

Computadores e Programação

Visão Geral dos Sistemas (Aula 2 e 3)

Sistema de Computador

Consistem de hardware e sistemas de software que funcionam juntos para executar aplicações do usuário. As implementações específicas podem variar com o tempo, mas os conceitos fundamentais não mudam.

Bits + contexto

Um programa começa com um programa fonte escrito por um programador, em uma linguagem de alto nível e armazenado em um arquivo (*hello.c*)

O programa fonte é uma **sequência de bits** (0s e 1s), organizados em grupos de 8 bits chamados **bytes**.

Cada byte representa um **caractere** alfa numérico (de texto) A maioria dos sistemas modernos representa caracteres usando o padrão ASCII.

Arquivos como *hello.c*, constituídos exclusivamente de caracteres ASCII (i.e., sem formatação), são conhecidos como **arquivos texto**.

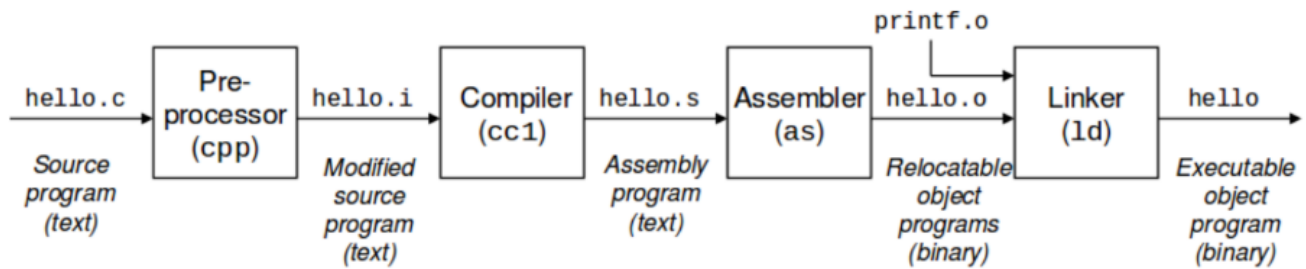
Todos os outros arquivos são conhecidos como **arquivos binários**

- Toda informação em um sistema (incluindo arquivos no disco, programas ou dados do usuário na memória, dados transferidos na rede) é representada como conjunto de bits.
- A única coisa que diferencia é o contexto no qual a informação (conjunto de bits) é vista (ex., a mesma sequência de bits pode ser vista como um número inteiro, uma string, uma instrução de máquina, etc.)

Tradução de programas

- Para executar o programa *hello* no sistema, as sentenças C devem ser traduzidas por outros programas em uma sequência de instruções de linguagem de máquina
- Essas instruções são então empacotadas no formato chamado programa (ou arquivo) objeto executável e armazenado como arquivo binário no disco
- No UNIX, a tradução de arquivo fonte em arquivo executável pode ser feita pelo programa gcc:

Sistema de Compilação



A tradução de programas é feita em quatro fases:

1. Pré-processamento (para gerar arquivo fonte com as inclusões especificadas)
2. Compilação (para gerar código de montagem)
3. Montagem (para gerar código binário relocável)
4. Ligação (para gerar código binário executável)

Os programas que executam as quatro fases de tradução são chamados coletivamente de sistema de compilação

Pré-Processamento

O pré-processador (`gcc -E`) modifica o programa original C tratando as diretivas que começam com o caracter `#`

Ex: a diretiva `# include <stdio.h>` diz para o pré-processador ler o conteúdo do arquivo `stdio.h` e inserir o conteúdo no programa fonte, gerando o arquivo `hello.i`

Compilador

O compilador (`gcc -S`) traduz o arquivo texto `hello.i` no arquivo texto `hello.s`, o qual contém o programa em linguagem de montagem

Cada sentença no programa em linguagem de montagem descreve exatamente uma instrução de máquina em um formato de texto padrão

A linguagem de montagem provê uma linguagem de saída comum para diferentes compiladores e linguagens de alto nível

Montagem

O montador (`gcc -c`) traduz o arquivo `hello.s` em instruções de linguagem de máquina, empacotadas no formato conhecido como programa objeto relocável e armazena em um arquivo `hello.o`

Trata-se de um arquivo binário, cujos bytes codificam instruções em linguagem de máquina (ao invés de caracteres)

Ligação

O programa `hello` chama a função `printf` que não é implementada em `hello.c` (faz parte da biblioteca C padrão)

A função `printf` reside em um arquivo objeto precompilado chamado `printf.o`, o qual deve ser

incorporado ao programa hello. o (ligação)

O ligador (gcc) trata essa incorporação

O resultado é o arquivo hello, um arquivo objeto executável que está pronto para ser carregado na memória e executado pelo sistema

Leitura e interpretação de instruções