Estrutura de Dados (20-2)
Revisão de Conceitos Básicos

Profs.: Denio e Geomar

Programas

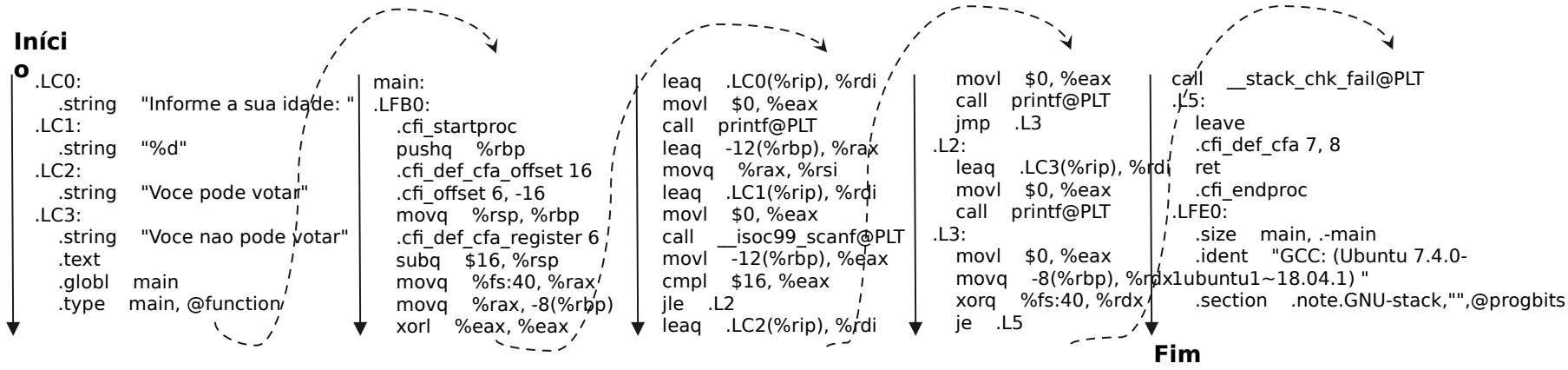
- Os primeiros programas eram implementados em linguagem de máquina
- Era muito difícil programar (décadas de 1940 e 1950)
- Por exemplo, um programa que fizesse os seguintes passos

```
Informe a sua idade: 20
Você pode votar
```

Programas

- Em linguagem de baixo nível (máquina) ficaria assim:

```
Informe a sua idade: 20
Você pode votar
```



Programas

- Utilizando uma linguagem de alto nível:

```
Informe a sua idade: 20
Você pode votar
```

```c
//C
#include <stdio.h>
int main(void)
{
   int id;
   printf(“Informe a sua idade: “);
   scanf(“%d”,&id);
   if (id>=17)
      printf(“Você pode votar\n”);
   else
      printf(“Você não pode votar\n”);
   return 0;
}
```

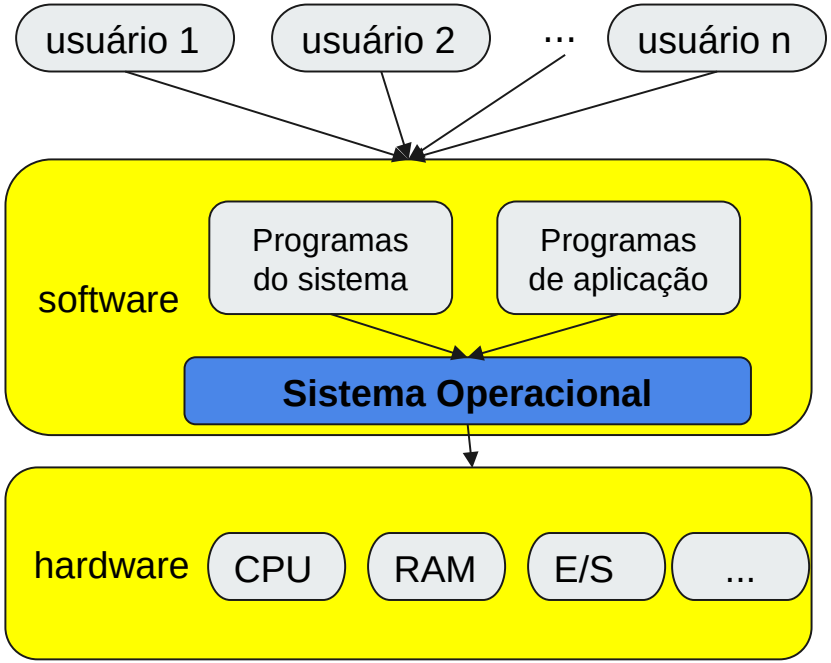
```python
#python
id=input(‘Informe a sua idade: ‘)
id=int(id)
if id>=17:
   print(‘Você pode votar’)
else:
   print(‘Você não pode votar’)
```

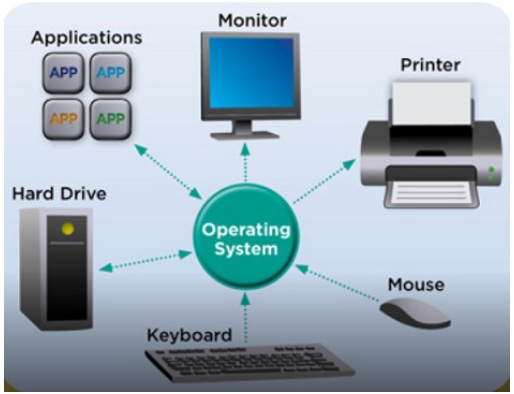
```
{Pascal}
program vota;
var
  id : integer;
begin
  write('Informe a sua idade: ');
  readln(id);
  if id>=17
  then
      writeln('Você pode votar')
  else
      writeln('Você não pode votar')
end.
```

Executando Programas

- Como um programa escrito em linguagem de alto nível é **entendível** pelo computador?
  - Existe **um programa mestre** (sistema) que torna os computadores mais acessíveis pelos humanos
  - Faz as transformações de nossas requisições de entrada (teclado, voz, mouse, ...) para o computador
  - Transforma as respostas do computador para um formato entendível por nós (monitor, impressora, som, video, ...)
- Sistema operacional (SO)
  - Gerenciar os recursos de hardware e software do computador, e oferece serviços comuns para os programas serem executados

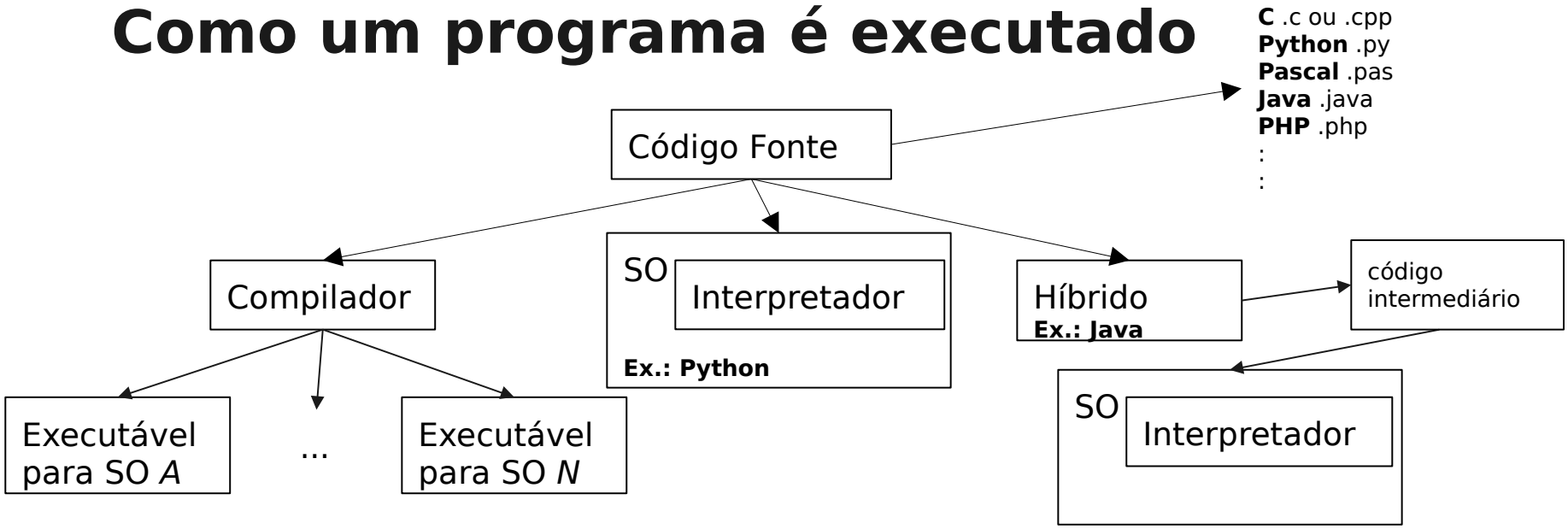
Sistema Operacional (SO)





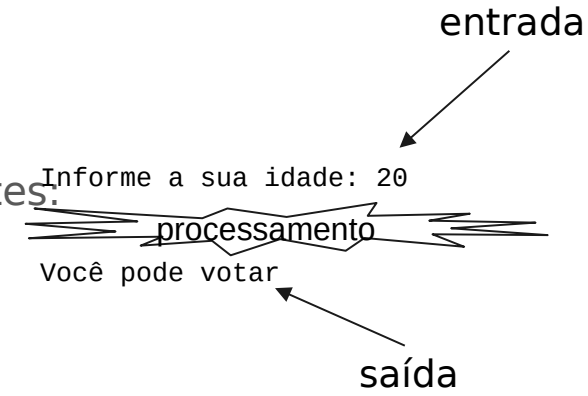
Sistema Operacional (SO)

Como um programa é executado



Algoritmos

- Um algoritmo/programa é formado por 3 partes:
  - Entrada
  - Processamento
  - Saída




```
#python
id=input('Informe a sua idade: ')
id=int(id)
if id>=17:
   print('Você pode votar')
else:
   print('Você não pode votar')
```

entrada

processamento

saída

Algoritmos

- Entrada
  - Informações (dados) que o programa precisa para resolver o problema
  - No nosso exemplo: **a idade do usuário**
- Processamento
  - Processar os dados de entrada para obter a saída
  - No nosso exemplo: verificar se a **idade informada é maior ou igual a 17 ou não** (idade para votar)
- Saída
  - Apresentar o resultado do processamento. Este resultado deve ser condizente com o problema que o programa está resolvendo
  - No nosso exemplo: imprimir uma mensagem informando se alguém com a idade digitada pode ou não votar.

Entrada

- Os dados informados pelo usuário durante a entrada devem ser armazenados em uma região da memória principal do computador (chamada RAM)
  - Variável
  - O programador deve criar as variáveis dando nomes as mesmas:
    - O nome deve começar com uma letra. Não pode ter espaços, caracteres especiais (por exemplo: #, @, &, (,), !, entre outros)
    - Nomes válidos: idade, tipo2, val_hora, nomeCliente, minhaHora, salario, ...
    - Inválidos: nome cliente, #idade, salário, tempo(dia), eu&voce, ...
- As entradas geralmente são feitas pelo **teclado**, **mouse**, **telas** do tipo **touch screen**, l**eitores de códigos de barras**, entre outros.

Processamento

- Os dados de entrada, representados pelas variáveis são utilizados para produzir a saída
  - Expressões matemáticas
  - Expressões lógicas
  - Comandos condicionais
  - Comandos de iteração (laços)
  - :

Saída

- Os resultados obtidos pelo processamento (que podem ser armazenados em variáveis) são apresentados para o usuário do aplicativo
  - Geralmente na tela do computador/celular através de comandos específicos (**print**)
  - Na impressora
  - Pelo alto falante (som)
  - Podem ser armazenadas diretamente do disco

Exemplo

- Suponha um algoritmo para dado o ano do nascimento do usuário e outro ano qualquer, calcular a idade do usuário no ano qualquer
  - **Entrada**: devemos solicitar ao usuário que informe o ano de nascimento dele e também o ano para calcular a idade
    - Vamos armazenar o ano de nascimento na variável `anonasc` e o ano para calcular a idade de `anoatual`
  - **Processamento**: agora temos que encontrar a fórmula matemática para calcular a idade, pensando um pouco, sabemos que se subtrairmos o ano para calcular a idade (`anoatual`) pelo ano de nascimento (`anonasc`), encontraremos a idade do usuário. O resultado do cálculo deve ser armazenado em uma variável para ser utilizado na saída (`idade`)
    - idade = anoatual - anonasc
  - **Saída**: agora basta apresentarmos o resultado do cáculo
    - Imprimir o conteúdo da variável idade