

# **PCS 3216**

# **Sistemas de Programação**

João José Neto

Aula 03 – Loaders e dumpers. Preparação  
de programas para a execução

- Um expediente primitivo, historicamente utilizado com essa finalidade, consistiu na inclusão, no hardware da máquina, de extensões auxiliares, que disponibilizassem aos operadores da máquina uma via de acesso mais confortável.
- Através dessas adições, programas e dados puderam passar a ser introduzidos na memória do computador, em seu formato binário (na forma de cadeias de zero e uns) mediante o acionamento de botões e de chaves binárias (“on” - “off”) dispostas em um painel de controle.

# Entrada de dados via painel de chaves



Mostradores de LEDs

Lâmpadas sinalizadoras

Chaves binárias

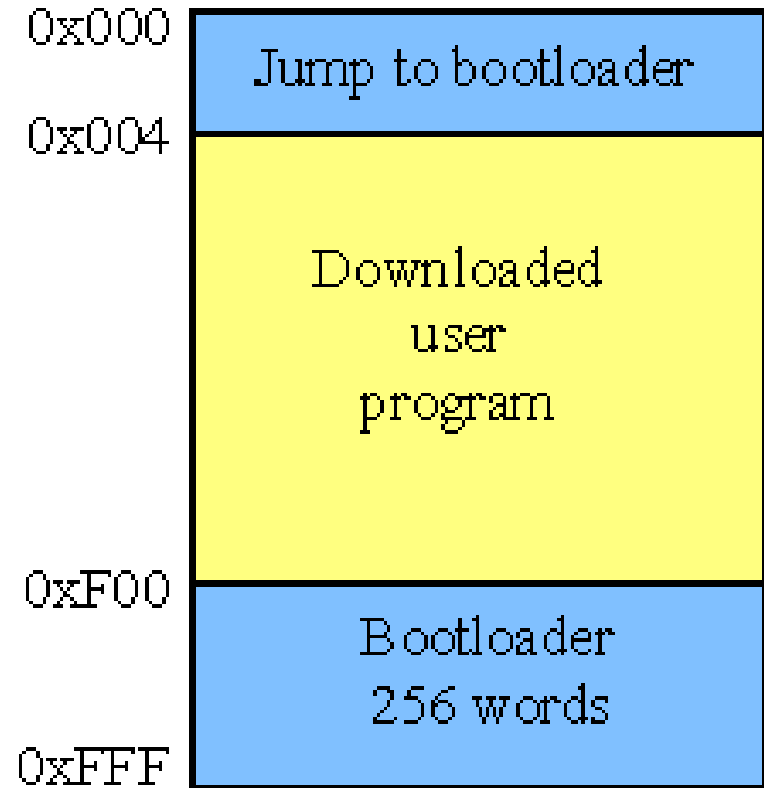
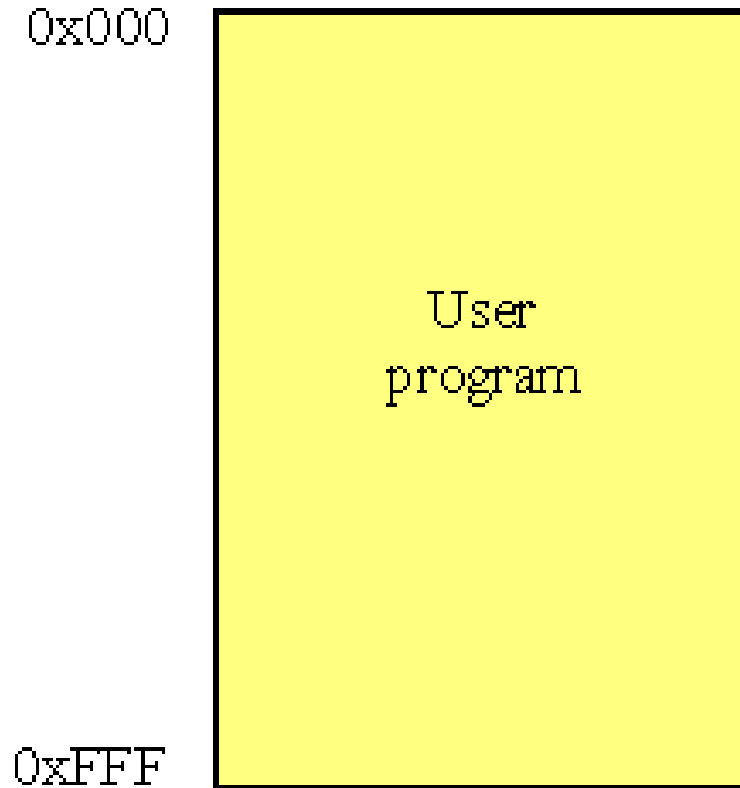
Botões de acionamento

Painel de chaves

# Loader Absoluto (bootloader)

**No  
bootloader**

**Bootloader  
installed**



# Entrada de dados via teclado hexadecimal



- Dispondo-se, dessa forma, de algum dispositivo de entrada de dados, como, por exemplo, um teclado, torna-se possível, com seu auxílio, introduzir na máquina dados ou instruções, não mais codificados em formato binário, mas em alguma forma mais compacta e portanto menos sujeita a erros, como proporcionam, por exemplo, as notações numéricas usuais: octal, decimal e hexadecimal.

# Loader binário

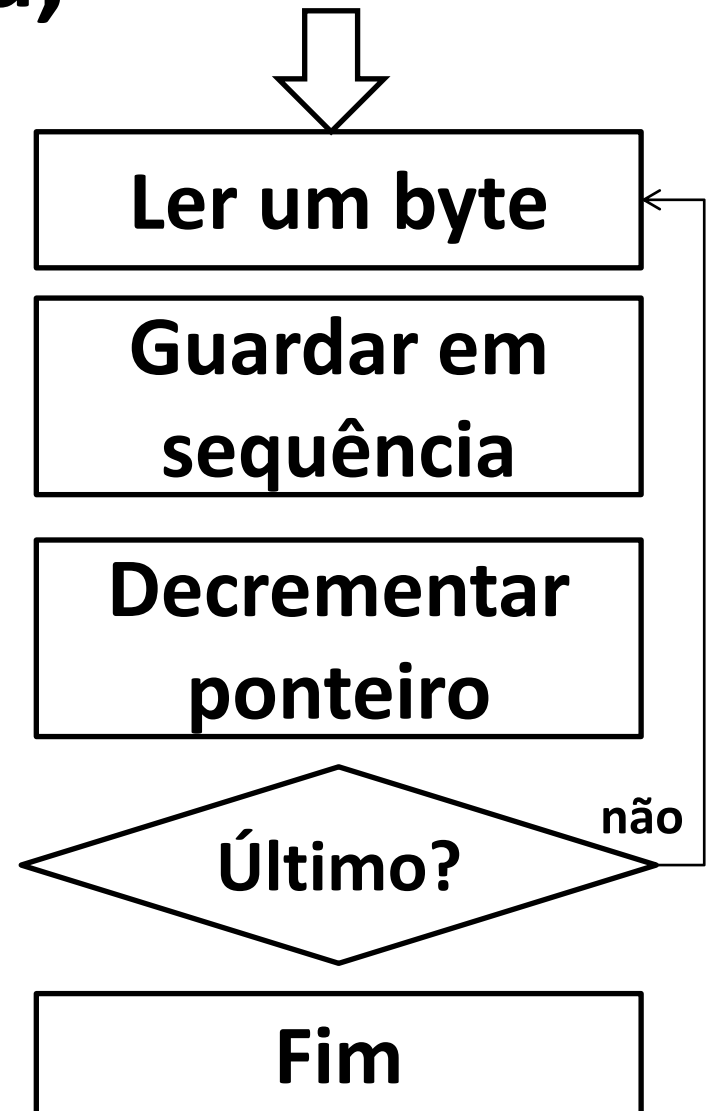
Para isto, mostra-se de grande ajuda um pequeno programa, encarregado da leitura dos dados no dispositivo de entrada disponível, para que os dados lidos sejam armazenados nas posições adequadas de memória do computador.

```

                                ORG      0
000CE 16      .      FNC      /E6      LÊ UM BYTE DA FITA
002CE 21      .      SAL      /E1      "
00400 02      .      PLA      *-2      "
006CE 40      .      ENTR     /E0      "
00820 20      .      ARM      /020     ARMAZENA
00AE0 09      .      SUS      .-+1     ATUALIZA PONTEIRO E TESTA SE ACABOU
00C00 00      .      PLA      .-2     NÃO, CONTINUA
*
*      SIM: RODA LOADER, QUE FOI CARREGADO
*      DESDE A POSIÇÃO /00E ATÉ /080
*      ESTE, POR SUA VEZ, CARREGA O CARRE-
*      GADOR ABSOLUTO DA POSIÇÃO /F80.
*      SE TUDO OK, PÁRA COM 0 NO ACUMULADOR
*      PROTEGER A MEMÓRIA E DAR NOVA
*      PARTIDA PARA RODAR O CARREGADOR
*      ABSOLUTO A PARTIR DA POSIÇÃO /F80.
```

# Ou seja,

A lógica ao lado permite construir um programa loader binário muito simples com o qual é possível carregar um número fixo de bytes para uma sequência fixa de posições de memória.

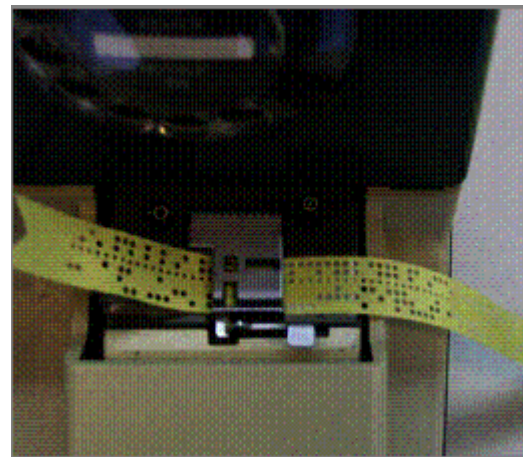
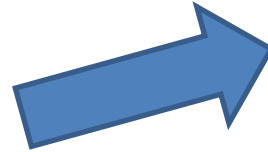




# Carga de um programa



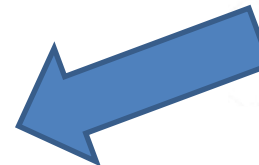
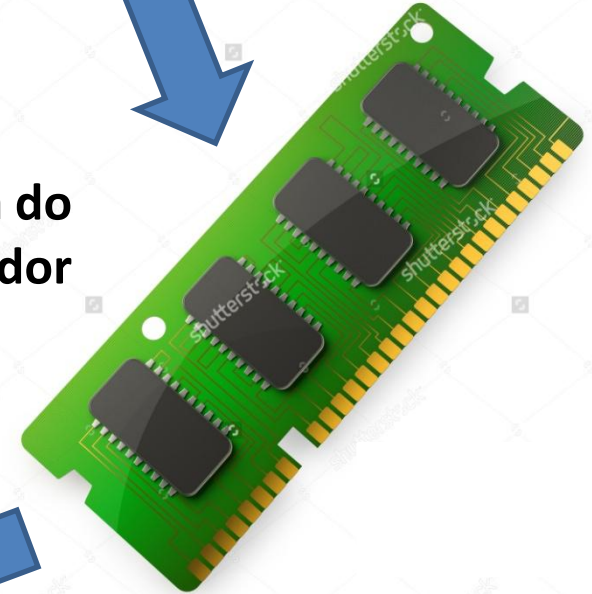
Fita contendo a Imagem binária do programa executável



Leitora de fita perfurada

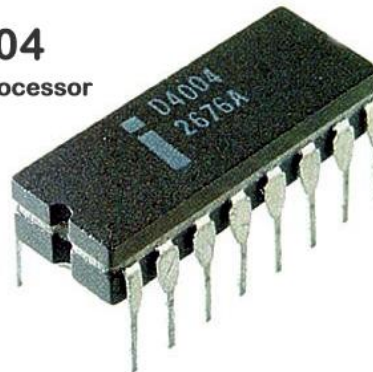


Memória do computador

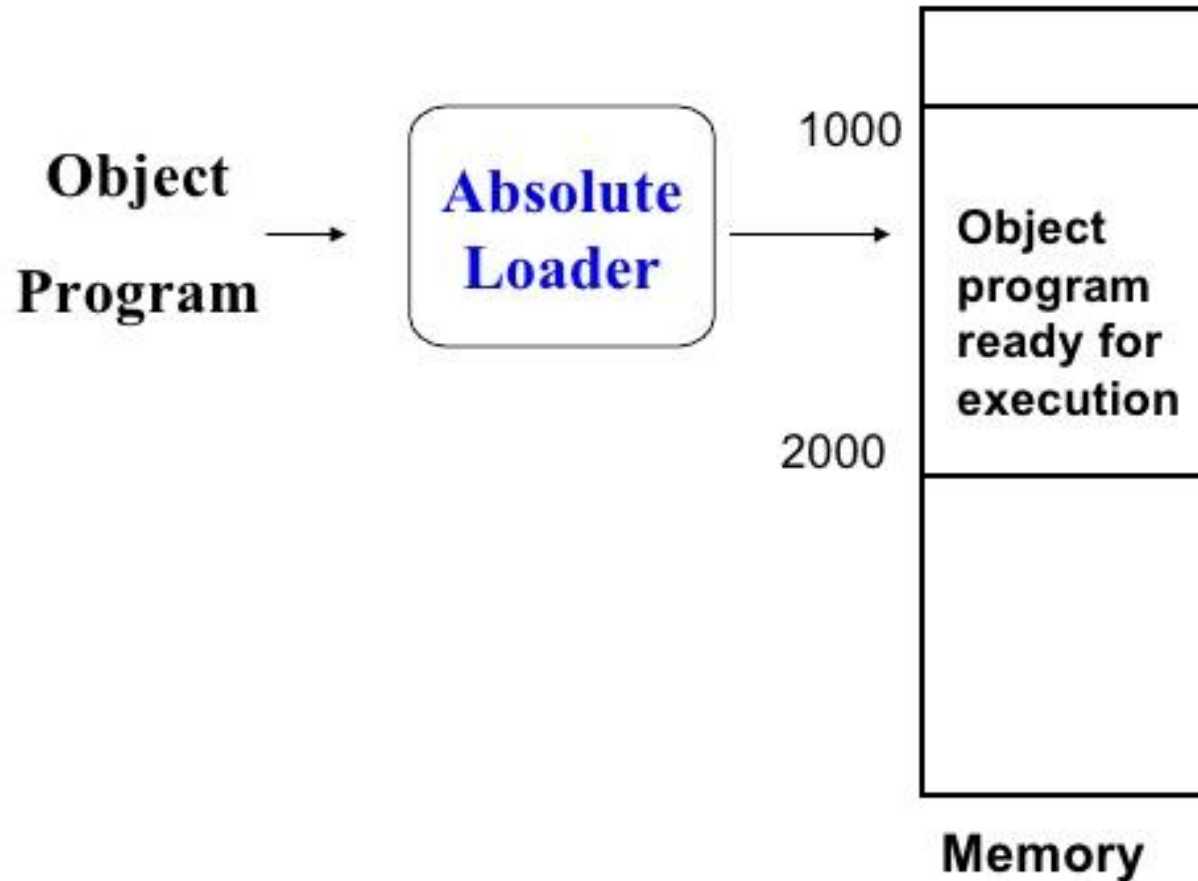


INTEL 4004  
The first microprocessor

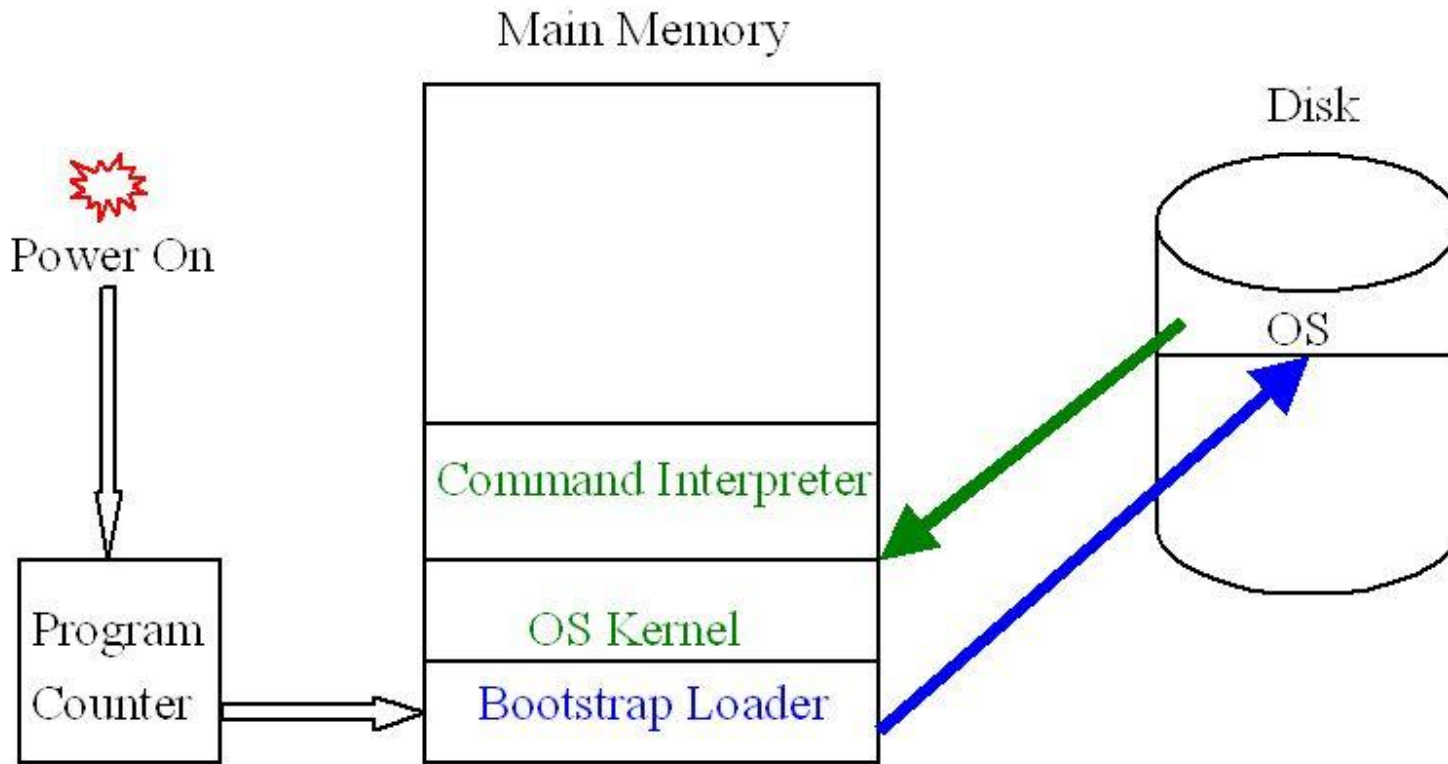
Processador



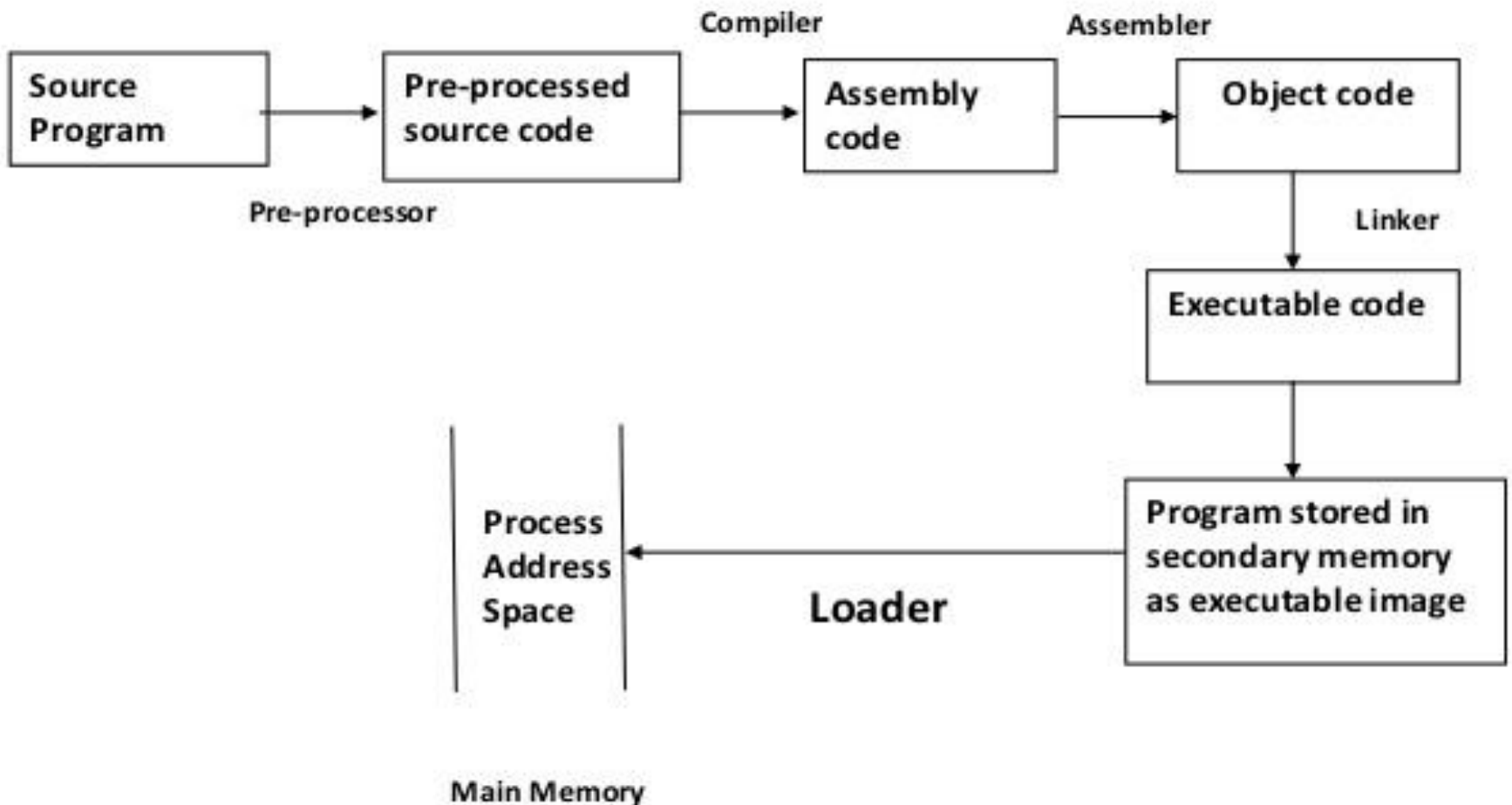
# Função do carregador absoluto



# Carregador absoluto



# Processo de construção de programas



- Surge conceitualmente, desta maneira, um primeiro programa de sistema, conhecido como carregador de códigos binários (representando programas e/ou dados) para a memória da máquina (também denominado "loader" ou "bootstrap" ou, abreviadamente, "boot").
- Portanto, um carregador de códigos binários, em sua forma mais rudimentar, efetua pura e simplesmente a leitura de uma sequência de números binários (eventualmente codificados em alguma base não binária de numeração), interpretando-a como sendo a imagem dos bits a serem armazenados na memória da máquina.

# Carregador binário

- Surge desta maneira, um primeiro programa conceitual de sistema: o carregador.
- Em sua forma mais rudimentar, o carregador (ou “loader”, “bootstrap”, “boot”) executa duas operações:
  - Leitura de códigos binários (de programas e/ou de dados)
  - Deposita esses códigos nas posições convenientes de memória, ficando assim prontos para execução

# Preparação prévia

- Obviamente, para que seja possível operar o carregador, é necessário que previamente o código de máquina do próprio carregador tenha sido, por sua vez, introduzido corretamente na memória da máquina, e que tenham sido preparados adequadamente, com antecedência, os dados correspondentes ao código binário do programa a ser carregado.
- A execução do carregador exige, assim:
  - O código de máquina do próprio carregador deve estar presente na memória da máquina
  - Em meio externo, deve estar disponível o código de máquina do programa a ser depositado na memória pelo carregador

# A carga inicial deve ser manual

- Dessa maneira, sendo o carregador ele próprio um programa, e considerando que quase certamente nessa ocasião não haveria presente na memória do computador nenhum outro programa executável disponível, efetuar o seu armazenamento inicial na memória da máquina (para que possa ser executado) deverá, naturalmente, ser uma tarefa executada manualmente através das chaves do painel, de forma análoga à carga de outro programa qualquer fornecido em linguagem binária de máquina:
  - Ao menos nas primeiras vezes que o computador é utilizado, costuma ser necessário efetuar manualmente a carga inicial do loader.
  - Um painel de chaves ou algum outro veículo de entrada de dados deve estar disponível para essa operação.



# **Agora outros programas podem ser carregados automaticamente**

- Dessa maneira, sendo o carregador ele próprio um programa, e considerando que quase certamente nessa ocasião não haveria presente na memória do computador nenhum outro programa executável disponível, efetuar o seu armazenamento inicial na memória da máquina (para que possa ser executado) deverá, naturalmente, ser uma tarefa executada manualmente através das chaves do painel, de forma análoga à carga de outro programa qualquer fornecido em linguagem binária de máquina.

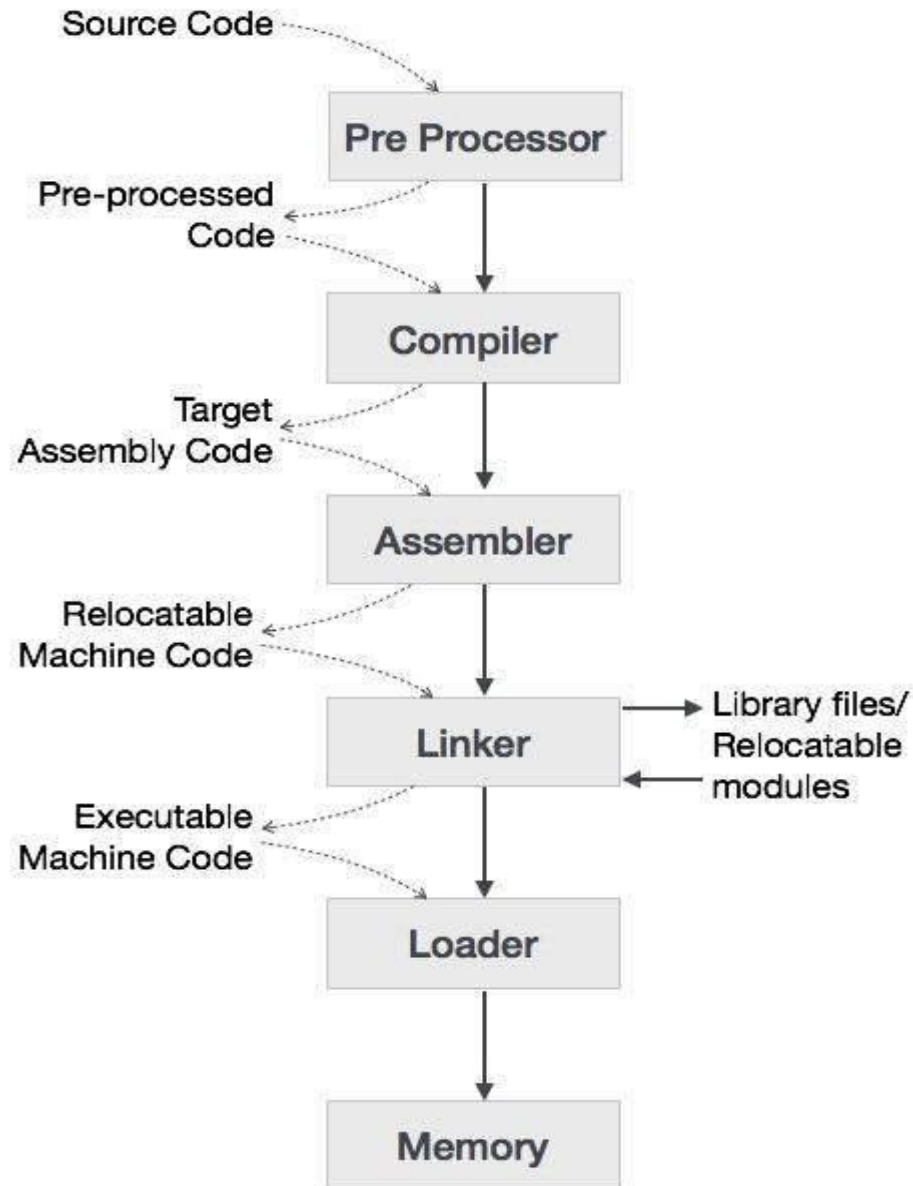
- Uma vez na memória, esse programa poderá ser utilizado para carregar outros programas (para evitar esse trabalho adicional e repetitivo, os programas correspondentes a esses carregadores binários costumam ser pré-gravados pelo fabricante do computador em memórias com acesso exclusivo para leitura).
  - A instalação do loader em uma memória pré-gravada, de leitura apenas, evita a repetição dessa tarefa toda vez que a máquina é ligada.
  - Para isso é necessário um hardware especial contendo a eletrônica adequada.
  - Subsequentemente, outros programas podem ser carregados pelo loader, para execução.

- A preparação de dados para serem usados pelo carregador binário pode ser feita de duas maneiras:
  - manualmente, através da digitação dos códigos de máquina binários diretamente no meio de armazenamento a ser utilizado no dispositivo de entrada disponível, ou
  - automaticamente, através do uso de um outro programa de sistema, o gerador de dados em formato carregável, que se incumba de, sem a intervenção do usuário, gerar os dados, a ele fornecidos, em formato adequado ao uso pelo carregador.

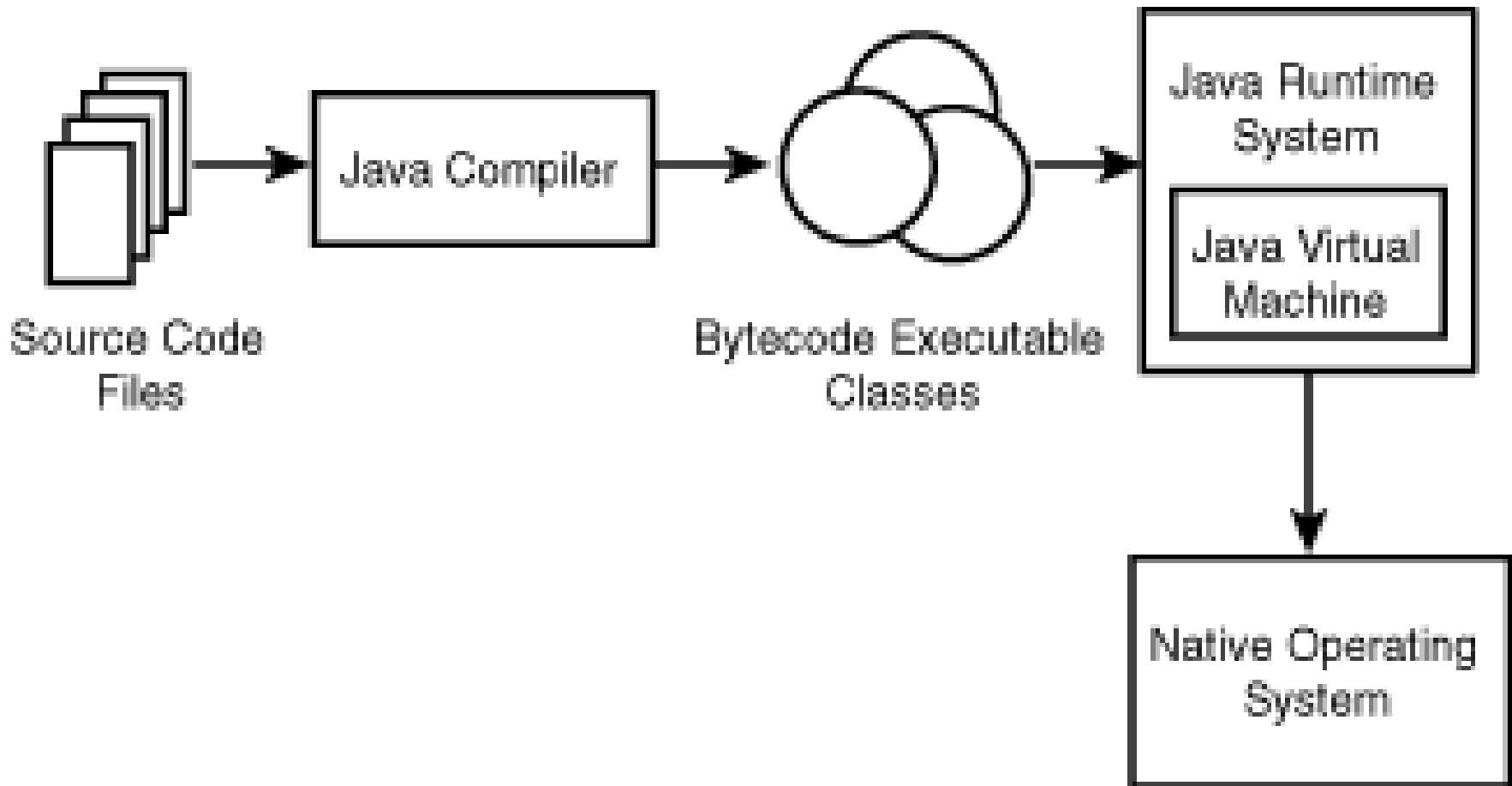
# Preparação manual ou automática de outros programas carregáveis

- O loader determina o formato dos dados a serem carregados na memória do computador.
- Para cada programa do usuário, é necessário preparar sua imagem de memória nesse formato.
- Isso pode ser feito codificando-o em linguagem binária de máquina e carregando-o através do dispositivo de entrada disponível.
- Pode também ser automatizado, usando outro programa que constrói o código do programa em meio externo, já no formato carregável.

# Preparação de programas para execução



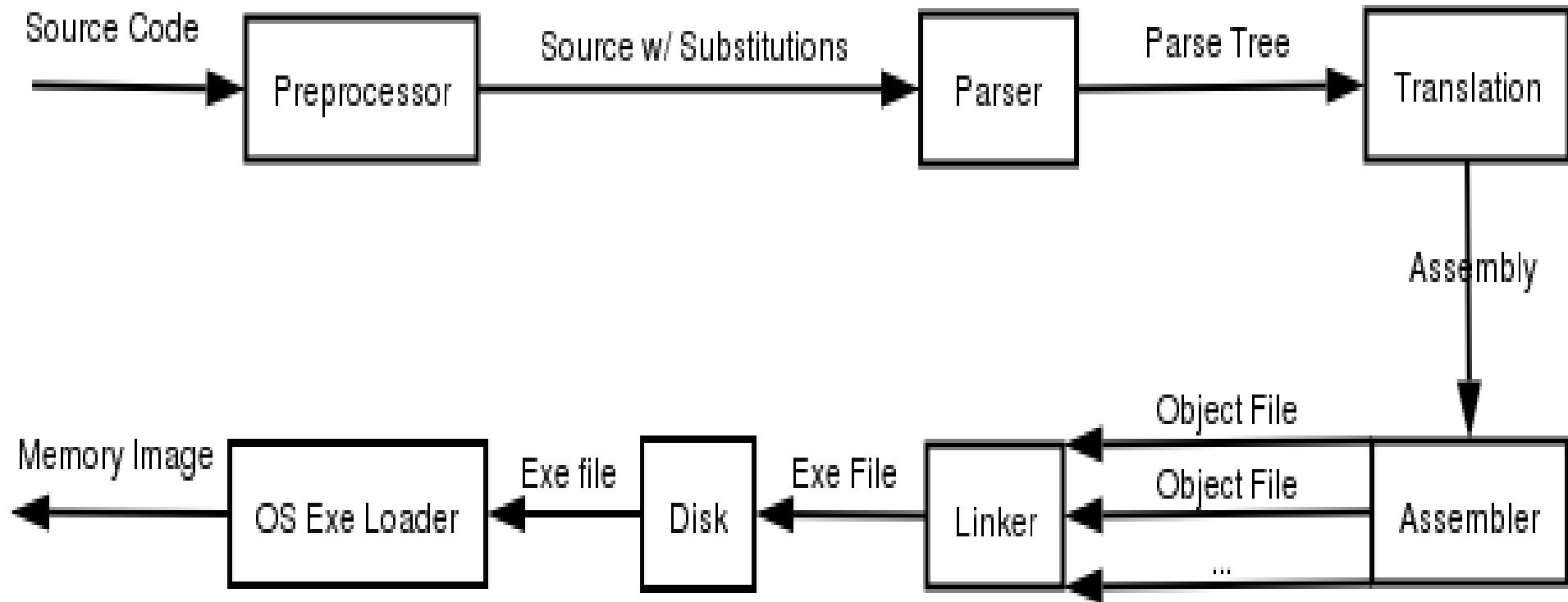
# Preparação de um programa para execução em máquina virtual



# **Bibliotecas de programas prontos para a execução**

- O surgimento desse novo conceito imediatamente suscita a ideia da criação de bibliotecas de programas, coleções de programas prontos, todos em formato carregável, para que possam ser confortavelmente reutilizados sem modificação, sempre que necessário.
- Programas já testados podem ser estocados e organizados em bibliotecas, em formato carregável, ficando disponíveis para reuso imediato.

# O processo de preparação de um programa de biblioteca





- De grande valia na elaboração de tais bibliotecas são os programas de sistema denominados preparadores de programas à imagem da memória, que os transferem da memória do computador, em que estiverem carregados, para um dispositivo de saída, preparando-os assim para reuso futuro.
- Com isso, o programador tem a possibilidade de fornecer à máquina os códigos binários correspondentes às instruções e dados que compõem o programa, a partir dos quais os preparadores de programa podem produzir, em meio externo de armazenamento e em formato carregável, um código binário equivalente que represente a imagem de memória a partir dos códigos binários fornecidos.

# Preparadores de programas à imagem da memória

- Transferem para meio externo, em formato carregável, o conteúdo da memória do computador, para que tal conteúdo seja recarregado em outra ocasião.

- Alternativamente, é possível, por meio de outra classe de programas de sistema denominados descarregadores de memória (“dumpers”) extrair diretamente da memória principal os códigos nela contidos, criando, a partir destes, uma imagem carregável (“dump”).
- De posse dessas ferramentas, o programador já pode estar liberado da trabalhosa tarefa de introduzir manualmente no computador os códigos binários dos seus dados e programas na máquina, todas as vezes que for executar um programa.
- No entanto, na ausência de outras ferramentas, as atividades de codificação, de depuração e de execução do programa ainda devem ser efetuadas integralmente sobre os códigos em linguagem binária de máquina.

# Dumpers binários em formato carregável

- Descarregadores de memória (*dumpers*) criam em meio externo imagens carregáveis do conteúdo da memória.
- Isto dispensa a carga manual dos programas.

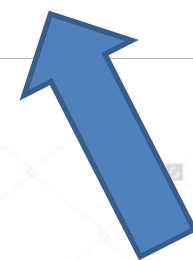
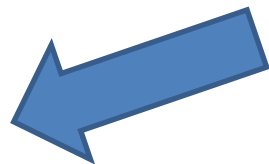
# Produção de uma fita carregável



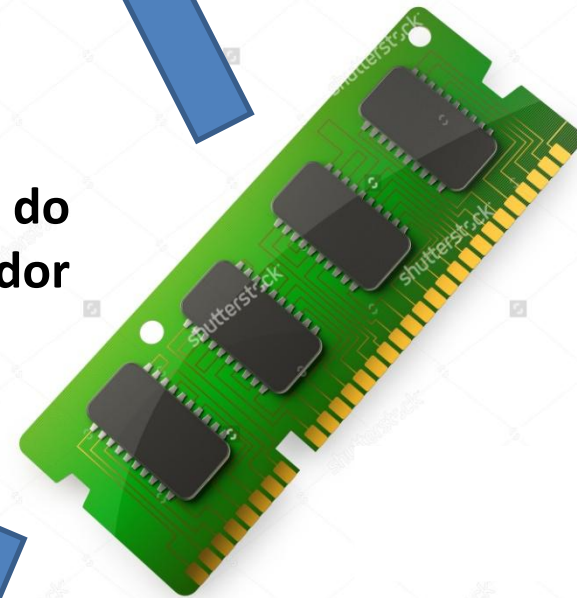
Fita contendo a Imagem binária do programa executável



Perfuradora de fita

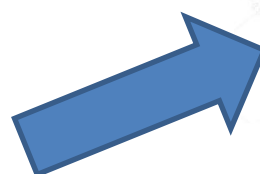
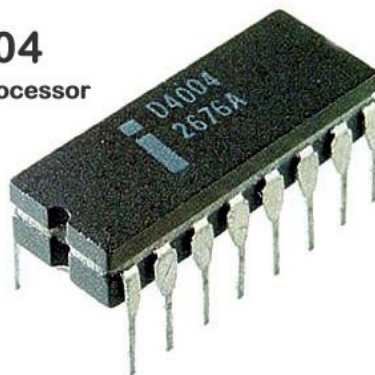


Memória do computador



INTEL 4004  
The first microprocessor

Processador



- A substituição dos códigos binários por outros equivalentes, denotados em alguma forma mais compacta, como é o caso da notação hexadecimal ou octal, em nada altera conceitualmente o nível de abstração com que é efetuada a programação, embora garantidamente contribua, de modo significativo, para aumentar a compreensão dos códigos, a facilidade de programação, a segurança, a confiabilidade e a rapidez de inserção dos códigos na máquina.
- É trivial implantar essa inovação, bastando para isso que sejam incorporadas ao sistema novas versões dos carregadores e preparadores de programas, que façam uso da notação escolhida em lugar da notação binária originalmente adotada.

# Códigos não binários

- São mais compactos e fáceis de ler:
  - Octal
  - Decimal
  - Hexadecimal
- Seu uso aumenta
  - a legibilidade dos códigos
  - a facilidade de programação
  - a segurança
  - a confiabilidade
  - a rapidez de inserção dos códigos na máquina

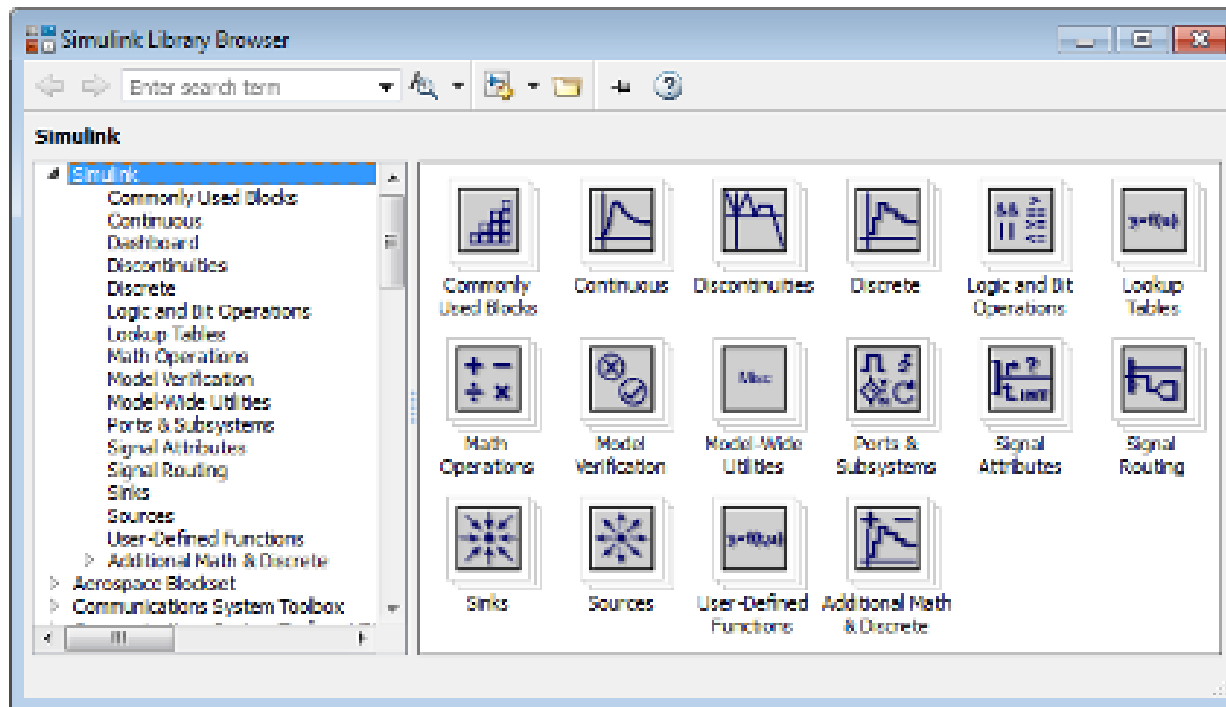
# Entrada/Saída em lugar de chaves/luzes do painel

- Note-se que, nessa fase da evolução dos sistemas computacionais, para fins de carga e descarga de programas entre a memória do computador e o meio externo, é intenso o uso de equipamentos de entrada e saída, que agora atuam em substituição às chaves binárias do painel de controle da máquina.
- Com a introdução de vias alternativas para a carga e descarga de código binário, passa-se a utilizar intensamente os dispositivos de entrada e saída na operação do computador.



# Programotecas

- Viabilizam-se as programotecas, que podem passar a existir na forma de coleções de programas em formato carregável, disponibilizando ao usuário um conjunto já testado e operante de programas, resultante de trabalhos anteriores, podendo dar aos programadores maior facilidade no intercâmbio de seus trabalhos.



# Pequeno retrospecto

- Até aqui estudamos panoramicamente a evolução dos sistemas computacionais, em suas componentes de hardware e de software.
- Chegamos ao ponto em que o hardware se apresenta como um equipamento parcialmente eletro-eletrônico (fonte de alimentação, painel, processador e interfaces) e parcialmente eletromecânico (dispositivos ou equipamentos de entrada/saída de dados).

# Sistema de Programação

- Do ponto de vista do sistema de programação, foi visto que as motivações principais dos progressos nessa área foram:
  - A dificuldade e a baixa confiabilidade da programação e da operação manuais dos computadores e dos seus programas.
  - O caráter repetitivo e altamente sujeito a erros das atividades associadas à programação em linguagem numérica (especialmente a binária)

# Contribuições mais significativas

- Tal evolução se caracteriza pelas contribuições seguintes:
- No hardware:
  - Painel de chaves, lâmpadas e botões
  - Teclado numérico (octal, decimal, hexadecimal)
  - Equipamentos de entrada e saída de dados
- No software:
  - Linguagem absoluta
    - Linguagens de máquina absolutas, binária e simbólica. Montador absoluto
    - Loader binário (ou bootstrap, ou boot, ou carregador absoluto)
    - Formato binário absoluto carregável para entrada via loader
    - Dumper binário absoluto compatível com o formato para o loader
    - Biblioteca de programas executáveis, em formato absoluto carregável
  - Linguagem relocável
    - Linguagens de máquina relocáveis, binária e simbólica. Montador relocável
    - Relocador (relocator)
    - Formato binário relocável, para alimentar o linker
    - Biblioteca reutilizável de procedimentos relocáveis
    - Ligador (ou linker)

# Passos seguintes

- Com todos esses progressos, estamos aptos a produzir programas com maior rapidez, e até com um certo conforto, mas estamos longe do ideal, em que o programador deve preocupar-se apenas com a programação, deixando os outros quesitos da execução dos programas a cargo do sistema de programação.
- Nessa rota, para darmos o passo evolutivo seguinte, já é possível elaborar alguma forma de trabalho que permita evitar totalmente o uso das linguagens de máquina, numéricas ou simbólicas, pois nesta situação constituem o maior entrave a uma programação confortável.
- Na próxima aula vamos estudar como são as linguagens de alto nível, e como podem elas ajudar-nos a alcançar esse objetivo.