em três categorias, de acordo com a maneira de utilização: cartões-mestre, cartões-detalle e cartões-resumo.

Um *cartão-mestre* é aquele que contém informações permanentes, ou melhor, informações que variam com pouca frequência, podendo, assim, ser reutilizados muitas vezes, durante um longo período. Por exemplo, um cartão com o nome e o endereço de um cliente é um cartão-mestre.

Um *cartão-detalle* é aquele que contém informações que só valem em determinado momento; portanto, o cartão-detalle é usado uma só vez em determinado tipo de processamento e, a seguir, é jogado fora ou guardado num arquivo morto. Por exemplo, num serviço de emissão de contas de água, o consumo de uma residência durante dado mês será perfurado num cartão-detalle. (Note-se que o cartão-detalle poderá ser processado mais de uma vez, porém, em serviços diferentes: uma vez para emitir a conta, uma vez para fazer uma relação das contas, uma vez para fazer uma estatística de consumo etc.)

Um *cartão-resumo* é aquele que constitui um resumo dos dados de dois ou mais cartões. Por exemplo, numa operação de faturamento, cada item da fatura, geralmente, é perfurado num cartão-detalle. Assim, a emissão de uma fatura com 12 itens exige 12 cartões-detalle. Se quisermos, podemos perfurar para cada fatura um cartão que contenha apenas a identificação do cliente e o valor total da fatura. Este será um cartão-resumo.

Observação

Os conceitos de «mestre», «detalle» e «resumo» (ou «total») são aplicáveis não somente a cartões perfurados, mas a qualquer forma de registro de entrada e saída. Por exemplo, uma fita magnética, onde estão gravados registros com informações-mestre (registros-mestre), chama-se fita-mestre; uma fita magnética, onde estão gravados registros de detalle, chama-se fita de detalle ou fita de movimento.

Máquinas de cartão perfurado

Existem numerosas máquinas destinadas à perfuração e ao processamento preliminar de cartões perfurados. Essas máquinas, em seu conjunto, constituem o que se chama "equipamento convencional", para distingui-lo do "equipamento eletrônico", ou seja, dos computadores. De fato, a maioria dessas máquinas trabalha com princípios eletromecânicos; entretanto, nos últimos anos, têm sido lançadas máquinas perfuradoras e manipuladoras de cartões, dotadas de circuitos eletrônicos e mesmo de memória de núcleos magnéticos.

Antes do advento do computador, em 1951, só existiam máquinas convencionais, que executavam todas as funções de processamento de dados, inclusive o cálculo. Uma instalação de processamento de dados, que não possui computador eletrônico, mas somente máquinas convencionais, chama-se instalação convencional.

Até hoje, muitas organizações fazem o seu serviço apenas com equipamento convencional, quer por não poderem arcar com o custo de um computador, quer porque o seu volume de serviços não justifica a sua instalação.

A seguir, apresentamos a relação de algumas máquinas convencionais de cartão perfurado. Note-se que, numa instalação eletrônica, o computador substitui grande parte dessas máquinas, havendo necessidade de manter somente as 4 primeiras da relação:

- 1) Perfuradora;
- 2) verificadora;
- 3) interpretadora;
- 4) classificadora;
- 5) intercaladora;
- 6) tabuladora;
- 7) calculadora eletromecânica;
- 8) reprodutora.

Desenho de um cartão perfurado

Quando se usam cartões perfurados como veículo de entrada de dados, é preciso estabelecer o formato do cartão, isto é, fixar a localização dos campos em que deverão ser perfuradas as várias informações.

Suponhamos, por exemplo, que um cartão de movimentação de conta bancária deva conter os seguintes dados: data de transação, número da conta, código da transação e valor da transação.

Para cada informação a ser perfurada é preciso designar um campo, escolhendo uma extensão e a localização. Muitas vezes, as infor-

mações a serem perfuradas têm extensão variável; nestes casos, deve-se escolher um campo de tamanho correspondente à maior extensão esperada. Por exemplo, admitindo-se como valor máximo de uma única transação bancária a importância de Cr\$ 99.999.999,99, deverão ser reservadas 10 colunas para o valor da transação (os sinais de pontuação não são perfurados). Portanto, o formato do cartão de movimento de conta corrente poderia ser o seguinte:

<i>Informação</i>	<i>N.º da Coluna</i>	<i>Localização</i>
Data da transação	6	Colunas 1 a 6
Número da conta	8	Colunas 9 a 16
Código da conta	1	Coluna 18
Valor da transação	10	Colunas 21 a 30

O desenho desse cartão seria o seguinte:

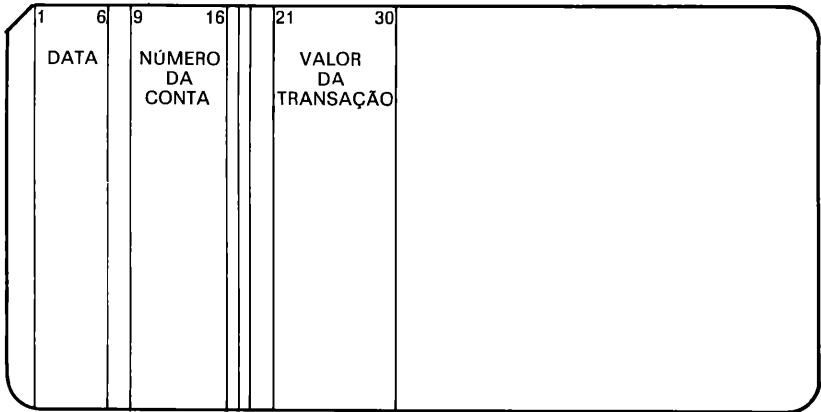


Figura 9.2.

O FORMULÁRIO CONTÍNUO

O mais importante veículo de saída de um computador eletrônico é o formulário contínuo sanfonado, utilizado pelas impressoras do sistema. Trata-se de um meio de saída bastante rápido. Atualmente, a velocidade mínima de impressão é da ordem de 600 linhas por minuto (10 linhas por segundo!), sendo comuns velocidades superiores a 1000 linhas por minuto.

O formulário contínuo sanfonado é uma longa tira de papel, dividida em folhas individuais através de picotes e dobrada em forma de sanfona. Vem acondicionado em caixas de 3000 folhas, em uma via. Existem formulários contínuos em várias vias, com folhas de carbono entre uma via e outra. Neste caso, evidentemente a capacidade de cada caixa é menor.

O formulário contínuo possui uma série de perfurações, nas duas margens, sendo que a impressora possui peças dentadas que produzem o avanço do formulário, através dessas perfurações.

A largura do formulário depende de dois fatores: das características da impressora e da natureza do documento a imprimir.

A densidade de escrita é padronizada, sendo de 10 caracteres por polegada (cerca de 4 caracteres por centímetro). O número de posições de impressão de cada linha pode variar, sendo usuais os seguintes valores: 96, 120, 128, 132 e 160 caracteres. A densidade de linhas, com espaçamento simples, é de 6 linhas por polegada.

Descrição do formato de uma linha

A cada linha a ser impressa corresponde uma área de memória (área de impressão), que contém os dados a serem impressos. O conteúdo dessa área de memória é uma réplica da linha a ser impressa, cabendo ao programa colocar, nos lugares adequados da área de impressão, os dados a serem impressos.

Para facilitar a localização dos dados, dentro da linha de impressão, pode-se usar um formulário especial denominado "Gabarito de Impressão"; nesse formulário, desenha-se uma réplica do documento impresso a ser produzido, e localizam-se os diversos campos que devem constituir cada linha. Este desenho servirá de base ao programador quando escrever a parte do programa que compõe a linha de impressão.

Exemplificando, suponhamos que se queira imprimir uma linha de um relatório de controle de estoque. Nessa linha, queremos imprimir o número da peça, a descrição da peça, o valor unitário, a quantidade existente e o valor total. A especificação do formato dessa linha poderia ter o seguinte aspecto:

<i>Localização das posições de impressão</i>	<i>Número de posições de impressão</i>	<i>Informação contida</i>
1 — 10	10	Número de catálogo
16 — 45	30	Descrição
51 — 58	8	Valor unitário
61 — 67	7	Quantidade existente
71 — 82	12	Valor total

Note-se que esta especificação de formato deverá levar em conta os caracteres de edição. Por exemplo, se a quantidade existente de

determinado artigo tiver no máximo 6 algarismos, deveremos reservar 7 posições de impressão, pois não iremos imprimir um número da forma 123456, e sim um número de forma 123 456 ou 123.456.

Analogamente, as 8 posições reservadas para o valor unitário representam um número de 6 algarismos, impresso no formato:

x.xxx,xx

Note-se, ainda, que um mesmo serviço pode envolver vários formatos diferentes de impressão. Por exemplo, um relatório ou mapa estatístico apresenta, geralmente, 3 tipos de linha:

- Uma ou mais linhas de cabeçalho, dando o nome do relatório, a data da impressão, os títulos das colunas etc.
- Uma série de linhas que constituem o corpo da listagem; essas linhas recebem o nome de linhas-detanha.
- Eventualmente, uma ou mais linhas de total, quando há necessidade de imprimir subtotaís, totais de fim de página, totais gerais etc.

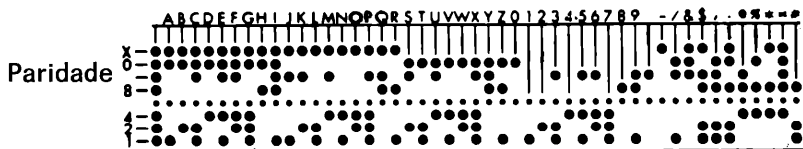
A FITA DE PAPEL PERFURADA

A fita de papel perfurada é uma tira de papel grosso onde se armazenam informações sob forma de perfurações redondas ou retangulares, feitas em posições predeterminadas.

A fita de papel pode apresentar-se em bobinas ou em tiras. Quando enrolada em bobinas, apresenta, em geral, a extensão de 300 metros; as tiras de papel têm extensão variável.

A largura da fita de papel pode ser de uma polegada, 7/8 de polegada ou 11/16 de polegada.

O desenho abaixo mostra um trecho de uma fita de papel perfurada.



Fita de 8 canais

Figura 9.3. — Trecho de uma fita de papel perfurada, mostrando letras, algarismos e símbolos especiais

Como se vê pelo desenho, se considerarmos uma reta transversal à fita, as perfurações podem encontrar-se em várias posições sobre essa transversal. Em outras palavras, há várias posições de perfuração bem determinadas. Na realidade, podemos considerar a fita dividida em uma série de faixas longitudinais; a cada uma corresponde uma posição de perfuração. Cada uma dessas faixas longitudinais recebe o nome de *canal*. O número de canais, ou seja, o número de posições de perfuração, é variável, o que explica a variação na largura da fita. Existe fita de 5, 6, 7 e 8 canais.

No desenho, o leitor notará uma linha de furos de diâmetro menor; trata-se dos furos-guia (sprocket holes), que têm uma dupla finalidade; além de serem usados no processo de avanço mecânico da fita, permitem controlar seu correto posicionamento durante a leitura. O canal dos furos-guia também recebe o nome de canal de arrasto. Note-se que os furos-guia somente existem nos sistemas que trabalham com perfurações redondas.

Se considerarmos uma faixa transversal da fita, de largura correspondente ao diâmetro de um furo, poderemos definir sobre esta faixa certo número de posições de perfuração.

Essa faixa transversal recebe o nome de *frame*. Num frame, haverá tantas posições de perfuração quantos forem os canais da fita. Cada frame da fita de papel armazena um caráter alfanumérico (algarismo, letra ou símbolo especial). A densidade de perfuração, isto é, o número de frames numa unidade de comprimento da fita, é de 10 frames por polegada, o que equivale aproximadamente a 4 frames por centímetro.

O desenho abaixo ilustra os conceitos de canal, frame e posição de perfuração.

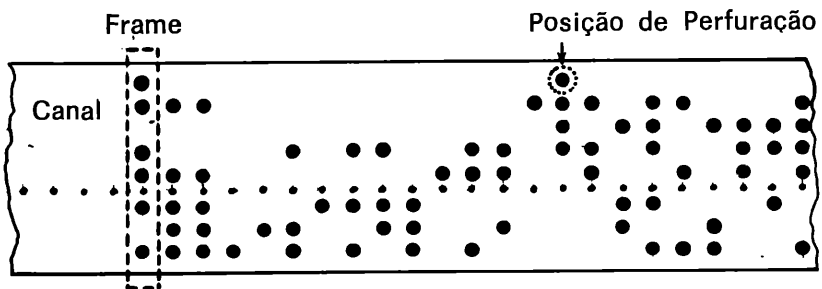


Figura 9.4.

Leitura da fita

Cada posição de perfuração dentro de um frame é interpretada como um bit; quando uma posição de perfuração contiver efetivamente um furo, ela representa um bit 1; as posições de perfuração que não contiverem furos são interpretadas como bit 0. Portanto, em cada frame há uma configuração de presenças e ausências de furos, que representam determinado caráter, no código convencional para dada aplicação.

A leitura pode ser de contato elétrico, ou seja, de escovas. Nas leitoras de fita mais modernas, a leitura é feita fotoeletricamente, o que dá mais segurança e permite maiores velocidades de leitura.

Na leitura de escovas, a presença de um furo permite que uma escova metálica faça contato com uma peça metálica, produzindo um sinal de leitura (bit 1); nos lugares não furados, a escova não faz contato com a peça metálica, registrando-se, então, a ausência de leitura (bit 0).

Na leitura fotoelétrica, a presença de um furo permite que a luz de uma lâmpada incida sobre uma célula fotoelétrica, produzindo uma corrente elétrica (bit 1); quando não existe o furo, a luz não consegue atravessar a fita e a célula fotoelétrica não recebe luz, deixando assim de emitir corrente (bit 0).

Nas leitoras que trabalham com escovas, a velocidade de leitura é da ordem de algumas dezenas ou algumas centenas de frames por segundo; nas leitoras fotoelétricas, as velocidades de leitura são da ordem de 1000 frames por segundo, havendo unidades capazes de ler 2000 frames por segundo.

Canal de paridade

Em alguns tipos de fita de papel, um dos canais é reservado para a função de verificação da leitura, sendo esta verificação feita pelo princípio de paridade. Neste caso, o canal em questão recebe o nome de canal de paridade.

A FITA MAGNÉTICA

A fita magnética, um dos mais importantes veículos de entrada e saída de dados, é uma fita plástica recoberta de uma camada de material impregnado de numerosas partículas de um óxido metálico.

Graças à existência de tais partículas, determinadas regiões da fita podem adquirir uma imantação permanente e, assim, armazenam informações, sob a forma de marcas magnéticas.

Para fins didáticos, admitiremos que um minúsculo trechinho da superfície da fita representa um bit 1, se estiver imantado, e representa um bit 0, se não estiver imantado.

Esta explicação é um modelo conveniente para o estudo da fita magnética, embora, em certos casos, o mecanismo de armazenagem de informações seja mais complexo.

A fita magnética apresenta uma largura de 1/2 polegada (aproximadamente 13 mm) e vem em carretéis com diversos comprimentos: 600 pés, 1200 pés, 2400 pés. O comprimento mais usual é de 2400 pés, ou seja, cerca de 720 metros.

A capacidade de armazenagem de um rolo de fita é da ordem de milhões de caracteres: o valor exato depende do comprimento da fita, bem como da disposição dos dados na fita. Voltaremos a falar sobre isto mais adiante.

Canais e frames

Na fita magnética, as marcas magnéticas devem ficar em posições bem determinadas; uma dessas posições recebe o nome de posição de gravação. As dimensões físicas de uma posição de gravação (isto é, da região da superfície da fita correspondente a uma marca magnética) são diminutas: a altura é da ordem de 1 mm, e a largura é da ordem de 0,1 mm, ou menos.

Se considerarmos uma reta transversal à fita, sobre essa reta existem várias posições de gravação. Na realidade, podemos considerar a fita dividida em uma série de faixas longitudinais, cada uma das quais correspondente a uma posição de gravação. Essas faixas longitudinais em que está dividida a fita recebem o nome de *canais*. Existem fitas magnéticas de 6, 7, 8 e 9 canais. Isto é equivalente a dizer que, ao longo de uma transversal à fita, existem 6, 7, 8 ou 9 posições de gravação.

O conjunto de posições de gravação, situadas sobre a mesma transversal à fita, recebe o nome de frame. Geralmente, nas fitas de 6 e 7 canais, a cada frame corresponde um caráter, mas, nas fitas de 8 e 9 canais, a um frame pode corresponder mais de um caráter.

A largura da fita magnética é sempre a mesma, qualquer que seja o número de canais; o aumento do número de canais é obtido não pelo aumento da largura da fita, mas tornando os canais mais estreitos e reduzindo a distância entre eles. Note-se, ainda, que o mesmo tipo de fita pode ser usado com códigos de 6, 7, 8 ou 9 canais, pois o número de canais não depende da fita, e sim das características da unidade de leitura e gravação.

O desenho abaixo ilustra graficamente os conceitos de posição de gravação, frame e canal.

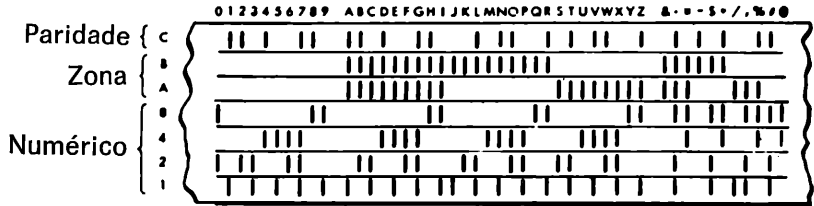


Figura 9.5 — Posições de gravações em fita magnética.

Disposição dos dados na fita magnética

Normalmente, uma fita gravada não se apresenta com uma seqüência ininterrupta de frames. Ao invés disso, a gravação é descontínua, sendo constituída de séries de frames, separadas por intervalos.

Uma série de frames consecutivos recebe o nome de *bloco*, e o intervalo que separa dois blocos consecutivos chama-se *intervalo entre blocos*, ou espaço entre blocos. A existência dos blocos e dos intervalos é uma necessidade.

Com efeito, a fita magnética não pode ser lida de uma só vez, pois nenhum computador tem memória suficiente para armazenar todos os dados contidos numa fita, e nem teria capacidade para processar todos esses dados ao mesmo tempo. Na prática, lê-se certa quantidade de frames, ou seja, um bloco, faz-se o processamento de dados deste bloco e, só então, lê-se o bloco seguinte.

Ora, o processo físico de leitura e gravação da fita magnética exige que a fita esteja em movimento, com velocidade constante; por outro lado, entre os processos de leitura de dois blocos consecutivos, a fita fica parada. Portanto, a cada leitura ou gravação de bloco irá corresponder uma aceleração e uma desaceleração, e é isto que torna necessários os intervalos entre blocos. O comprimento de tais intervalos pode variar de um modelo de computador para outro, mas eles são geralmente da ordem de 2 cm.

O desenho da página seguinte ilustra graficamente o conceito de bloco e de intervalo entre blocos.

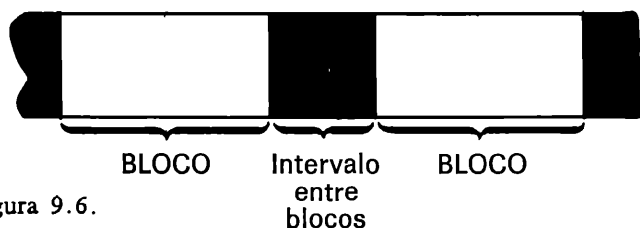


Figura 9.6.

Subdivisão dos blocos em registros

Do ponto de vista da unidade de fita magnética, o bloco é a unidade básica de informação, isto é, um comando de leitura produzirá a leitura de um bloco, e um comando de gravação produzirá a gravação de um bloco. Não se pode ler ou gravar parte de um bloco.

Por outro lado, existem casos em que a extensão de um registro a ser armazenado é menor que a extensão do bloco. Por exemplo, suponhamos que os dados referentes a uma fatura ocupem 120 bytes. Ora, na prática, não é conveniente formar blocos tão pequenos; por isso, agrupamos vários registros no mesmo bloco, subdividindo o bloco em unidades independentes de informação.

Portanto, a unidade básica de informação, do ponto de vista da lógica de processamento, é o *registro*, também chamado *registro lógico*, enquanto o bloco recebe também o nome de *registro físico*.

O desenho abaixo ilustra os conceitos de bloco e registro.

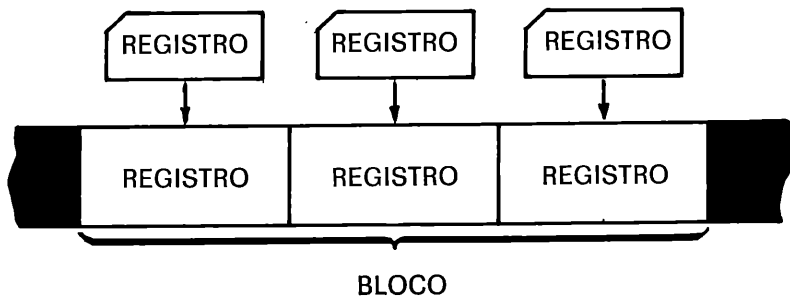


Figura 9.7.

Quando um bloco está subdividido em dois ou mais registros, dizemos que se trata de *registros agrupados em blocos* ou *registros blocados*.

Quando cada bloco se compõe de um único registro, dizemos que se trata de *registros não agrupados em blocos* ou *registros não blocados*.

Densidade de gravação — Velocidade de transferência

Chama-se densidade de gravação o número de frames por unidade de comprimento da fita. As densidades de gravação variam conforme o modelo de unidade de fita, mas as mais usuais são as seguintes:

- 200 frames por polegada;
- 556 frames por polegada;
- 800 frames por polegada;
- 1 600 frames por polegada.

Por exemplo, a densidade de 200 frames por polegada corresponderia a cerca de 8 frames por milímetro, isto é, a largura de um frame, a esta densidade, é de 0,125 milímetros.

A velocidade de transferência é o número de frames que a unidade de fita é capaz de ler ou gravar, na unidade de tempo. Obtém-se esta velocidade multiplicando a densidade de gravação pela velocidade de avanço mecânico da fita. Exemplificando, para uma densidade de gravação de 800 frames por polegada e uma velocidade mecânica de 75 polegadas por segundo, a velocidade de transferência será:

$$v = 800 \times 75 = 60\,000 \text{ frames por segundo}$$

Muitas vezes, na prática, usa-se o nome “quilociclo” (kc) para designar 1000 frames por segundo. Portanto, a fita do exemplo acima teria uma velocidade de transferência de 60 kc.

VELOCIDADE INSTANTÂNEA E VELOCIDADE EFETIVA DE TRANSFERÊNCIA

O termo “velocidade de transferência” pode ser interpretado de duas maneiras, e estas duas interpretações correspondem aos conceitos de “velocidade instantânea de transferência” e “velocidade efetiva de transferência”.

Com efeito, dizer que a velocidade de transferência de determinada unidade de fita magnética é de 60 000 frames por segundo não significa que, num segundo contado no relógio, essa unidade leia ou grave 60 000 frames. Isto só seria verdade se a unidade ficasse lendo ou gravando durante um segundo, sem parar, o que dificilmente acontece.

Na prática, a unidade de fita lê ou grava um bloco de fita, gastando nisto algumas dezenas de milissegundos, e pára, até receber uma nova instrução de leitura ou gravação.

O valor citado acima, de 60 000 frames por segundo, constitui a *velocidade instantânea de transferência*, que pode ser interpretada da seguinte maneira: se uma unidade de fita operou ininterruptamente

durante t milissegundos e, durante esse tempo, leu ou gravou n frames, dividindo-se n por t , obtém-se a velocidade instantânea de transferência. Por exemplo, se a unidade leu 1200 frames em 20 milissegundos, temos:

$$v = \frac{1\ 200\ \text{frames}}{0,020\ \text{segundos}} = 60\ 000\ \text{frames/segundo}$$

Por outro lado, a *velocidade efetiva de transferência* é o número de frames efetivamente lidos ou gravados em um segundo. Por exemplo, se em determinado segundo a unidade de fita leu 30 blocos de 1200 frames, sua velocidade efetiva durante esse segundo foi de 36 000 frames/segundo.

Assinalamento da fita magnética

Um rolo de fita magnética apresenta, geralmente, vários assinalamentos, cujo objetivo é o de facilitar sua identificação e seu processamento. Existem dois tipos de assinalamento: grupos de caracteres gravados na fita ou marcas físicas que se colocam na fita.

Note-se que um rolo de fita, considerado como parte de um arquivo, recebe o nome de "volume". O assinalamento feito por grupos de caracteres gravados na fita serve para identificar o volume, identificar arquivos, assinalar o fim do volume, assinalar o fim do arquivo etc.; o assinalamento mediante marcas físicas indica o início e o fim da parte utilizável de um rolo de fita.

ASSINALAMENTO FEITO POR GRUPOS DE CARACTERES GRAVADOS

O primeiro bloco de cada rolo de fita é, geralmente, um bloco de identificação do volume, que contém o número do volume, a data de gravação, a data de expiração etc.

Cada arquivo gravado numa fita é precedido de um bloco de identificação de arquivo (etiqueta de arquivo). Essas etiquetas podem obedecer a um formato-padrão (etiquetas padronizadas), ou podem não obedecer ao formato-padrão (etiquetas não-padronizadas). O fim dos dados gravados num rolo de fita é assinalado por uma etiqueta de fim de volume.

Existe, ainda, um caráter especial chamado "marca de fita" (tape mark), que é gravado em determinados pontos da fita. A etiqueta de fim de volume é geralmente seguida de duas marcas de fita.

ASSINALAMENTO FEITO POR MARCAS FÍSICAS

O início da parte utilizável da fita é assinalado por um retângulo refletor de alumínio colado na fita, que é detectado por uma fotocélula.

la. Esse retângulo é a marca de início de fita, cujo nome em inglês é *Load Point Mark*.

Quando uma fita é colocada na unidade de fita, o operador aperta a tecla "Load Point", com o que a unidade posiciona a fita automaticamente no "ponto de carga". A unidade de fita reconhece a condição de "Load Point" quando a luz de uma lâmpada incide na marca refletora retangular, e a luz refletida é detectada pela fotocélula.

O término da parte utilizável da fita é assinalado pela marca de fim de fita, que é um retângulo refletor de alumínio, colado na outra extremidade da fita, e perto do outro bordo da fita. Quando a luz de uma lâmpada é refletida por esta marca e incide numa fotocélula, a unidade de fita emite um sinal de fim de fita (End of Tape). O programa deverá, sempre, testar a presença deste sinal e tomar as providências cabíveis, quando ele estiver presente.

Exemplo de formato de registro de fita

Quando as informações são registradas em fita magnética, não existe uma extensão fixa para o registro, que poderá ter centenas ou mesmo milhares de caracteres ou bytes. Neste caso, o programador determina a extensão total do registro, bem como a subdivisão do mesmo, isto é, a localização dos vários campos no registro. Para isto, as posições do registro são numeradas consecutivamente, a partir de 1.

Consideremos, por exemplo, o cadastro de funcionários de uma empresa, gravado em fita magnética, em registros de 200 caracteres. Este registro poderia ser constituído dos seguintes campos:

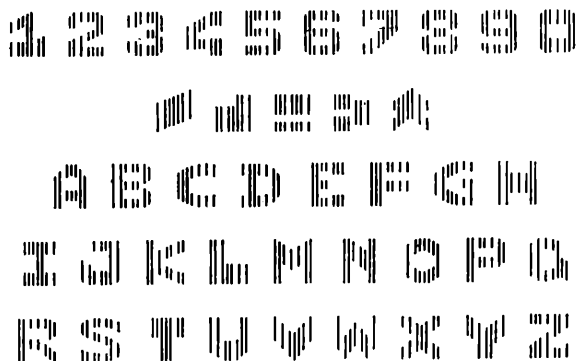
<i>Localização do campo</i>	<i>Extensão do campo</i>	<i>Informação contida</i>
001-006	6	Número do funcionário (chapa)
007-008	2	Departamento
009-010	2	Seção
011-015	5	Código da função
016-040	25	Nome da função
041-080	40	Nome do funcionário
081-120	40	Endereço do funcionário
121-140	20	Local de nascimento
141-146	6	Data de nascimento
147-148	2	Número de dependentes
149-156	8	Número da Carteira de Trabalho
157-164	8	Número do documento de identidade
165-172	8	Número do Título de Eleitor
173-180	8	Número do Certificado Militar
181-190	10	Ordenado fixo
191-200	10	Gratificação de função

Este formato de registro é anotado num formulário, chamado “Gabarito de Formato de Registro”. O mesmo formulário pode ser utilizado para registros de fita e de disco. No caso de fita magnética, entretanto, existe a seguinte particularidade: se os blocos forem de extensão variável, é preciso reservar 4 posições, no início do bloco, para a especificação da extensão do bloco; e, se além disto, os registros forem de extensão variável, é preciso reservar mais 4 posições no início de cada registro, para especificar a extensão do registro.

DOCUMENTO ORIGINAL COMO VEÍCULO DE ENTRADA

Existem técnicas modernas que permitem, em certos casos, a utilização do próprio documento original como veículo de entrada de dados. Uma delas é o uso de caracteres magnéticos, aplicados, principalmente, para o processamento de cheques.

Para este fim, o sistema mais empregado é o CMC-7. Cada caracter é formado por 7 bastonetes separados entre si por espaços curtos e longos. É através da disposição destes espaços (e não pelo formato do caracter em si) que a máquina identifica os dados do documento. Exemplos:



IO

ETAPAS DA ELABORAÇÃO DE UM PROGRAMA

Antes de iniciar qualquer trabalho que tenha características criativas, isto é, que não seja repetição rotineira de uma tarefa já feita anteriormente, convém fazer um planejamento que estabeleça de maneira mais ou menos detalhada a seqüência de operações.

Esse planejamento, por sua vez, abrange várias etapas. Por exemplo, para construir um edifício, elabora-se, primeiramente, o anteprojeto, depois o projeto e, finalmente, as plantas detalhadas.

Antes de começar os trabalhos de filmagem de uma película, é preciso escrever um argumento, depois um roteiro, e assim por diante.

Em particular, a confecção de um programa para a execução de determinado serviço de processamento de dados é precedida de várias etapas de planejamento, que são o levantamento da tarefa, a análise do problema e a resolução lógica do problema. Segue-se o trabalho de programação propriamente dita (codificação) e, finalmente, existe o trabalho de depuração (teste dos programas).

Uma vez obtido o programa depurado, isto é, o programa de onde foram expurgados todos os erros, prepara-se a documentação do programa, que deverá ser arquivada para referência futura. Depois disto, o programa está pronto para ser utilizado, em regime de produção.

Resumindo, a obtenção de um programa livre de erros, pronto para ser introduzido no computador e executado, requer as seguintes etapas de preparação:

- Levantamento da tarefa;
- análise do problema;
- resolução lógica do problema;
- programação propriamente dita;
- depuração;
- documentação.

Vamos descrever sucintamente cada uma dessas etapas.

LEVANTAMENTO DA TAREFA

O levantamento da tarefa é a descrição exata, qualitativamente e quantitativamente, da tarefa a executar. A descrição quantitativa indica os volumes de trabalho, bem como a periodicidade, prazos etc.

O resultado dessa etapa do planejamento é o **ENUNCIADO DO PROBLEMA**.

Para exemplificar, o enunciado do problema referente a um serviço de folha de pagamento poderia ser o seguinte:

“Fazer a folha de pagamento de uma indústria que tem 1 600 empregados, sendo 1 450 horistas e 150 mensalistas. A empresa paga seus empregados quinzenalmente, nos dias 5 e 20 de cada mês. É preciso imprimir a folha, os cheques e as relações referentes aos descontos efetuados.”

ANÁLISE DO PROBLEMA

A análise do problema consiste na descrição do formato dos dados de entrada, do processamento a realizar e do formato de saída desejado. Note-se que a descrição do processamento é extremamente sucinta. Nesta fase do estudo, a descrição do processamento diz apenas O QUE deve ser feito, sem mostrar COMO isto deve ser feito.

O resultado desta etapa do planejamento é um gráfico denominado **FLUXOGRAMA DE SISTEMA**.

O fluxograma de sistema, também chamado *rotina de serviço* ou simplesmente *fluxograma*, procura dar uma idéia geral do processamento, mostrando quais os dados iniciais disponíveis, quais os resultados que se pretende obter com o processamento e qual o caminho geral a ser seguido.

O fluxograma de sistema define, em linhas gerais, o fluxo do serviço dentro da organização, mostrando onde os dados se originam, como são registrados, que processamento deve ser realizado e se este processamento deve ser feito através de um programa de computador ou por meio de algum dispositivo “off-line”, como, por exemplo, uma classificadora mecânica.

Para fazer um fluxograma de sistema, utiliza-se um gabarito (ver figura da página seguinte). São apresentados, em seguida, os símbolos de uso mais freqüente, juntamente com seus significados.

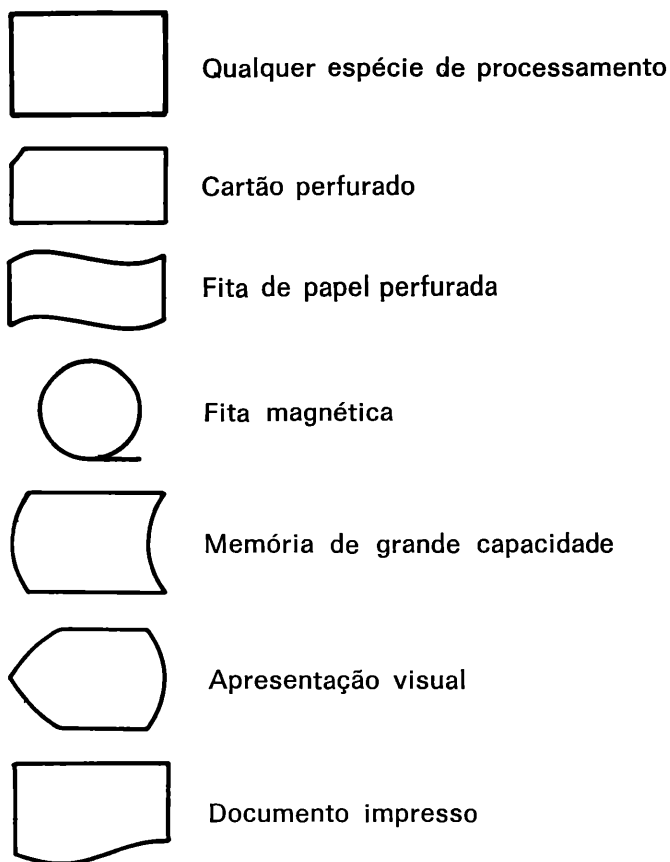


Figura 10.1.

Exemplificando, o fluxograma de um processo de atualização de saldos, que usa como elementos de entrada uma fita contendo os saldos anteriores e cartões de movimento, e que fornece como elementos de saída uma fita contendo os saldos atualizados e uma relação impressa desses saldos, tem o aspecto da figura 10.2 da página seguinte.

RESOLUÇÃO LÓGICA DO PROBLEMA

A resolução lógica do problema é a descrição detalhada das etapas lógicas de trabalho, ou seja, a seqüência das operações necessárias para a obtenção dos resultados finais desejados, a partir dos dados ini-

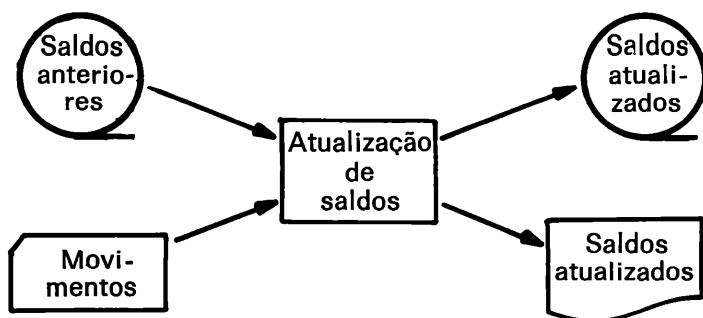


Figura 10.2.

ciais disponíveis. Esta é uma fase criativa, enquanto as duas etapas anteriores eram apenas descritivas.

O resultado desta etapa de planejamento é um gráfico denominado FLUXOGRAMA DE PROGRAMAÇÃO ou DIAGRAMA DE BLOCOS, que mostra, com maior ou menor detalhe, a lógica de determinado programa, ou seja, a seqüência das operações e decisões que constituem o programa.

No fluxograma de programação, cada etapa da solução lógica do problema é representada por um símbolo, cuja forma identifica o tipo de processo envolvido. Estes símbolos recebem o nome genérico de blocos, daí o nome “Diagrama de Blocos”.

A seguir, apresentaremos alguns desses símbolos.

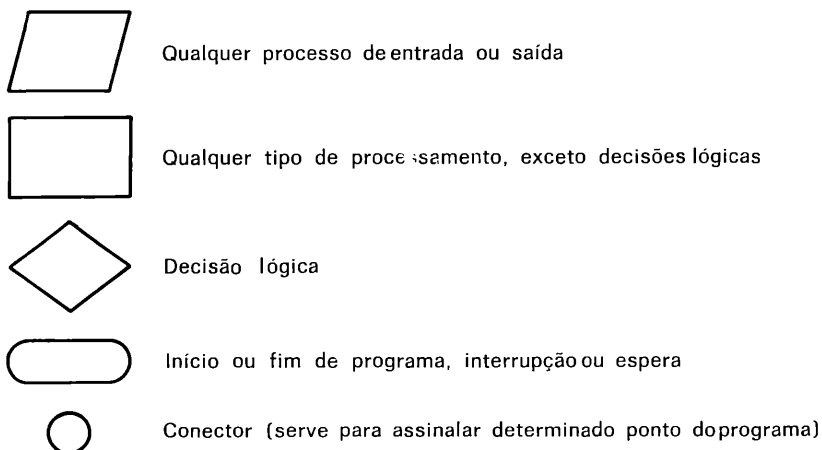


Figura 10.3.

EXEMPLOS DE FLUXOGRAMA DE PROGRAMAÇÃO

Sabe-se que quatro campos de memória M_1 , M_2 , M_3 , M_4 contêm, respectivamente, os valores A, B, C e D. Fazer o diagrama de uma rotina destinada a calcular a soma desses quatro valores, armazenando-a no campo M_5 .

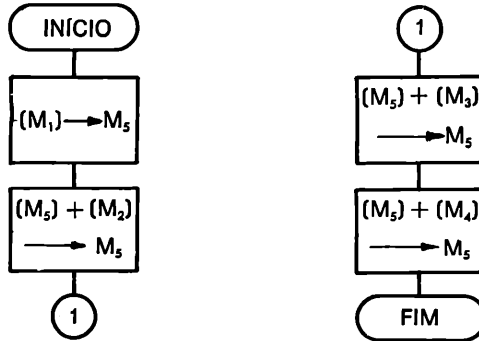


Figura 10.4.

A título de curiosidade, apresentamos na página seguinte um exemplo clássico de diagrama de blocos, intitulado: “Como procedemos quando levantamos de manhã”.

PROGRAMAÇÃO

A fase de programação propriamente dita consiste em traduzir a lógica da solução, expressa pelo diagrama de blocos, para a linguagem específica de dado computador, a fim de obter o programa.

Geralmente, o trabalho de programação é feito em duas etapas:

- Primeiramente, o programador traduz a lógica da solução para alguma linguagem simbólica, escrevendo um programa simbólico (programa-fonte).
- A seguir, o próprio computador se encarrega de traduzir este programa simbólico para sua linguagem específica, produzindo-se, assim, o programa em linguagem de máquina (programa-objeto).

O resultado dessa etapa do trabalho é um programa que, entretanto, ainda não está pronto para ser executado, pois, geralmente, apresenta uma série de erros e imperfeições que deverão ser eliminados num processo denominado “depuração”.

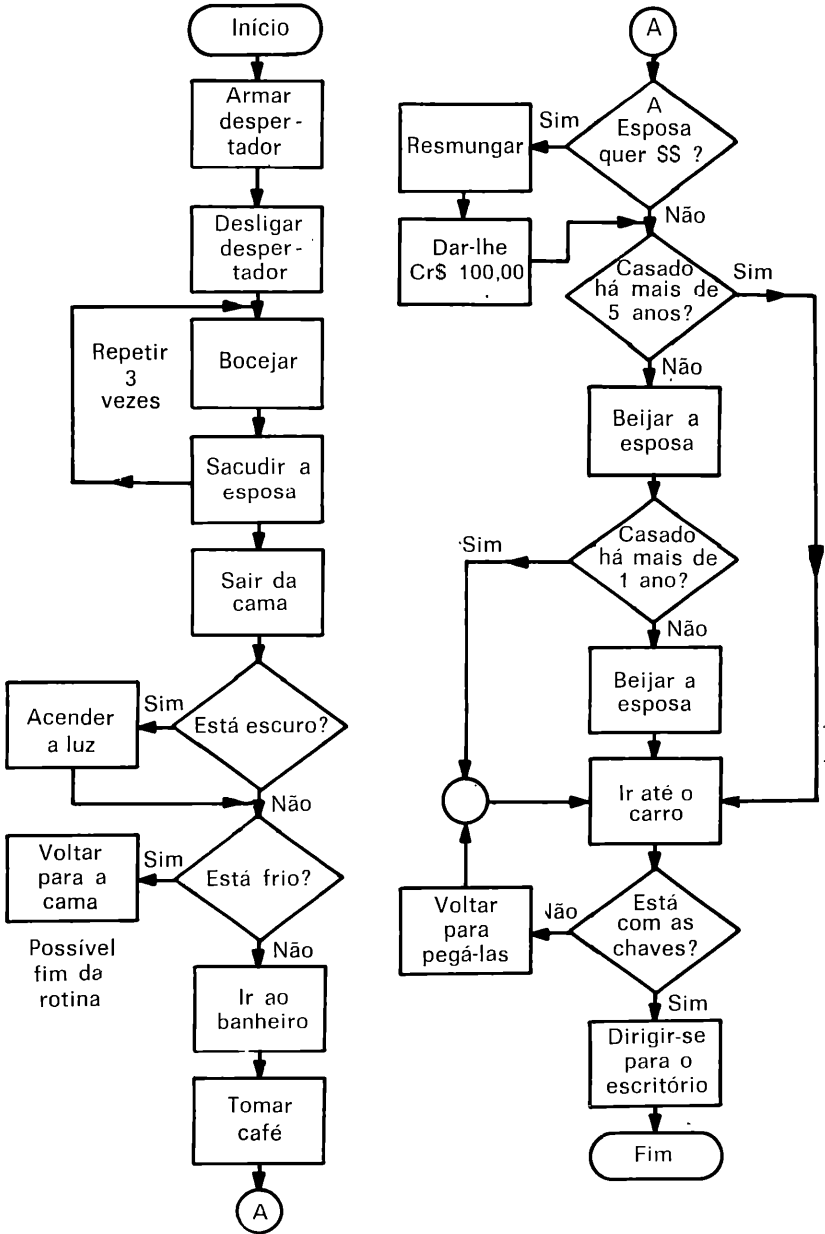


Figura 10.5.

DEPURAÇÃO

Nesta fase, o programador elimina os erros que, porventura, existam no programa. Esses erros podem ser de codificação ou de lógica.

Os erros de codificação (por exemplo, maneira errada de especificar um endereço, falta de um elemento numa instrução etc.) geralmente são percebidos pelo próprio computador, na fase de tradução, e indicados por uma mensagem de erro, podendo assim ser retificados pelo programador.

Quanto aos erros de lógica, há duas maneiras de detectá-los: mediante um exame cuidadoso do diagrama de bloco e através de dados de teste, ou seja, procurando executar o programa com dados especialmente preparados, cujos resultados finais são previamente conhecidos.

DOCUMENTAÇÃO

Nesta fase, prepara-se um registro escrito do serviço, em todos os seus aspectos: fluxograma, diagrama de blocos, listagem do programa, desenhos de cartões e/ou registros, modelos de formulários e instruções de operação.

A documentação completa e atualizada do serviço é da mais alta importância e destina-se a múltiplas finalidades: dar conhecimento das características do serviço aos funcionários do Centro, servir de base para revisões e atualizações, servir como ponto de referência para a implantação de serviços análogos etc.

LINGUAGENS SIMBÓLICAS DE PROGRAMAÇÃO

Um programa, em linguagem de máquina, é uma sucessão de instruções que, por sua vez, são seqüências de bits 0 e 1. Por exemplo, uma instrução a ser executada por um computador poderia ser a seqüência de bits:

00011000 00011011 (A)

que poderia ser representada, em forma abreviada, pelos caracteres hexadecimais:

1 8 1 B (B)

É evidente que escrever um programa diretamente sob a forma (A), ou mesmo sob a forma (B), é um trabalho extremamente monótono e demorado, além de cansativo, o que, por sua vez, dá origem a numerosos erros.

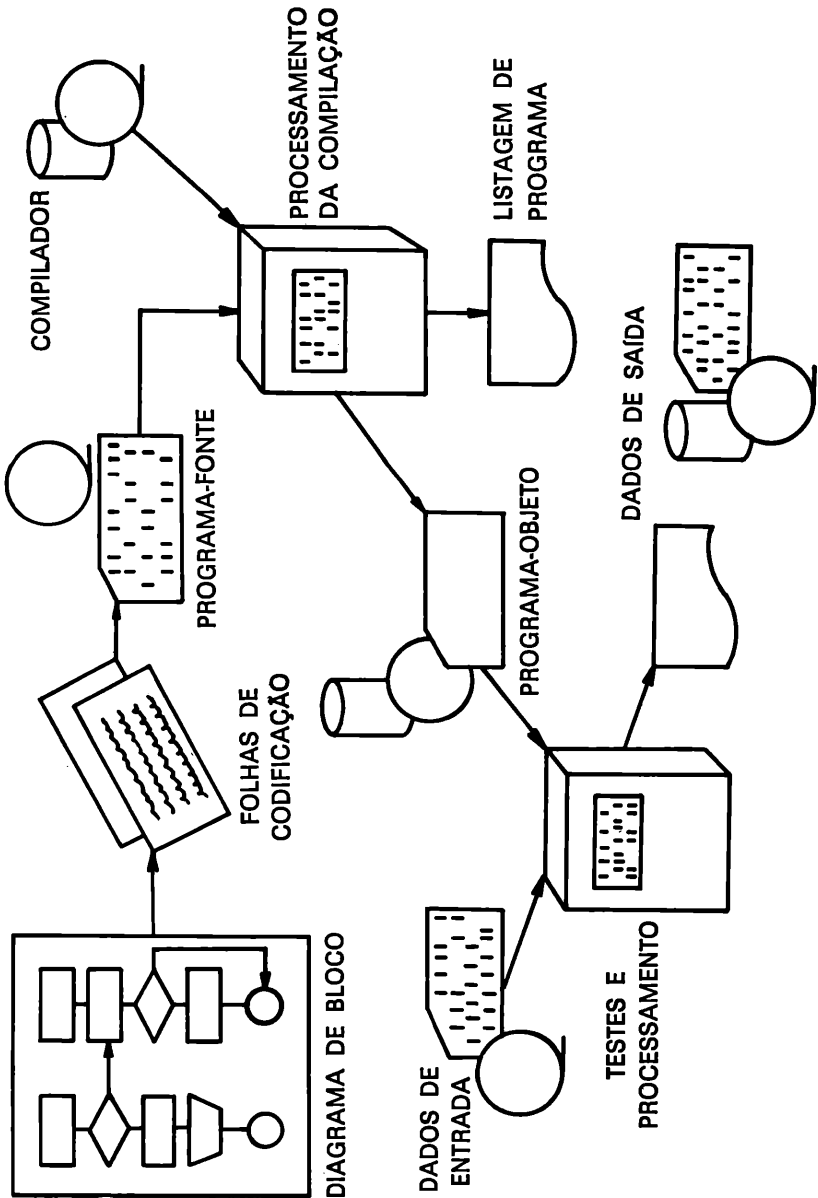


Figura 10.6 — Evolução de um sistema de processamento de dados.

Seria conveniente escrever as instruções diretamente em linguagem fluente; por exemplo:

TRANSFERIR PARA O REGISTRADOR 1 O CONTEÚDO DO REGISTRADOR 11.

Infelizmente, até o momento não existe a possibilidade de programar desta maneira. O que se fez, entretanto, foi criar uma série de linguagens simbólicas de programação, com características intermediárias entre a linguagem humana e a linguagem interna de máquina. Essa linguagem é suficientemente semelhante à linguagem humana para poder ser entendida por alguém familiarizado com certas regras, e, por outro lado, assemelha-se suficientemente à linguagem de máquina, para permitir que determinado programa faça a conversão da linguagem simbólica para a linguagem de máquina.

O exemplo seguinte ilustra o conceito de "Linguagem Intermediária" entre a linguagem humana e a linguagem interna da máquina.

LINGUAGEM HUMANA	MULTIPLICAR O NÚMERO DE HORAS PELA TAXA
LINGUAGEM INTERNA DA MÁQUINA	F C 5 2 3 2 A 4 3 2 C 8 ou 11111100 01010010 00110010 10100100 etc.
LINGUAGEM SIMBÓLICA	MP HORAS, TAXA

Como vemos, a linguagem simbólica permite que se tenha uma idéia da natureza da operação e dos elementos envolvidos, embora não seja fluente e redundante como a linguagem humana.

É importante notar que não basta criar a linguagem simbólica de programação; é preciso criar, também, o programa conversor que traduz um programa simbólico para a linguagem interna da máquina.

A seguir, vamos enumerar, sucintamente, algumas das vantagens decorrentes do uso de linguagens simbólicas de programação:

- 1 — Redução do tempo necessário para a programação.
- 2 — Portanto, redução no custo de programação.
- 3 — A programação em linguagem simbólica é um trabalho menos monótono e menos cansativo, o que acarreta um menor número de erros.

- 4 — Devido ao menor número de erros, haverá uma redução no tempo necessário para o teste de programa.
- 5 — Facilidade de introduzir alterações e correções num programa já existente.
- 6 — Facilidade para escrever rotinas padronizadas a serem utilizadas em vários programas.

Linguagem de alto nível e de baixo nível

As linguagens simbólicas de programação dividem-se em duas categorias: linguagens de alto nível e linguagens de baixo nível. As linguagens de *alto nível* são as linguagens orientadas para o problema. Estão nesta categoria, por exemplo, as linguagens FORTRAN, COBOL e ALGOL. As linguagens de *baixo nível* são orientadas para a máquina. Como exemplo de linguagem de baixo nível podemos citar o ASSEMBLER.

PRINCIPAIS CARACTERÍSTICAS DE UMA LINGUAGEM DE ALTO NÍVEL

- 1 — A linguagem é mais orientada para o problema, de modo que o programador deve dar pouca ou nenhuma atenção às características do computador que irá executar o programa.
- 2 — Um programa escrito numa linguagem simbólica de alto nível, para dado modelo de computador, pode ser aproveitado para um outro modelo de computador. Basta repetir o processo de compilação, utilizando o programa conversor do outro modelo de computador. (Geralmente, é preciso fazer algumas pequenas adaptações no programa simbólico.)
- 3 — O número de sentenças a escrever, em linguagem simbólica, é muito menor do que o número de instruções que terá o programador em linguagem de máquina. Isto acontece porque o compilador tem a capacidade de “compilar”, isto é, encaixar no programa sub-rotinas já existentes numa biblioteca de rotinas. Por exemplo, em FORTRAN, o programa pode requisitar uma complexa rotina matemática com uma única sentença.

PRINCIPAIS CARACTERÍSTICAS DE UMA LINGUAGEM DE BAIXO NÍVEL

- 1 — A linguagem é inteiramente orientada para a máquina. Portanto, o programador que vai escrever um programa numa linguagem de baixo nível deve saber para que modelo de máquina está programando e conhecer o repertório de instruções e as características desse modelo de máquina.

- 2 — Em geral, um programa escrito numa linguagem de baixo nível, para dado modelo de computador, não pode ser utilizado para um outro modelo, a não ser que os dois modelos de máquina sejam compatíveis em programação.
- 3 — O número de sentenças a escrever, numa linguagem simbólica de baixo nível, é aproximadamente igual ao número de instruções que o programa conversor produzirá, em linguagem de máquina.

Motivos para usar linguagens de baixo nível

Examinando as enumerações de características feitas nos parágrafos precedentes, parece, à primeira vista, que as linguagens de alto nível só apresentam vantagens, e as de baixo nível só apresentam desvantagens, de modo que se impõe a pergunta: Por que havemos de querer utilizar uma linguagem de baixo nível?

Existem vários motivos para isto. Em primeiro lugar, o programa conversor de uma linguagem de alto nível (compilador) exige mais memória do que o programa conversor de uma linguagem de baixo nível (montador). Assim sendo, para certas configurações de pequena capacidade de memória, somos obrigados a recorrer a uma linguagem de baixo nível (Assembler).

Independentemente disto, entretanto, o Assembler apresenta duas importantes vantagens em relação às linguagens de alto nível:

- Em linguagem de baixo nível, consegue-se um programa que ocupará menos espaços na memória, porque o programa é escrito para fazer o estritamente necessário, enquanto um programa produzido a partir de COBOL ou FORTRAN realiza determinadas operações de maneira generalizada e inclui passagens desnecessárias para o caso particular do programa em elaboração.
- Em linguagem de baixo nível, o programador está “mais perto da máquina”, tendo assim a possibilidade de usar artifícios de programação, aproveitando os recursos específicos de dado computador, de modo a obter uma redução na área de memória ocupada, ou no tempo de execução, ou em ambos.

A seguir, faremos uma descrição sucinta das linguagens mais usuais de programação.

Assembler

O Assembler, como já dissemos, é uma linguagem de baixo nível, que depende das características específicas de determinado computador.

Nesta linguagem, o processamento a realizar é descrito minuciosamente; o programador deverá, praticamente, escrever uma sentença em linguagem simbólica para cada instrução que deseja obter em linguagem de máquina.

O programa simbólico é escrito em formulários especiais, e recebe o nome de *programa-fonte*. O programa conversor transforma este programa simbólico num programa em linguagem de máquina, chamado *programa-objeto* ou *módulo*.

Exemplificando, em linguagem Assembler, a instrução que transfere o conteúdo de um campo chamado BETA para um campo chamado ALFA é escrito da seguinte maneira:

MVC ALFA, BETA

As letras MVC constituem um *código de operação mnemônico* proveniente do nome *MoVe Characters*.

Cobol

A linguagem COBOL destina-se a aplicações comerciais. A palavra COBOL é formada com as iniciais do nome *COMMON BUSINESS ORIENTED LANGUAGE* (Linguagem Comum Orientada para Aplicações Comerciais).

Nesta denominação, a palavra “Comum” refere-se ao fato de que a linguagem foi feita para ser comum a todos os computadores. Com efeito, o COBOL se aplica, com pequeníssimas alterações, a numerosos computadores dos mais diferentes modelos e marcas.

Embora até hoje não exista um sistema de programação em linguagem fluente, o COBOL é a linguagem que mais se aproxima deste ideal.

Exemplificando, a instrução que movimenta o conteúdo de um campo chamado BETA para um campo chamado ALFA, em COBOL, se escreve:

MOVE BETA TO ALFA

o que, em inglês, é praticamente linguagem fluente.

Um programa escrito em linguagem COBOL tem uma estrutura-padrão, compondo-se de 4 divisões, com os seguintes nomes:

IDENTIFICATION DIVISION

(Contém a identificação do programa.)

ENVIRONMENT DIVISION

(Contém a descrição do equipamento que deve fazer a compilação e do equipamento que deve executar o programa depois de compilado.)

DATA DIVISION

(Contém a descrição dos formatos dos dados de entrada e dos dados de saída.)

PROCEDURE DIVISION

(Contém o programa propriamente dito, ou seja, a descrição das operações a serem realizadas pelo computador.)

Fortran

O compilador FORTRAN é um compilador de finalidade matemática. A palavra FORTRAN é formada com as sílabas iniciais da frase *FOR*mula *TRAN*slation Language (Linguagem de Tradução de Fórmulas).

Esta linguagem permite que se escreva, de maneira simbólica, uma expressão matemática, e gera o programa que calcula o valor dessa expressão matemática.

Exemplificando, se quisermos calcular o valor da expressão $X = b^2 - 4AC$, podemos escrever, em linguagem FORTRAN, a seguinte sentença:

X=B2-4*A*C**

e o compilador FORTRAN produz o trecho de programa correspondente. Outra particularidade do compilador FORTRAN é a capacidade de incluir no programa, que está sendo compilado, rotinas já preparadas, que se encontram numa biblioteca de rotinas. Por exemplo, para calcular a raiz quadrada de um valor Y, podemos escrever a sentença: $X = \text{SQRTY}$, e esta sentença inclui no programa a rotina de extração de raiz quadrada (*S*quare *R*oot).

Algol

O ALGOL (*ALGO*rithmic *L*anguage) é um compilador avançado para fins matemáticos; seu uso é menos difundido.

PL/1

O compilador PL/1 (*Program*ming *L*anguage 1), criado em 1966, reúne características do COBOL e do FORTRAN. É provável que a linguagem PL/1 venha a assumir, num futuro próximo, um papel de destaque entre as linguagens simbólicas de programação.

Geradores de Relatório (RPG — Report Program Generators)

Um gerador de relatório é um sistema de programação destinado a produzir um programa capaz de fazer uma listagem, em determinado formato.

O programador, ao usar o RPG, não está criando um programa novo.

O arcabouço básico do programa de listagem já existe, e o programador apenas preenche determinados formulários para descrever as características específicas da listagem que deseja fazer; será, então, produzida uma variante do programa-padrão, de acordo com as especificações fornecidas pelo programador.

Apesar de ser um recurso bastante útil em determinados casos, o RPG é um sistema de programação muito limitado.

II

O CONCEITO DE SOFTWARE — OS PRINCIPAIS RECURSOS DE SOFTWARE

Nos primeiros tempos do processamento eletrônico de dados (aproximadamente entre 1951 e 1960), o próprio usuário fazia o programa de todas as fases do processamento, muitas vezes em linguagem de máquina. Isto dificultava sobremaneira a utilização dos computadores e constituía, mesmo, um entrave à maior difusão da computação eletrônica. Em vista disso, os fabricantes começaram a desenvolver uma série de recursos auxiliares destinados a tornar mais fácil e mais eficiente o trabalho de programação. Hoje em dia, o usuário conta com uma ampla variedade de recursos auxiliares, cuja importância é igual, senão superior, à importância da própria máquina.

Usa-se o termo **HARDWARE** para designar o próprio computador, seus circuitos, dispositivos e unidades, enquanto o termo **SOFTWARE**, para os programas, rotinas e técnicas que constituem os recursos auxiliares de programação.

Como já dissemos, o *software* teve um desenvolvimento extraordinário. Hoje em dia, a simples descrição do *software* de um sistema avançado pode abranger centenas de páginas impressas. Os inúmeros recursos de *software* podem ser divididos nas seguintes áreas:

- 1) programas conversores de linguagens;
- 2) rotinas de controle;
- 3) rotinas utilitárias;
- 4) programas de aplicação.

Nas páginas precedentes, já foram descritas as principais linguagens simbólicas de programação, fazendo-se menção aos respectivos programas conversores. A seguir, examinaremos, sucintamente, os demais recursos de *software*.

ROTINAS DE CONTROLE

As rotinas de controle são programas isolados, ou conjuntos de programas, que os fabricantes de computadores elaboram e colocam à disposição do usuário, para o controle do funcionamento do sistema.

Nos sistemas de segunda geração, as rotinas de controle eram muito simples e se limitavam, geralmente, à parte de controle das várias unidades periféricas.

Com o aparecimento dos sistemas de terceira geração, mais complexos e dotados de maiores recursos de processamento, foi necessário ampliar o âmbito das rotinas de controle, tendo em vista condições novas de operação, tais como a simultaneidade do processamento com as operações de entrada e saída, a execução simultânea de dois ou mais programas (multiprogramação), os usos de teleprocessamento etc.

Diante disso, foram escritas inúmeras rotinas de controle, já não mais rotinas isoladas e sim rotinas integradas, que formam Sistemas Operacionais.

Existem diversos tipos de Sistemas Operacionais, dependendo dos recursos do computador, cujo funcionamento irão controlar.

Os principais tipos são:

- TOS (Tape Operating System) — Este sistema destina-se a instalações que possuem fita magnética, mas não possuem discos magnéticos.
- DOS (Disc Operating System) — Este sistema destina-se a instalações que possuem discos magnéticos, podendo ou não ter fitas magnéticas.
- OS (Operating System) — Este é um sistema operacional mais amplo, orientado para memória de massa e teleprocessamento, destinado a permitir um grau elevado de multiprogramação.

ROTINAS UTILITÁRIAS

As rotinas utilitárias são rotinas destinadas a determinadas tarefas executadas com frequência numa instalação, e que o fabricante coloca à disposição do usuário, tirando-lhe assim o encargo de escrever os programas para essas tarefas.

As rotinas utilitárias podem ser divididas nas seguintes categorias:

- a) *Rotinas de manutenção de bibliotecas*
(Rotinas destinadas à criação, ampliação, modificação e atualização de bibliotecas de programas.)
- b) *Rotinas de fusão e classificação (Sort/Merge)*
(Rotinas destinadas à fusão e/ou classificação de arquivos de dados.)

- c) *Rotinas de serviço*
(Rotinas auxiliares para o tratamento de fitas magnéticas e discos magnéticos, executando tarefas tais como a criação e a verificação de blocos de identificação, a atribuição de áreas de memória, a atribuição de unidades periféricas, a duplicação de arquivos etc.)
- d) *Rotinas diagnósticas*
(Rotinas destinadas a facilitar o trabalho de teste de programas, executando tarefas tais como despejo de memória, geração de dados de teste, relato impresso da execução de um programa etc.)
- e) *Rotinas de conversão de unidades periféricas*
(Rotinas que transcrevem a informação de um meio para outro, por exemplo, de cartão para fita, de fita para disco, de disco para formulário, de um carretel de fita para outro, de um disco para outro etc.)

PROGRAMAS DE APLICAÇÃO

Os programas de aplicação (Application Packages) são programas ou conjunto de programas destinados a uma aplicação específica, como folha de pagamento, controle de estoque, programação linear, PERT/CPM etc. Quando o serviço do usuário se enquadra nas características do programa de aplicação do fabricante, ele poderá usá-lo, economizando, assim, o trabalho e o custo de programação.

A SEGURANÇA DO PROCESSAMENTO ELETRÔNICO DE DADOS

Num sistema de processamento eletrônico de dados, além dos trabalhos de processamento propriamente dito, tomam-se diversas providências, a fim de assegurar a segurança de funcionamento. Isto, além de ser uma evidente necessidade prática, tem também um importante valor psicológico. De fato, é perfeitamente compreensível que o usuário tenha, inicialmente, certo receio da máquina para ele desconhecida, e uma certa relutância em confiar seus preciosos dados à memória invisível, situada nas entranhas da máquina. Por isso, é duplamente importante adotar uma série de providências acauteladoras, a fim de lhe inspirar confiança e vencer o receio inicial.

A segurança do processamento envolve dois aspectos básicos:

- a) exatidão dos resultados;
- b) proteção dos dados.

Exatidão dos resultados

A exatidão dos resultados, produzidos pelo processamento eletrônico (isto é, o fato de serem resultados corretos), depende, praticamente, apenas da exatidão dos dados iniciais. Com efeito, uma vez produzido e testado o programa e fornecidos dados iniciais corretos, é praticamente impossível a obtenção de dados errados. Uma Unidade Central em perfeitas condições de funcionamento não erra nunca; por outro lado, se os circuitos eletrônicos tiverem algum defeito, começarão a aparecer mensagens de “Erro de Paridade”, as quais constituem uma indicação de que algo vai mal. Em contrapartida, com dados iniciais errados, obteremos, certamente, resultados errados. Os americanos designam esta verdade acadiana com a frase “Garbage In — Garbage Out”, que poderíamos traduzir por: “Entra Lixo — Sai Lixo”, onde a palavra “Lixo” designa uma informação incompleta, isto é, sem valor.

Note-se que uma informação não precisa estar errada para constituir “Lixo”; basta que não tenhamos certeza de que ela está certa, e ela se torna inaproveitável.

Por todas estas razões, é de extrema importância a verificação da exatidão dos dados iniciais. A seguir, descreveremos alguns métodos utilizados para esta verificação.

Verificação dos dados iniciais

Normalmente, os dados iniciais são transcritos num cartão perfurado ou numa fita perfurada, ou mesmo gravados diretamente em fita magnética, a partir de algum documento original, e esta fase de transcrição (que envolve o elemento humano) representa a maior fonte de erros, razão pela qual a maioria das providências de verificação se concentra na transcrição de dados. As providências mais comuns para a verificação da transcrição manual de dados são as seguintes:

- a) Transcrição repetida;
- b) totais por lotes;
- c) dígitos de controle;
- d) crítica (verificação de coerência).

A *transcrição repetida* é usada principalmente quando o veículo da informação é o cartão perfurado. Neste caso, a informação do documento original é transcrita duas vezes, geralmente por duas pessoas diferentes. A primeira perfuradora faz a transcrição, produzindo efetivamente os cartões perfurados. A segunda perfuradora recebe os mesmos documentos originais, na mesma ordem, e os cartões já perfurados. Essa segunda pessoa usa uma máquina diferente, chamada verificadora, que não produz perfuração, mas ape-

nas verifica se a tecla apertada corresponde à informação já perfurada no cartão.

Outro método de verificação é o dos *totais por lotes*. Neste caso, os documentos originais são agrupados em lotes, e o total de cada lote é calculado previamente. Ao processar esses dados, o computador recebe as informações transcritas e o total por lote, e pode comparar este total com o total apurado pelo processamento.

A técnica de *dígitos de controle* tem o intuito de evitar um tipo muito comum de erro de transcrição, que consiste na inversão de caracteres (erro de transposição). Esse erro é cometido, por exemplo, quando se perfura o número 125871 em vez de 125781, e pode fazer com que um valor seja creditado ou debitado a um número de conta errado. Para evitar os erros de transposição, acrescentamos aos dígitos de informação um ou mais dígitos de controle, obtidos através de algum processo matemático. A título de ilustração, vamos apresentar um exemplo de uma técnica muito simples de dígito de controle.

Consideremos o número 125781, citado acima.

Somando o primeiro, o terceiro e o quinto algarismos, a partir da esquerda, obtém-se $1 + 5 + 8 = 14$, e, desprezando o algarismo das dezenas, temos 4, que é o primeiro dígito de controle; a seguir, somando o segundo, o quarto e o sexto algarismos, a partir da esquerda, obtém-se $2 + 7 + 1 = 10$, e, desprezando o algarismo das dezenas, temos 0, que é o segundo dígito de controle, e o "número de controle" é 40. Então, a perfuradora deverá perfurar: 12578140, e o computador refaz o cálculo citado acima, com os 6 primeiros algarismos, comparando o resultado com o número de controle.

Suponhamos, agora, que a perfuradora, por engano, tenha perfurado 12587140. Se o computador fizer os cálculos para obter o número de controle, obterá:

$$1 + 5 + 7 = 13$$

$$2 + 8 + 1 = 11$$

Número de controle: 31.

Portanto, o computador percebe que ocorreu um erro de perfuração.

A *crítica* dos dados (verificação de coerência) é feita por um programa que aplica uma série de critérios para verificar se os dados apresentam alguma anomalia em relação aos valores esperados. Por exemplo, o valor do salário líquido de um funcionário, em condições normais, não deve ultrapassar certo valor. Fazendo uma comparação com determinado limite, detectam-se erros como, por exemplo, o de um salário de Cr\$ 1.500,00, que foi convertido em Cr\$ 15.000,00, por

ter sido perfurado nas colunas erradas. Outro exemplo de crítica de coerência: o cartão de vencimentos de um horista não pode ter gratificação de função, e o de um mensalista não pode ter indicação de horas normais.

Proteção de dados

A proteção dos dados destina-se a evitar a perda de informações devido a falhas de máquina ou a erros de operação. Estes métodos de proteção aplicam-se, principalmente, aos dados armazenados em fita magnética e disco magnético.

PROTEÇÃO DOS DADOS GRAVADOS EM FITA MAGNÉTICA

O maior risco a que estão expostos os dados da fita magnética é a desgravação acidental, devido a um erro de operação. Para evitar esta desgravação, podemos adotar três providências:

- a) Uso de anel de proteção;
- b) uso da data de expiração;
- c) reenrolamento com bloqueio.

O anel de proteção (anel de permissão de escrita) é um anel de plástico inserido no carretel de fita, a fim de *permitir* que esta fita receba gravação. Quando a fita contém dados importantes que não devem ser destruídos, basta retirar do carretel o anel de proteção. A unidade de fita não aceita um comando de gravação se o carretel estiver sem o anel de permissão de escrita, o que torna impossível a destruição dos dados já existentes nessa fita.

Outro método de proteção consiste em colocar no bloco de identificação de um arquivo (Label Block) a data de expiração, isto é, até a data em que devem ser preservados os dados contidos na fita.

A rotina de controle de fita magnética, antes de gravar dados num rolo de fita (destruindo, assim, seu conteúdo anterior), compara a data do dia com a data de expiração. Se a data do dia for anterior à data de expiração, o comando de gravação não será executado.

O reenrolamento com bloqueio (rewin with unload) consiste no seguinte: normalmente, após o término da gravação de um rolo de fita, este é reenrolado até o "Load Point".

Isto permite que, por uma falha de programação ou descuido do operador, após o reenrolamento, sejam gravados novos dados na mesma fita, destruindo os dados que acabaram de ser obtidos. Neste caso, o movimento de retrocesso passa do Load Point, o que torna impossível uma nova gravação neste rolo de fita, antes que o operador coloque a fita no Load Point, apertando uma tecla na própria unidade de fita.

Além destas providências, destinadas a impedir a perda acidental das informações, existem medidas destinadas a permitir a reconstituição dos dados, caso ocorra alguma falha. Dentre essas medidas podemos citar a técnica das três gerações.

A técnica das três gerações (também chamada técnica de avô, pai e filho) consiste em guardar sempre três gerações consecutivas de fita. O conceito de geração é o seguinte: quando se tem uma fita-mestre F-1 e se processa esta fita para obter uma fita-mestre atualizada F-2, dizemos que a fita F-2 é da geração seguinte à de F-1. Por sua vez, processando a fita F-2 e obtendo uma nova fita atualizada F-3, dizemos que F-3 é de uma geração seguinte à de F-2. Portanto, as fitas F-1, F-2 e F-3 constituem 3 gerações consecutivas de fita magnética. Conservando sempre 3 gerações, se houver a perda acidental dos dados da fita F-3, podemos reconstituí-la, processando a fita F-2 juntamente com os dados de alteração que produziram a fita F-3. E se, por uma coincidência extraordinária, forem destruídos também os dados da fita F-2, dispomos ainda da fita F-1 para reconstituir a fita F-2.

PROTEÇÃO DOS DADOS GRAVADOS EM DISCO MAGNÉTICO

O processamento de arquivo de disco magnético é essencialmente diferente do processamento de arquivos de fita magnética. No trabalho com fita magnética, a partir de um "arquivo anterior", é produzido um "arquivo atualizado", num carretel de fita diferente; após o processamento, dispomos tanto do arquivo anterior como do arquivo atualizado, e podemos guardar o rolo de fita que contém o arquivo anterior, para o sistema de proteção de avô, pai e filho.

No caso de processamento de arquivos em disco, o registro modificado é escrito por cima do registro anterior, destruindo-o. Portanto, após o processamento, dispomos do arquivo atualizado, mas não temos mais o arquivo anterior. Para termos a possibilidade de reconstituição dos dados, em caso de algum incidente, podemos proceder de várias maneiras, das quais citaremos duas.

A primeira consiste em duplicar o arquivo anterior, em outro disco, antes de efetuar as modificações. A segunda consiste em gravar periodicamente (por exemplo, no início de cada dia) o arquivo de disco numa fita magnética, e fazer um registro gravado de cada alteração efetuada, mediante uma rotina que grava numa fita, para cada alteração, o registro em sua forma anterior e o registro alterado.

ALGUNS CONCEITOS DE PROCESSAMENTO DE DADOS RELACIONADOS COM O GRAU DE SIMULTANEIDADE OPERACIONAL

Introdução

O aparecimento de computadores eletrônicos digitais, na década de 1950, representou um enorme avanço tecnológico. No entanto, esses primeiros computadores tinham um conceito de utilização extremamente rudimentar. Dado computador só tinha capacidade de executar um serviço de cada vez, e em seqüência linear: leitura do primeiro registro, processamento do primeiro registro, saída do primeiro resultado; leitura do segundo registro, processamento do segundo registro, saída do segundo resultado, e assim por diante. Nesta modalidade de trabalho, em cada instante apenas uma das unidades do sistema estava sendo utilizada, e as demais estavam ociosas.

Dessa época para cá, registrou-se um progresso extraordinário, não somente na tecnologia de construção dos computadores, mas também nos conceitos de utilização. A seguir, examinaremos alguns desses conceitos.

Funcionamento simultâneo do processador e das unidades de entrada e saída

O conceito de buffer

O conceito de "buffer" apareceu com o intuito de permitir que o processamento continuasse a ser realizado mesmo durante o período em que estava ocorrendo uma operação de entrada ou saída, ou que estes dois tipos de operação tivessem pelo menos uma parte do tempo em comum. Nos primeiros computadores, as unidades periféricas mandavam os dados diretamente à memória principal, utilizando para isto os mesmos circuitos destinados às operações aritméticas e lógicas, o que impedia a realização do processamento durante as operações de entrada ou saída. O "buffer" era, inicialmente, uma memória especial, independente da memória principal, destinada à armazenagem temporária das informações que uma unidade de entrada introduz ou uma unidade de saída retira da memória. Graças a este recurso, podia haver processamento durante quase todo o tempo da operação de E/S.

O único intervalo de tempo em que o processamento está bloqueado é aquele em que se dá, efetivamente, a transferência da informação entre a memória e o "buffer", e este bloqueio (interlock) perfazia uma fração ínfima do tempo total de processamento.

Hoje em dia, existe uma concepção mais moderna, que permite a introdução (ou retirada) de dados na memória, praticamente sem bloquear o processamento, e sem necessidade de um "buffer" físico. A única unidade periférica cujas características exigem a conservação do "buffer" físico é a impressora. As demais unidades de E/S funcionam como se fossem dotadas de "buffer", sem que este realmente exista.

Vamos ilustrar este conceito com o caso particular da leitora de cartões. Nesta unidade, a passagem física de um cartão pelas fotocélulas de leitura exige algumas dezenas de milissegundos e, nos primeiros computadores, uma grande porcentagem deste tempo (em alguns casos, 70% ou mais) era gasta apenas na operação de leitura, com o processamento bloqueado. O "interlock" era contínuo, desde o momento em que a coluna 1 do cartão passava sob as fotocélulas até o término da leitura do cartão. Com o conceito atual, o bloqueio não é contínuo, e sim intermitente; cada vez que uma das 80 colunas do cartão está posicionada sob as fotocélulas, a informação desta coluna entra diretamente na memória, interrompendo a execução do programa durante alguns milionésimos de segundo. Esses bloqueios parciais, quando somados, não atingem um milésimo de segundo.

Interrupção de unidade periférica

Nos computadores mais antigos, em que uma operação de E/S provocava um bloqueio contínuo do processador central, o processamento e as operações de entrada e saída alternavam-se de modo natural: o programa chegava a um ponto em que precisava de uma informação de entrada ou queria escrever uma informação de saída; neste ponto, o programa dava a instrução correspondente a uma unidade periférica e ficava esperando o término da operação de E/S, para, só então, continuar o processamento. Hoje em dia, as coisas acontecem de maneira mais complexa. O programa emite uma instrução de entrada ou saída e, imediatamente após, continua o trabalho de processamento. Na realidade, o processamento pode estar ocorrendo simultaneamente com duas ou mais operações de entrada e saída e os tempos de execução das várias operações de E/S diferem, conforme a velocidade de cada unidade e a quantidade de caracteres a ler ou a escrever. Assim, sendo, o programa não tem mais meios de acompanhar o andamento de cada uma das operações periféricas, ou seja, de prever quando dada unidade (por exemplo, a impressora) terminará a execução de um comando de E/S e estará pronta para aceitar o comando seguinte.

Por esta razão, foi criado um mecanismo de controle, por meio do qual a unidade periférica que termina a execução de uma operação dá um aviso ao processador. Este aviso provocará uma interrupção

ção momentânea na execução do programa em andamento, sendo o controle transferido para uma rotina (ou um conjunto de rotinas) de controle, com o que são tomadas as providências que a interrupção requer.

Note-se que a *interrupção de entrada e saída* é apenas um dos tipos de interrupção que podem ocorrer num moderno sistema de processamento de dados.

Outros tipos de interrupção

Sem entrar em maiores detalhes, citaremos alguns outros tipos de interrupção dos atuais computadores eletrônicos.

- Interrupção de “hardware” — Causada por alguma falha do equipamento.
- Interrupção devida a erros de programa — Certos erros muito graves de programa podem produzir uma interrupção do processamento, como, por exemplo, o uso de um endereço maior do que a capacidade de memória da máquina, ou o uso de um código de operação inválido.
- Interrupção devida a erros nos dados — Alguns erros nos dados podem produzir uma interrupção do processamento, como, por exemplo, um divisor igual a zero.
- Interrupção do “relógio” — Em sistemas capazes de executar vários programas ao mesmo tempo, usa-se, muitas vezes, o sistema de atribuir “fatias de tempo” (time slicing) aos vários programas, numa base de rodízio. Nestes casos, usa-se um relógio eletrônico (timer) destinado a limitar o tempo atribuído a dado programa. Se dado programa utilizar, consecutivamente, um tempo superior a um valor prefixado (geralmente da ordem de alguns milissegundos), o “timer” produz uma interrupção do processamento.

Normalmente, a *interrupção no processamento* não significa a parada total das atividades do computador, mas uma transferência de controle, do programa que foi interrompido, para uma rotina especial que analisa a causa da interrupção e toma as providências adequadas para cada caso.

Voltando ao conceito de simultaneidade operacional, diremos que, acima do funcionamento simultâneo da unidade central de processamento e das unidades de entrada e saída, está a multiprogramação, que examinaremos a seguir.

MULTIPROGRAMAÇÃO

Dizemos que um sistema de processamento de dados opera em regime de multiprogramação quando ele tem dois ou mais programas

armazenados na memória e executa estes programas paralelamente. A operação é “paralela”, e não “simultânea”, pois o processador só pode executar uma instrução de cada vez.

O que existe é uma alternância, em que o sistema dedica alguns milissegundos de processamento ao primeiro programa, depois alguns milissegundos a um outro programa, e assim por diante. Um observador humano, entretanto, terá a impressão de que todos os programas estão sendo executados ao mesmo tempo, quando, na realidade, eles são executados em intervalos diferentes do mesmo período.

Geralmente, no trabalho em regime de multiprogramação, alguns dos programas são principais e fazem muito uso do processador, enquanto outros são apenas programas de conversão periférica, que utilizam uma fração mínima do tempo do processador.

MULTIPROCESSAMENTO

No multiprocessamento, os vários programas estão armazenados na memória ao mesmo tempo e são executados, ao mesmo tempo, numa verdadeira simultaneidade operacional. Neste caso, em dado microssegundo, o sistema pode executar duas instruções diferentes, pertencentes a programas diferentes. O multiprocessamento só é possível quando o sistema possui mais de uma unidade central de processamento, sendo que as várias unidades de processamento podem ter acesso ao mesmo sistema de memória.

Só um sistema de porte muito grande pode trabalhar em regime de multiprocessamento. No mundo inteiro, existe um número muito reduzido de tais sistemas.

PROCESSAMENTO EM LOTES E PROCESSAMENTO EM TEMPO REAL

Existem duas maneiras fundamentalmente diferentes de efetuar o processamento de dados: uma delas é o processamento em lotes (batch processing), e a outra é o processamento em tempo real (real time processing).

O processamento em lotes consiste em deixar acumular os dados referentes a determinado período (por exemplo, todos os pedidos que entraram em determinado dia) e processar todos estes dados de uma só vez. Esta modalidade de processamento apresenta a desvantagem de uma defasagem entre o momento em que se origina a informação e o momento em que ela é processada. Se tomarmos, por exemplo, um depósito bancário efetuado em dinheiro às três horas da tarde, é provável que esta informação somente seja processada várias horas

depois, digamos, à meia-noite. Se considerarmos um caso mais complexo, de um cliente que paga um título em determinada agência, que foi colocado em cobrança em outra agência, podem decorrer 24 horas ou mais entre o momento do pagamento e o momento do crédito ao cedente. Este atraso é eliminado com o conceito de *Processamento em tempo real*.

Nesta modalidade de trabalho, não esperamos até o fim do período para iniciar o processamento, mas processamos cada informação no momento em que ela ocorre. Isto faz com que as informações existentes na memória do computador estejam sempre atualizadas até o último segundo, o que é de grande importância em determinadas aplicações como, por exemplo, reservas de lugar em transporte aéreo. De fato, não nos adianta saber que o lugar estava disponível há duas horas atrás, ou há cinco minutos atrás; precisamos saber se o lugar está disponível *agora*.

O processamento em tempo real pressupõe que o computador tenha unidades de telecomunicação que lhe permitam receber informações e/ou consultas de terminais remotos, e que disponha de um sistema de memória de grande capacidade para armazenar o acervo de informações a ser processado em tempo real.

O que caracteriza o processamento em tempo real é o fato de que a estação remota que envia uma informação, e/ou uma consulta, espera receber uma resposta dentro de um intervalo de tempo inferior a um valor preestabelecido.

TELEPROCESSAMENTO

O teleprocessamento é a modalidade de operação que combina o processamento eletrônico de dados com a transmissão de dados à distância, através de qualquer tipo de linhas de comunicação (telefone, telégrafo, telex, microondas etc.).

O teleprocessamento pode ser "on-line" ou "off-line". No teleprocessamento "on-line", as mensagens recebidas de um local remoto entram diretamente na memória do computador e são processadas na hora. Isto é processamento em tempo real. No teleprocessamento "off-line", as informações provenientes de um terminal remoto são armazenadas em algum portador de dados (disco, fita magnética etc.), para serem processadas mais tarde.

MULTIACESSO SIMULTÂNEO (Time Sharing)

O multiacesso simultâneo é uma forma especial de multiprogramação, em que vários usuários têm acesso ao mesmo computador. Esta forma de processamento pressupõe a capacidade de operar no

sistema de tempo real. Cada usuário tem em suas instalações um terminal transmissor e receptor, diretamente ligado ao computador, que poderá estar localizado em outro bairro ou mesmo em outra cidade. Quando tem um serviço a executar, o usuário transmite um sinal pedindo uma parcela do tempo do computador. Os pedidos recebidos são atendidos pelo computador dentro de uma escala de prioridade. Quando o usuário recebe licença para “entrar na linha”, envia os dados ao computador, que os processa, e devolve os resultados através da linha de comunicação. O computador deverá ter memória bastante grande para armazenar os programas dos vários usuários. Além disso, deverá ter velocidade e capacidade de processamento suficientes para atender às diversas solicitações que recebe, muitas delas praticamente ao mesmo tempo.

O USO DAS TELECOMUNICAÇÕES NO PROCESSAMENTO DE DADOS

Atualmente, a telecomunicação está sendo usada em escala cada vez maior no processamento de dados. Isto ocorre porque os sistemas de processamento são cada vez mais poderosos e têm capacidade de atender não apenas a dado departamento, mas a toda uma organização ou mesmo a um grupo de organizações.

São inúmeras as possibilidades de utilização de sistemas de comunicações, e vamos citar apenas algumas.

Uma situação muito comum é aquela em que existe um Centro de Processamento de Dados na sede da empresa e, nas várias dependências da empresa, existem terminais de transmissão e recepção de dados.

Com isto, o computador pode ser informado instantaneamente dos fatos que ocorreram nos vários departamentos.

Outra situação é aquela em que a sede da empresa possui um computador de grande porte, que é centro de uma rede de computadores, sendo que, nas agências ou sucursais, existem computadores de porte menor. Neste caso, os computadores exigem grandes recursos de máquina. Se houver, ocasionalmente, necessidade de resolver um problema de maior complexidade, os dados são transmitidos ao computador maior, que resolve os problemas e devolve os resultados, através das linhas de comunicações. Outra vantagem prática de uma rede de computadores é a de permitir uma distribuição melhor da carga de trabalho. Se, em dado momento, algum dos computadores que integram o sistema está sobrecarregado de serviço e um outro apresenta folga, este segundo pode encarregar-se de parte do trabalho

do outro, recebendo os dados, efetuando o processamento e devolvendo os resultados.

Dois tipos de processamento, que têm como pressuposto básico o uso de telecomunicações, são o processamento em tempo real e o multiacesso simultâneo, já descritos num parágrafo anterior.

I2

O COMPUTADOR DENTRO DA EMPRESA

Qualquer organização pode beneficiar-se com a instalação de um computador eletrônico?

Não necessariamente. Existem certas condições para que a utilização do processamento eletrônico de dados seja técnica e economicamente recomendável. Essas condições relacionam-se com o tamanho da empresa, o volume de serviço e a natureza dos serviços.

É evidente que somente empresas ou organizações acima de certa dimensão têm condições econômicas para poderem adquirir um computador eletrônico. Aliás, há poucos anos, um computador custava mais de um milhão de dólares e somente organizações muito grandes podiam ter um computador; hoje, o custo de computadores reduziu-se bastante, de modo que eles já estão ao alcance das organizações média e pequena. Entretanto, é óbvio que uma oficina de encanador, ou uma quitanda, não têm possibilidades de ter um computador eletrônico. Isto decorre, apenas, do tamanho da empresa, e não do ramo de atividades; por exemplo, um serviço de assistência técnica com várias filiais e numerosos encanadores poderia usar um computador.

Outra consideração importante é o volume de serviço de processamento de dados, que não é, necessariamente, proporcional ao tamanho ou ao faturamento da empresa. Cita-se o caso de uma empresa distribuidora de produtos de petróleo cujo faturamento mensal vai a bilhões, mas que emite apenas algumas dezenas de faturas por mês. Neste caso, evidentemente, não há necessidade de computadores eletrônicos para fazer o faturamento.

Além do porte da empresa e do volume de serviços, é preciso levar em conta a natureza dos serviços. Se os serviços forem muito diversificados e difíceis de enquadrar em padrões básicos, a aplicação

de computadores eletrônicos será tecnicamente difícil e, provavelmente, antieconômica. Por outro lado, se tivermos um grande volume de um pequeno número de serviços básicos, teremos as condições ideais para utilizar com proveito o processamento eletrônico de dados.

VANTAGENS QUE O PROCESSAMENTO ELETRÔNICO DE DADOS PODE TRAZER À EMPRESA

Desde que existam as condições básicas que tornem técnica e economicamente recomendável a utilização de computadores eletrônicos, estes poderão trazer à empresa uma série de vantagens, tanto diretas como indiretas.

Vantagens diretas

- Serviços rápidos, limpos e sem erros.
- Capacidade de absorção de acréscimos esporádicos de serviço, sem necessidade de aumento de pessoal.
- Possibilidade de fazer serviços que, pelos métodos convencionais, eram inexecutáveis, excessivamente demorados ou antieconômicos.
- Obtenção de relatórios e estatísticas atuais, que permite à Administração identificar tendências que ainda estão atuando e tomar as providências correspondentes.

Vantagens indiretas

- Maior controle da Gerência e da Auditoria sobre as operações da empresa.
- Aumento do conceito da empresa entre seus clientes e seus próprios funcionários (Imagem da Empresa).
- Função preventiva em relação a atos lesivos à empresa, desde que os funcionários se sintam "fiscalizados" eletronicamente.
- Possibilidade de obtenção de uma renda mediante prestação de serviços a terceiros.

MODIFICAÇÕES QUE A INTRODUÇÃO DO COMPUTADOR PRODUZ NA EMPRESA

Em qualquer empresa, existe certa rotina para o fluxo de informações, certo sistema de trabalho. Essa rotina de trabalho depende dos recursos de que dispõe a empresa para o processamento de dados: máquinas de contabilidade, equipamento convencional ou computadores eletrônicos. Assim sendo, a conversão para a computação eletrônica acarreta, geralmente, uma reformulação dos processos de traba-

lho, que atinge não somente o setor de processamento de dados, mas, praticamente, todas as áreas da empresa. Especificamente, poderá haver mudança no sistema de coleta inicial de informações, modificação de formulários, eliminação de certos controles etc.

Em geral, estas modificações são para melhor, com tendência a reduzir a produção e a movimentação de papéis. Reduz-se a variedade de formulários, reduz-se o número de vias, reduz-se o trabalho de transcrição, elimina-se a duplicação de esforços.

Os controles que, antes, dependiam de um "paralelo" são agora feitos eletronicamente.

Um ponto importante é o seguinte: a conversão deve ser gradual, mantendo-se um serviço paralelo durante algum tempo, e só passando-o totalmente para o computador depois de aparadas todas as arestas.

PROBLEMAS MATERIAIS DA IMPLANTAÇÃO DO PROCESSAMENTO ELETRÔNICO DE DADOS NUMA EMPRESA

Como já dissemos, a conversão para a computação eletrônica requer uma reformulação da rotina de trabalho dentro da empresa.

Em particular, é preciso projetar novos tipos de formulários, desenhar cartões etc.

Entretanto, o principal problema de ordem material é a preparação do local para a instalação do computador. Em primeiro lugar, é preciso obter um local amplo (se possível, com reserva de espaço para futuras ampliações) e de localização conveniente. Depois, este local deve ser preparado especialmente para receber o computador: piso reforçado e à prova de abalos, entrada de força, iluminação etc. Um dos aspectos mais delicados da preparação de local, que freqüentemente ocasiona atrasos na instalação, é o sistema de ar condicionado. Por isto, este aspecto deve merecer atenção especial por parte dos usuários.

Atualmente, existem firmas especializadas em assessoria de preparação de local.

PROBLEMAS HUMANOS

Uma empresa que pretende passar a trabalhar com computadores eletrônicos deverá contar com uma série de problemas humanos e preparar-se para enfrentá-los.

Provavelmente a reação inicial dos funcionários será predominantemente negativa, principalmente entre os de mais idade. Há uma sé-

rie de razões para isso. Em primeiro lugar, todos nós temos certa relutância em aceitar inovações, devido à força do hábito e ao comodismo. No caso do processamento eletrônico de dados, a resistência à inovação tem uma série de motivos específicos: o receio de perder o emprego, o receio de não poder acompanhar o progresso tecnológico, o receio de perder prestígio etc.

Por todas estas razões, é importante que a empresa prepare psicologicamente os seus funcionários, mostrando-lhes que o objetivo do computador não é o de reduzir o número de funcionários, que ninguém perderá o emprego, que haverá novas oportunidades profissionais etc.

SITUAÇÃO DO CENTRO DE PROCESSAMENTO DE DADOS DENTRO DA EMPRESA

Um ponto de vital importância para o sucesso da instalação de um Centro de Processamento de Dados é a sua situação dentro do organograma de uma empresa.

No tempo das máquinas eletromecânicas (equipamento convencional), costumava-se enquadrar o CPD como uma das seções do Departamento, geralmente do Departamento de Contabilidade, porquanto as suas funções se restringiam a atividades contábeis.

Com o advento dos computadores eletrônicos, os Centros de Processamento de Dados ganharam novas funções e responsabilidades. Além das aplicações mais rotineiras dos negócios da empresa, os responsáveis pelos Centros de Processamento de Dados preocupam-se e desenvolvem projetos mais arrojados, cujo êxito depende, fundamentalmente, do cumprimento das normas e procedimentos ditados para o novo sistema, o que só se consegue dando ao CPD um nível hierárquico bastante elevado.

Dentro das empresas mais modernas, o CPD é hoje tratado em nível de Departamento, estando diretamente ligado aos executivos de cúpula (diretor, vice-presidente, diretor financeiro etc.).

A PARTICIPAÇÃO HUMANA NO PROCESSAMENTO ELETRÔNICO DE DADOS

Faremos uma descrição sucinta de algumas atividades profissionais relacionadas com o funcionamento de um Centro de Processamento de Dados. Falaremos sobre o trabalho do analista de sistemas, do programador e do operador, dos serviços de manutenção e da Chefia do Centro.

Uma coisa que o dirigente de empresa deve ter sempre em mente é que todas estas categorias profissionais exigem elementos especiali-

zados e altamente qualificados, que devem estar constantemente motivados por suas condições de trabalho e por uma remuneração adequada. Tentar economizar algumas centenas de cruzeiros no salário de um desses profissionais é fazer uma economia no lugar errado, e isto poderá até mesmo comprometer a eficiência de funcionamento do C.P.D.

O trabalho do analista de sistemas

Um dos importantes aspectos do trabalho do analista de sistemas é o dimensionamento do equipamento eletrônico para dada empresa. Entretanto, admitindo-se que o computador já se encontre instalado, seu trabalho será o planejamento dos serviços, em termos gerais, sem entrar em detalhes.

Assim, cabe ao analista determinar:

- a) quais os serviços que podem ser executados no computador;
- b) quais as etapas de execução desses serviços;
- c) que configuração de equipamento será usada para cada etapa;
- d) quais os dados de entrada e os dados de saída de cada etapa.

O analista de sistemas deve conhecer o equipamento que irá executar o serviço, mas seu conhecimento não precisa entrar em detalhes de programação.

Por outro lado, o analista deverá conhecer muito bem as aplicações, ter experiência em racionalização do trabalho e cálculos de tempo. Hierarquicamente, o analista de sistemas está acima do programador.

O trabalho do programador

O programador recebe as linhas mestras da solução de um problema de processamento de dados e, a partir delas, deve preparar um programa que capacite o computador a resolver este problema.

O trabalho do programador tem 3 etapas:

- a) Primeiramente, estabelece a lógica da solução, isto é, a seqüência da operação, entrando nos mínimos detalhes e prevendo todos os casos possíveis, o que muitas vezes não é fácil. Esta solução é representada graficamente por meio de um conjunto de símbolos chamado fluxograma de programação ou, simplesmente, fluxograma.
- b) Em seguida, baseando-se no fluxograma, o programador escreve o PROGRAMA, ou seja, a sucessão de operações básicas que o computador deverá executar para obter os resultados desejados. Nos primeiros tempos da computação eletrônica, o programa era

escrito diretamente em linguagem de máquina, o que era enfadonho, cansativo e sujeito a inúmeros erros. Hoje em dia, o programa é escrito numa linguagem simbólica mais fácil de usar, cabendo ao próprio computador a tarefa de converter esse programa simbólico num programa em linguagem de máquina.

- c) Finalmente, o programa deverá ser testado para serem eliminados os erros da lógica e os erros formais. Esta fase do programa do programador recebe o nome de depuração do programa.

O operador e suas responsabilidades

O operador é a pessoa encarregada de executar os serviços, obedecendo a um plano preparado pela gerência do Centro. Introduz no computador os programas necessários e alimenta os dados de entrada, coloca os formulários nas impressoras, coloca os carretéis de fita nas unidades de fita, bem como retira os cartões processados, as fitas processadas, os impressos produzidos.

Cabe ao operador apertar os botões que acionam todas as partes do sistema. Além disso, em caso de parada do sistema, o operador deverá investigar as causas da parada e, se possível, tomar as providências que permitam o prosseguimento do trabalho.

Se a parada for motivada por falhas do equipamento ou condições anormais, o operador poderá identificar essas falhas por meio de luzes indicadoras localizadas no processador e nas unidades periféricas. Por outro lado, se a parada for decorrente de condições de programa ou de erro nos dados de entrada, ela poderá ser identificada por meio de mensagens impressas no console do computador. Essas mensagens podem ser geradas pelo próprio programa do usuário ou por rotinas do sistema operacional. A providência a ser tomada o operador a encontrará na pasta de operação e no manual de utilização do equipamento.

A responsabilidade do operador é muito grande, pois o custo de operação por hora do sistema é muito elevado, e qualquer descuido do operador pode inutilizar o serviço de várias horas. Por exemplo, num serviço de emissão de avisos de imposto, basta alimentar, de maneira errada, algumas constantes para que os avisos saiam todos errados, acarretando a perda não só das horas de trabalho do sistema, como também de milhares de formulários, cujo custo não é desprezível.

A manutenção

O serviço de manutenção fica, geralmente, a cargo do fabricante, sendo executado por técnicos de manutenção, sob a orientação de um engenheiro de manutenção. Somente centros muito grandes têm o seu próprio pessoal de manutenção.

Há dois tipos de manutenção: a manutenção preventiva e a reparação de defeitos. A manutenção preventiva é feita periodicamente, e visa detectar falhas em potencial antes que se manifestem.

Quando ocorre algum defeito de máquina, chama-se o técnico para sanar o defeito.

Serviços auxiliares

Em quase todas as instalações existe um setor de serviços auxiliares cuja principal função é a de preparar os dados de entrada; para isto, os dados dos documentos originais são transcritos em cartões perfurados, fitas perfuradas etc. Isso exige o trabalho de perfuradoras, verificadoras, operadores de equipamento convencional etc. Com a adoção progressiva da leitura ótica, esta parte da instalação tende a desaparecer.

O gerente do Centro de Processamento de Dados

Podemos dizer, sem receio de exagerar, que o gerente de um Centro de Processamento de Dados é a sua viga mestra: o sucesso das operações do Centro depende, em grande parte, da atuação do seu Gerente. Por esta razão, impõe-se grande cuidado na seleção deste elemento-chave.

Dentre as várias atribuições do gerente de um C.P.D. podemos citar as seguintes:

- 1) Coordenação dos serviços de análise e programação;
- 2) planejamento geral do serviço de processamento;
- 3) seleção do pessoal técnico do Centro; e
- 4) avaliação do desempenho dos elementos do Centro.

O gerente de um C.P.D. deve possuir uma série de qualidades que são difíceis de encontrar reunidas numa pessoa. Por um lado, deve-se fazer respeitar pelos subordinados, estimulando-os a se esforçarem e a se dedicarem ao Centro, o que requer boas relações humanas, para a criação de uma atmosfera agradável de trabalho.

Por outro lado, o gerente deve conquistar o respeito dos seus superiores e ser suficientemente combativo para defender os interesses do Centro perante a alta administração, que nem sempre compreende, de imediato, as necessidades do Centro e as aspirações de seus elementos. Uma das condições para ser prestigiado pelos subordinados e superiores é um bom nível de conhecimentos técnicos de processamento de dados. Por isso, é comum recrutar o gerente de um Centro entre os analistas de sistema.

Além disso, atualmente, a maioria das empresas exige que o gerente do seu C.P.D. tenha curso superior.



*Impresso nas Oficinas
da*

EDITORA "AVE MARIA" LTDA.

*C.E.P. 01226 — Rua Martim Francisco, 656
Vila Buarque*

Tel.: 826-6111 — Inscr. Mun. 0.221.653-1

*Inscr. Est. 101.067.554 — CGC 60.494.200/0001-70
São Paulo*

processamento de dados; a entrada, saída e armazenamento de informações no computador; sistemas de numeração; o conceito de programa, instrução e linguagens de programação; os principais recursos de software; o computador dentro da empresa.

NOTA SOBRE O AUTOR

Ernesto Mário Haberkorn é graduado pela Escola de Administração de São Paulo, da Fundação Getúlio Vargas. Dentro da área de processamento de dados, especializou-se na programação de diversas linguagens de computadores, análises de sistemas, sistemas operacionais e aplicações de teleprocessamento, tendo feito estágios nos Estados Unidos e na Europa. É professor de Análise de Sistemas e autor do livro **INTRODUÇÃO À ANÁLISE DE SISTEMAS**, pertencente ao fundo editorial da Atlas.

APLICAÇÃO

- * Livro de texto para cursos básicos de Processamento de Dados.
- * Leitura complementar para as disciplinas: Organização e Métodos, Administração de Materiais, Administração de Operações, Administração de Materiais, Pesquisa Operacional e Informática.
- * Leitura recomendada para: administradores de empresas, técnicos de contabilidade e produção, especialistas em Organização e Métodos e especialistas em Processamento de Dados.

PROCESSAMENTO DE DADOS

COMPUTAÇÃO BÁSICA E PROGRAMAÇÃO

Ruy J. A. Campos

FUNDAMENTOS DE PROCESSAMENTO DE DADOS

Dayr Ramos Américo dos Reis

INTRODUÇÃO À ANÁLISE DE SISTEMAS

Ernesto M. Haberkorn

INTRODUÇÃO AO PROCESSAMENTO DE DADOS

W. A. Bocchino

PROCESSAMENTO DE DADOS NAS EMPRESAS

Elias M. Awad

PROGRAMAÇÃO COBOL

Daniel D. McCracken e U. Garbassi

PROGRAMAÇÃO PL/1 — MANUAL DE
INSTRUÇÃO PROGRAMADA

Edna H. Barbour

SISTEMAS DE INFORMAÇÃO NA ADMINISTRAÇÃO
DE EMPRESAS

A. L. Cautela e E. G. Franco Polloni

PUBLICAÇÕES DA EDITORA ATLAS S.A.