

Um Computador Simplificado

Material disponível em: <http://homepages.dcc.ufmg.br/~rodolfo/aedsi-2-07/ComputadorSimplificado/computador-simplificado.html>

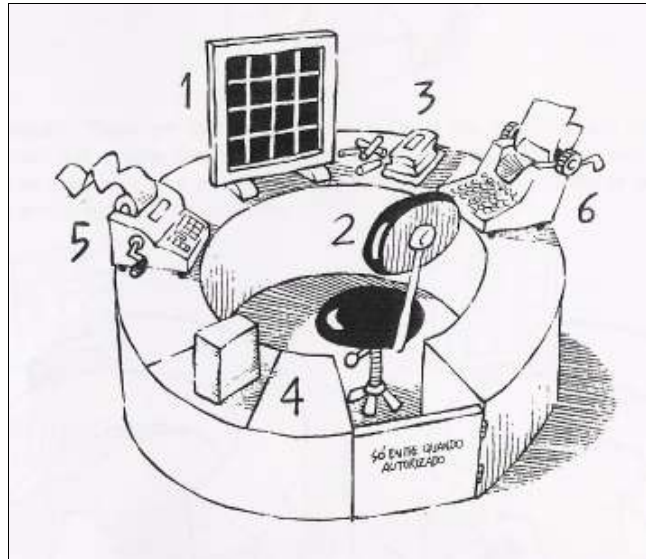


figura 1 - O computador simplificado

Um computador envolve um grande número de elementos cada um envolvendo um grande número de conceitos. Para introduzir alguns destes elementos e alguns dos conceitos definimos o *computador simplificado*. O Computador simplificado utiliza elementos do dia a dia como metáfora para alguns dos elementos existentes em um computador. Considere os elementos da *figura 1*:

- (1) Um conjunto de escaninhos (cada espaço é um escaninho). Em um escaninho existem valores binários que podem ser interpretados como um valor inteiro ou uma instrução. A escrita ou colocação de instruções nos escaninhos só pode ser feita uma única vez antes da pessoa iniciar seu trabalho.
- (2) Uma pessoa cujo trabalho consiste em pegar instruções que estão em um quadro-negro subdividido em vários espaços e executar as instruções conforme detalhado à frente. Dizemos que o quadro-negro subdividido em vários espaços corresponde a memória.
- (3) Um apagador e um giz usado para apagar e escrever comandos e valores nos escaninhos.
- (4) Cartões, os quais contêm um valor por cartão.
- (5) Uma calculadora para realizar somas, subtrações, etc.
- (6) Uma máquina de escrever para imprimir valores.

Existem duas maneiras de fornecer valores a esta pessoa: (i) um escaninho (subdivisão do quadro-negro) pode conter um valor, este valor pode ser interpretado como uma instrução ou pode ser interpretado como um valor inteiro (uma constante) ou então (ii) podemos utilizar os cartões.

A pessoa quando inicia seu trabalho deve seguir o seguinte procedimento:

Passo 0: Escreva em uma folha de papel a identificação do primeiro escaninho (E0);

Passo 1: Pegue a instrução que está no escaninho indicado pela folha de papel

Passo 2: Escreva na folha de papel a identificação do escaninho seguinte;

Passo 3: Faça o que manda a instrução;

Passo 4: Volte para o Passo 1.

Um computador tem um componente denominado *Unidade Central de Processamento (UCP)* ou *processador* que trabalha de forma similar à pessoa descrita acima. O processador pode contar com vários recursos entre eles, por exemplo, a folha de papel onde anota qual é o escaninho que contém a próxima instrução a ser executada. A folha de papel, mencionada nos passos acima, em um computador de verdade, corresponde a um componente do processador denominado *registrador contador de programa* ou simplesmente *contador de programa*. O processador conta com a ajuda de circuitos que realizam operações de lógica e operações aritméticas, a calculadora acima corresponderia a aos circuitos de uma

Unidade Lógico-Aritmética ou *ULA*. Um componente chamado *memória principal* funciona de forma similar ao conjunto de escaninhos (ou subdivisões de um quadro-negro). Existem diversos tipos de memória para um computador. A memória *Read Only Memory* (p.ex. ROM, PROM, EPROM, EEPROM) seria equivalente a um conjunto de escaninhos onde a pessoa ou processador só pode ler sem poder escrever (o quadro-negro estaria escrito com tinta que não pode ser apagada). Os cartões correspondem a um dispositivo de entrada ou seja um dispositivo que fornece valores à pessoa ou processador. A máquina de escrever corresponde a um dispositivo de saída ou seja um dispositivo através do qual a pessoa ou processador pode interagir com o mundo externo.

Um processador pode ser representado pelo conjunto de instruções (ações, operações) que ele é capaz de executar. As instruções devem especificar (i) O QUE deve ser feito (operação) e (ii) SOBRE QUAIS objetos (operandos). As instruções do computador simplificado são as seguintes:

- **leia cartão e guarde em Ei** - O conteúdo da posição especificada (Ei) deve passar a ser o mesmo valor do cartão que estiver disponível; o cartão deve ser descartado de forma que a próxima instrução de leitura de cartão deverá utilizar o próximo cartão.
- **copie Ei em Ej** - O conteúdo da posição Ej deve passar a ser o mesmo da posição Ei;
- **some(subtraia, multiplique,...) Ei e Ej e guarde em Ek** - O conteúdo da posição Ek deve passar a ser a soma(subtração, etc.) do conteúdo das posições Ei e Ej;
- **vá para Ei** - O conteúdo da folha de papel deve passar a ser Ei;
- **se Ei (maior, menor, maior ou igual,...) Ej vá para Ek** - Se o conteúdo da posição Ei é maior(menor etc) que o da posição Ej então escrever na folha de papel Ek, caso contrário deixe a folha como está.
- **imprima Ei** - Datilografe o conteúdo da posição Ei.
- **pare** - Pare de executar o procedimento que especifica um ciclo envolvendo os passos 1, 2, 3 e 4;

Podemos verificar que a instrução **leia cartão** tem como operando uma posição de memória, onde deverá ser guardado o valor lido em cartão. A instrução **copie** tem um operando fonte e um operando destino; o operando fonte fornece o valor a ser copiado e o operando destino perde o seu valor anterior passando a ter o mesmo valor da posição de memória fonte. A instrução de desvio condicional tem três operandos: os valores das posições de memória de dois dos operandos são comparados e, caso a comparação seja satisfeita, o terceiro operando é usado como origem de novas instruções. A instrução **pare** não tem operandos.

Um **algoritmo** é uma receita [ou processo ou método ou técnica ou procedimento ou rotina] que permite a um agente resolver um problema. Um algoritmo se constitui de uma seqüência de passos deve terminar depois de especificados um número finito de passos. Cada passo de um algoritmo deve ser definido de maneira precisa envolvendo ações que podem ser cumpridas de maneira rigorosa. Um algoritmo pode ter nenhuma, uma ou mais **entradas**, isto é valores que estão disponíveis conforme as especificações dos passos do algoritmo. Um algoritmo pode ter uma ou mais **saídas**, isto é valores que são produzidos através da *execução* do algoritmo. Cada um dos passos de um algoritmo deve poder ser executado de maneira exata e em um tempo finito.

Algoritmos podem ser representados de diversas formas, uma das possíveis representações de algoritmos é o programa de computador, o agente neste caso é um computador.

Um **programa** consiste em uma seqüência de instruções e constantes que podem ser armazenadas na memória do computador para poder atingir um determinado fim. Um programa tem um aspecto **estático** correspondente à seqüência de instruções e constantes e um aspecto **dinâmico** correspondente à execução das instruções conforme descrito nos passos de 0 a 4 acima.

O aspecto dinâmico de um programa depende não só das instruções mas também dos valores que o programa obtém do mundo externo através de instruções tal como a instrução **leia cartão e guarde em Ei**. Os valores lidos pelo programa são denominados **Entrada**. Muitas vezes os programas podem interferir no mundo exterior através de instruções especiais. Um exemplo de tal tipo de instrução é: **imprima Ei**. Instruções deste tipo produzem modificações no mundo externo ao computador e são denominadas instruções de **Saída**.

Abaixo são mostrados alguns exemplos de programas. O conceito de **arquivo** é muito importante em um computador. Este conceito não existe no Computador Simplificado. Em um

computador existe um mecanismo que irá fazer a **carga** de um programa na memória do computador a partir de um arquivo. No Computador Simplificado iremos assumir que as instruções e os dados necessários para a execução de um programa "aparecem" de forma conveniente na memória.

Na memória de um computador as instruções e os dados são todos representados através de zeros e uns. Para simplificar a apresentação iremos abstrair da representação de zeros e uns e iremos representar os programas simbolicamente. Iremos utilizar símbolos mnemônicos. Em outro texto da disciplina iremos discutir como que podemos representar instruções e dados na forma de zeros e uns.

Uma possível sequência para desenvolver um programa:

- Os requisitos são levantados
- Uma solução é investigada e descrita
- Um programa correspondente à solução é codificado em uma linguagem de programação;
- O programa é testado e modificado até que seja considerado correto.
-

Após os passos acima o programa fica disponível para uso. Normalmente os programas representados na forma de zeros e uns são armazenados utilizando algum tipo de dispositivo de armazenamento (por exemplo, Discos rígidos, CDs).

Sempre que quisermos usar um programa o primeiro passo é *carregar* as instruções e dados do programa. Carregar o programa consiste em copiar as instruções e dados de algum dispositivo de armazenamento tal como Disco rígido ou CD para a memória do computador. O processador só executa instruções que estejam na memória do computador. Se as instruções estiverem em algum outro dispositivo de armazenamento é preciso que elas sejam primeiro transferidas para a memória. No Computador Simplificado não é definido o mecanismo de carga. A descrição do programa assume que ele esteja na memória através de um mecanismo não definido.

O programa abaixo lê dois valores e imprime o maior deles:

E0: leia cartão e guarde em E15	E1: leia cartão e guarde em E14	E2: se E15 maior que E14 vá para E5	E3: imprima E14
E4: vá para E6	E5: imprima E15	E6: pare	E7: ?
E8: ?	E9: ?	E10: ?	E11: ?
E12: ?	E13: ?	E14: ?	E15: ?

Outra forma de representar o programa acima:

```

E0: leia cartão e guarde em E15
E1: leia cartão e guarde em E14
E2: se E15 maior que E14 vá para E5
E3: imprima E14
E4: vá para E6
E5: imprima E15
E6: pare

```

O aspecto dinâmico de um programa pode ser descrito pelo *fluxo de controle*. O fluxo de controle corresponde à sequência de instruções que são executadas pelo processador. Conforme dito acima o computador simplificado sempre inicia a execução de um programa a partir da posição E0, portanto o fluxo de controle definido pela execução de um programa no computador simplificado sempre começa com E0. A sequência inicial do fluxo de controle do programa acima corresponde a E0, E1, E2; E2 corresponde a um desvio condicional e neste ponto o fluxo depende da **Entrada** do programa. Vamos assumir que o programa tenha lido 4 na instrução correspondente a E0 e tenha lido 3 na instrução correspondente a E1. Neste caso o fluxo de controle de E2 passa para E5, e termina em E6. Ou seja, neste caso o fluxo de controle corresponde a E0, E1, E2, E5, E6. Se considerarmos que os valores lidos foram 3 em E0 e 4 em E1, o fluxo de controle de E2 passa para E3 e daí para E4 e E6. Neste caso o fluxo de controle corresponde a E0, E1, E2, E3, E4, E6.

O programa abaixo apresenta uma outra solução para o problema de ler dois valores e imprimir o maior deles:

E0: leia cartão e guarde em E15	E1: leia cartão e guarde em E14	E2: se E15 maior que E14 vá para E5	E3: copie E14 em E13
E4: vá para E6	E5: copie E15 em E13	E6: imprima E13	E7: pare
E8: ?	E9: ?	E10: ?	E11: ?
E12: ?	E13: ?	E14: ?	E15: ?

Suponha que a instrução em E0 leia o valor 4 e a instrução em E1 leia o valor 3. Neste caso o fluxo de controle corresponde a: E0, E1, E2, E5,E6, E7. Se a instrução em E0 ler o valor 3 e a instrução em E1 ler o valor 4 então o fluxo de controle será: E0, E1, E2, E3, E4, E6,E7.

O programa abaixo lê valores em cartões até que apareça o valor 0 ou 1 em algum cartão. Observe que neste programa existem escaninhos com instruções e escaninhos com valores inteiros (E14 e E15).

E0: leia cartão e guarde em E13	E1: se E13 igual a E15 vá para E4	E2: se E13 igual a E14 vá para E4	E3: vá para E0
E4: pare	E5: ?	E6: ?	E7: ?
E8: ?	E9: ?	E10: ?	E11: ?
E12: ?	E13: ?	E14: 0	E15: 1

Outra forma de representação:

E0: leia cartão e guarde em E13
E1: se E13 igual a E15 vá para E4
E2: se E13 igual a E14 vá para E4
E3: vá para E0
E4: pare
E14: 0
E15: 1

O fluxo de controle no programa acima pode ser tornar tão longo quanto se queira, basta colocar um número adequado de cartões com valores diferentes de 0 e 1. Considere que existam 5 cartões com valor 3 e o sexto cartão tenha valor 1. O fluxo de controle será: E0, E1,E2,E3,E0, E1,E2,E3,E0, E1,E2,E3,E0, E1,E2,E3,E0, E1,E2,E3,E0,E1,E4.

Para pensar:

Todo algoritmo é um programa?

Todo programa é um algoritmo?

O que deve ser feito se a instrução a ser executada é a instrução **leia cartão e guarde em Ei** e não existe um cartão disponível? (sincronismo vs. assincronismo!)

Considere o trecho de programa:

E0: leia cartão e guarde em E15
e considere que seja lido o valor 5. Considere o trecho de programa:
E0: copie E14 em E15
...
E14: 5

Ambos os trechos fazem a atribuição da constante 5 para a posição E15. Explique a principal diferença entre utilizar uma constante como sendo um valor em uma posição de memória e uma constante

como sendo um valor escrito em um cartão (responda em termos do *tempo de vida* de um programa). Observe que as instruções do computador simplificado não permitem expressar diretamente um valor constante, ou seja, **não** existe uma instrução do tipo *copie 5 em Ej*; em *tempo de carga* do programa teremos que colocar o valor 5 em Ei e utilizar "Copie Ei em Ej".

Um processador possui alguns componentes denominados registradores. Os registradores são similares aos escaninhos. O processador pode escrever ou ler valores associados aos registradores. A "folha de papel" mencionada acima, que contém o endereço do escaninho cujo conteúdo corresponde à próxima instrução a ser executada, corresponde, metaforicamente falando, a um registrador denominado "Contador de programa". Considerando um processador genérico, qual é a influência do número de bits do contador de programa no tamanho dos programas que podem ser executados?

Para pensar:

A instrução subtraia de Ei Ej e guarde em Ek corresponde a guardar em Ek o valor de Ei-Ej; Qual o efeito da instrução subtraia de E15 E15 e guarde em E15 ? Qual é o efeito da instrução some Ei e Ei e guarde em Ei?