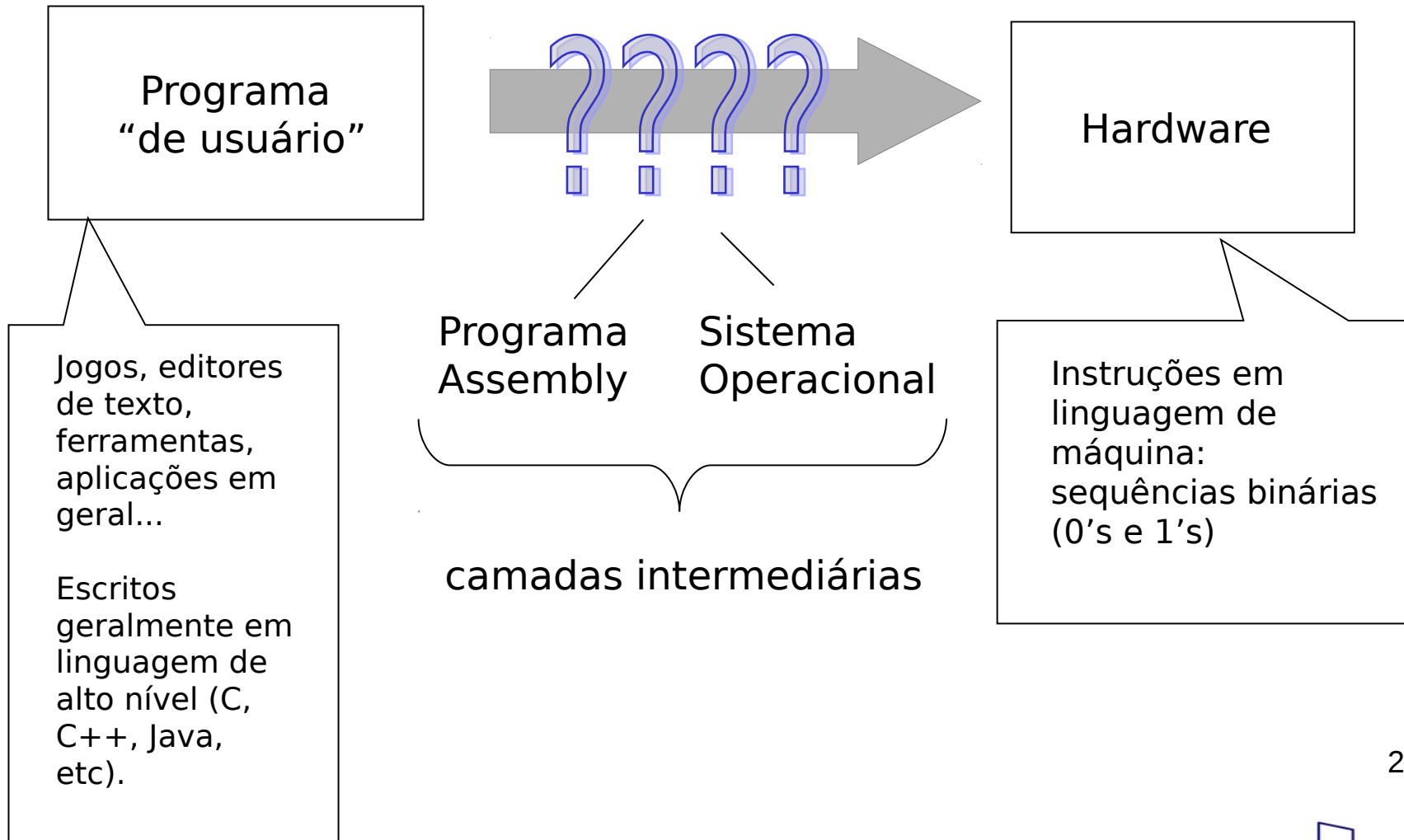


# Software Básico (INF1018)

Noemi Rodriguez  
Ana Lúcia de Moura

<http://www.inf.puc-rio.br/~inf1018>

# Hierarquia de Abstrações



# Objetivo do Curso

---

Entender como funciona um computador típico, como visto no nível de Linguagem de Montagem

Perspectiva de *software* (foco no **programador**)

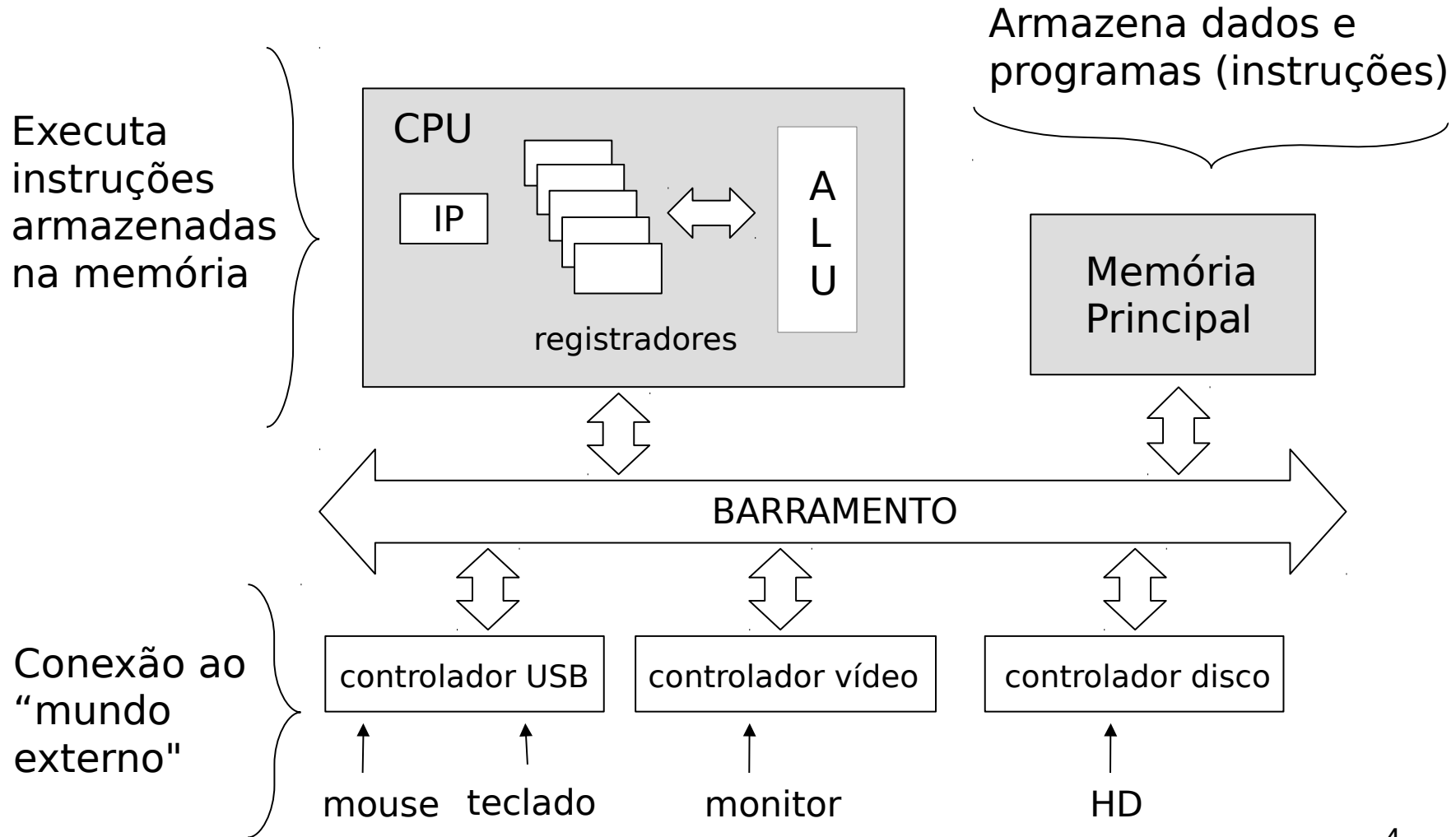
- suporte para abstrações de uma linguagem de programação (C)

Programadores precisam de um entendimento sólido da hierarquia de abstrações de um Sistema de Computação

- otimizar o uso de recursos/desempenho de programas
- entender e saber evitar bugs (representação de dados, manipulação de memória, estouro da pilha...)

**porque abstrações vazam !**

# Arquitetura Típica



# Geração de um Executável

---

*code/intro/hello.c*

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("hello, world\n");
6 }
```

*code/intro/hello.c*

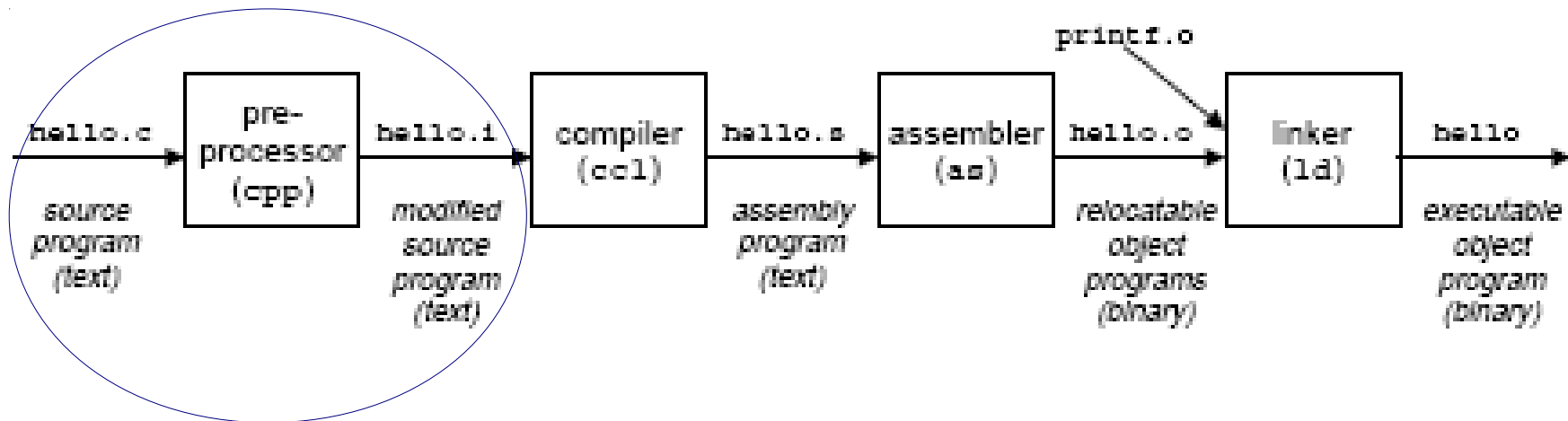
O programa **fonte** (arquivo texto) deve ser "traduzido" para uma sequência de instruções de linguagem de máquina, que é armazenada em um arquivo binário (**executável**)

➔ essa tradução é realizada em 4 passos

# Passo 1: pré-processamento

---

```
ana@sol:~/inf1018$ gcc -o hello hello.c
```

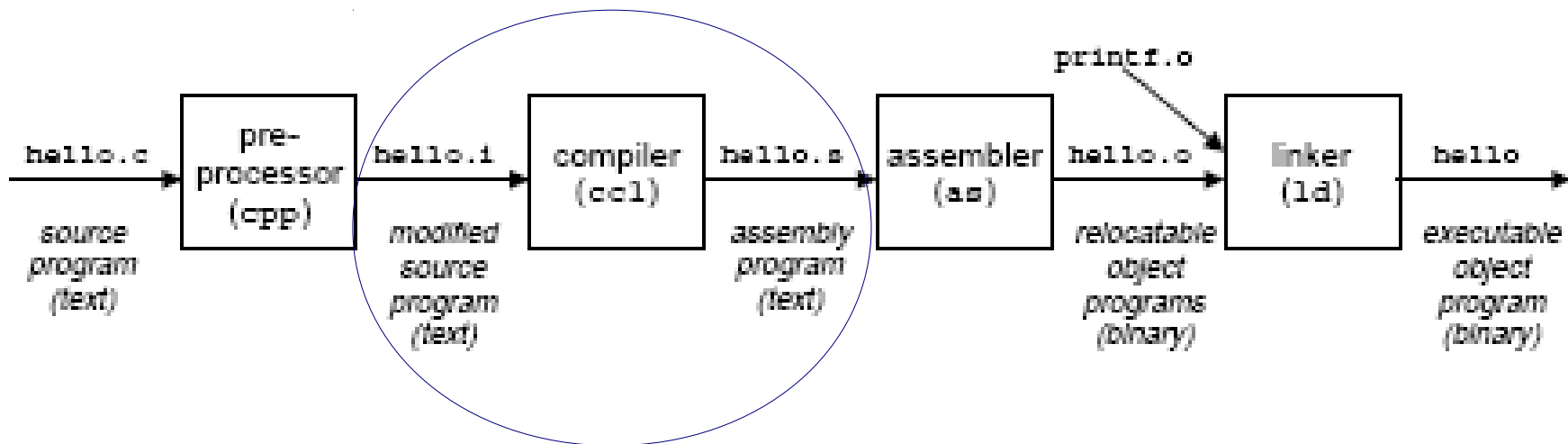


Modifica o programa fonte C de acordo com as diretivas começadas com #

**#include <stdio.h>** faz com que o pré-processador leia o arquivo **stdio.h** e o insira no programa fonte

# Passo 2: compilação

```
ana@sol:~/inf1018$ gcc -o hello hello.c
```



Traduz o programa fonte modificado para um programa em linguagem de montagem (**assembly**)

➔ é um formato de saída comum para os compiladores de linguagens de programação de alto nível

# Linguagem de Montagem

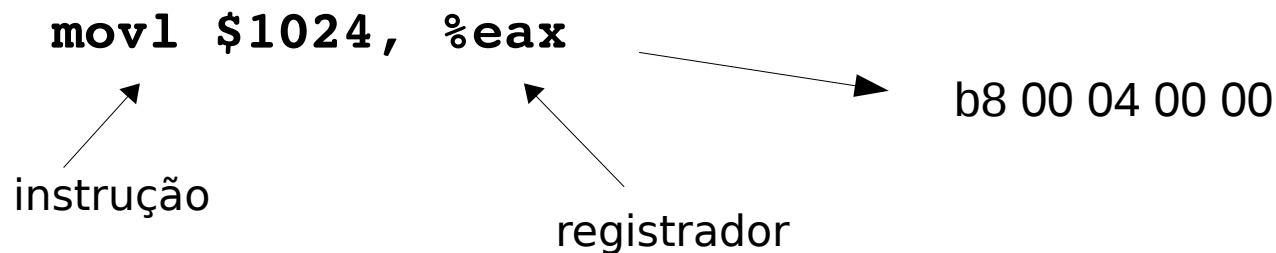
---

É um mapeamento bastante direto para a linguagem de máquina

- a linguagem de montagem (e de máquina) é específica para uma plataforma
- cada linha do código fonte corresponde a uma instrução para o processador

Tem várias facilidades para um programador

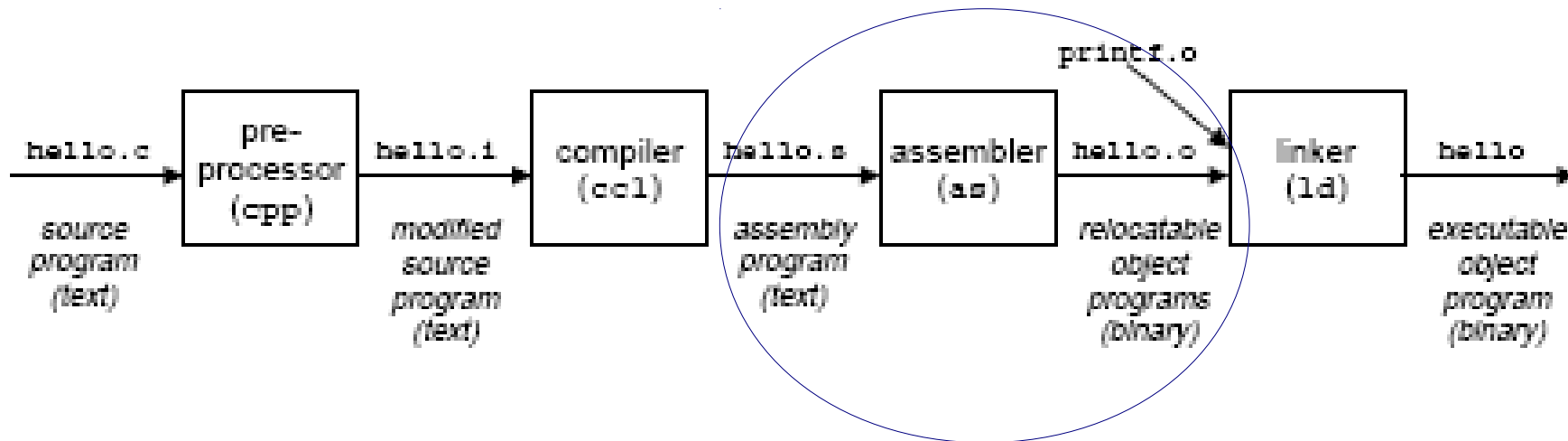
- tipos básicos de dados (inteiros, endereços)
- uso de *mnemônicos* (nomes) para representar instruções, registradores





# Passo 3: montagem

```
ana@sol:~/inf1018$ gcc -o hello hello.c
```

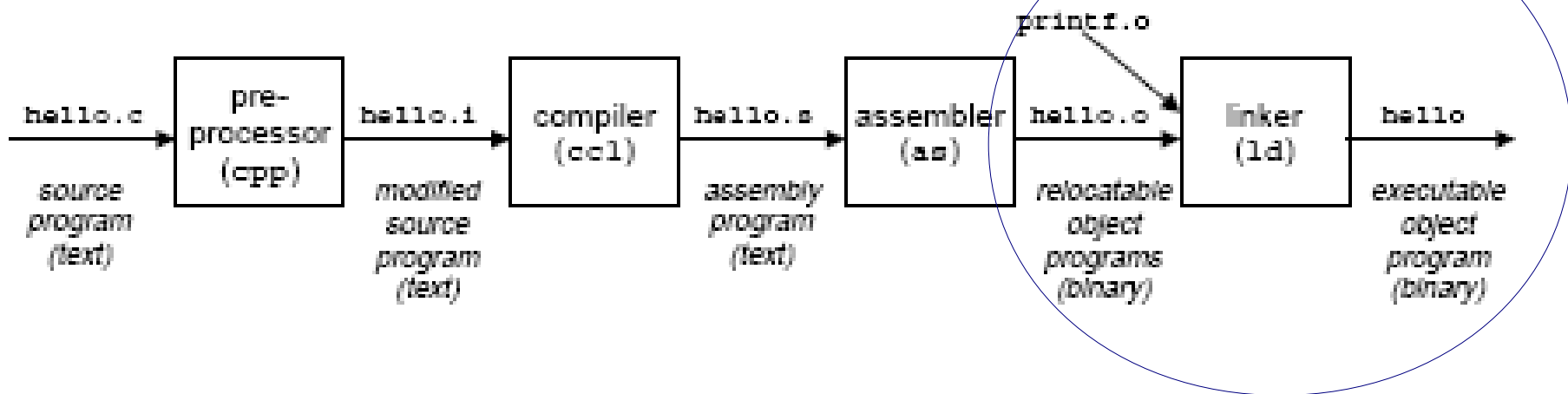


Traduz o programa fonte assembly para instruções da linguagem de máquina (**objeto**)

➡ armazenado como um arquivo **binário** com extensão **.o**

# Passo 4: ligação (amarração)

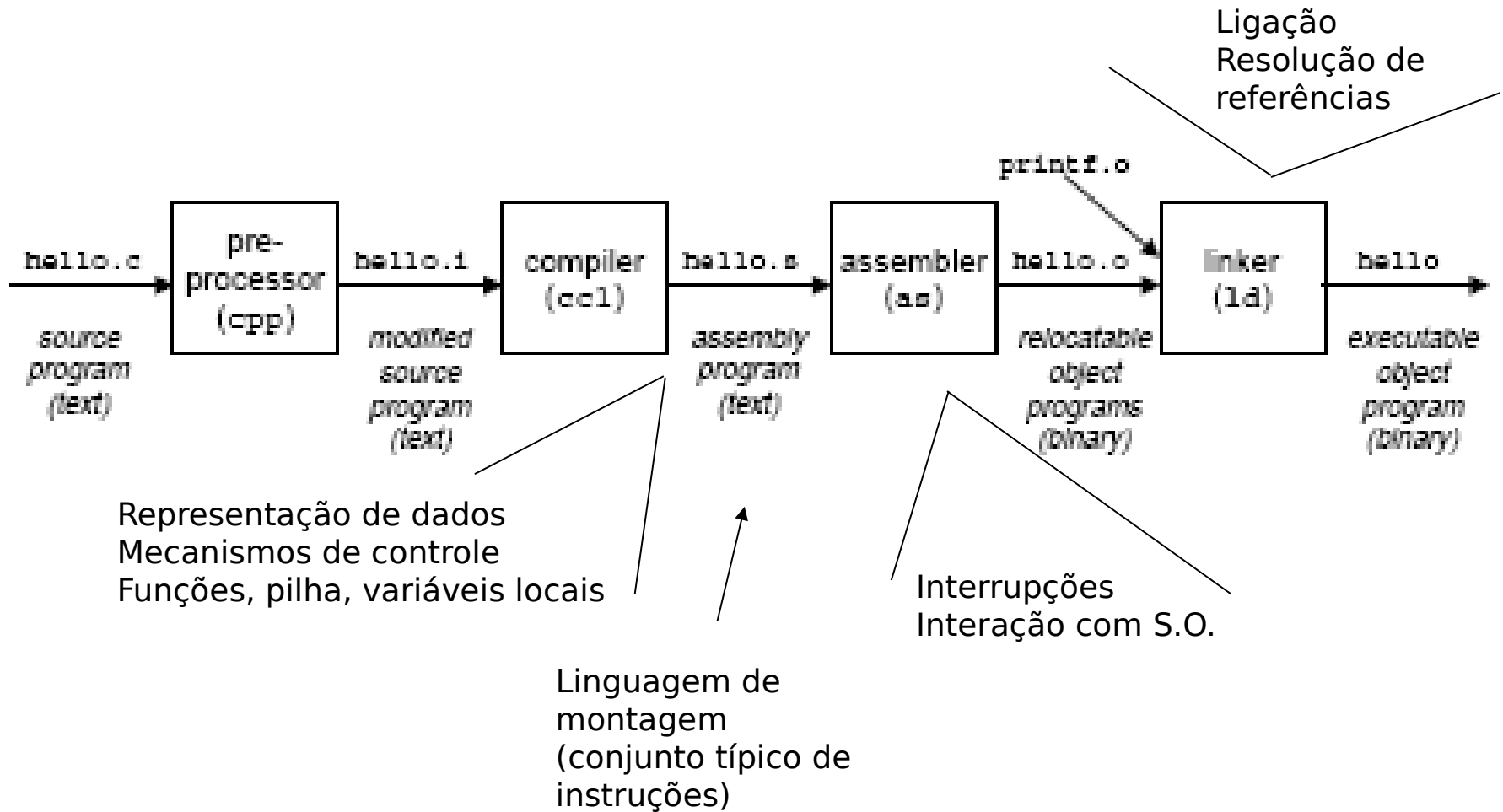
```
ana@sol:~/inf1018$ gcc -o hello hello.c
```



Gera um **executável** a partir do(s) módulo(s) objeto

- uma aplicação pode ser composta por vários arquivos fonte, cada um gerando um módulo objeto diferente
- alguns módulos objeto podem estar armazenados em **bibliotecas**
- o ligador faz a união dos módulos necessários para gerar o executável

# Conteúdo do Curso



# Material Básico de Referência

---

## Computer Systems, A Programmer's Perspective

Randal Bryant and David O'Hallaron

**pasta 161 CA Engenharia**

Slides, vídeos e exercícios no *site* da disciplina

**<http://www.inf.puc-rio.br/~inf1018>**

# Critério de Avaliação

---

Cada grau é a média geométrica de uma prova (peso 2) e um trabalho (em dupla)

$$G1 = \sqrt[3]{(P1^2 \times T1)}$$

$$G2 = \sqrt[3]{(P2^2 \times T2)}$$

$$\text{Média} = (G1 + G2) / 2$$

- se **G1 e G2 ≥ 3.0** e **M ≥ 6.0**, é a nota final (NF)
- caso contrário: **NF = (G1 + G2 + 2 x PF) / 4**
- se **NF ≥ 5.0** o aluno está aprovado