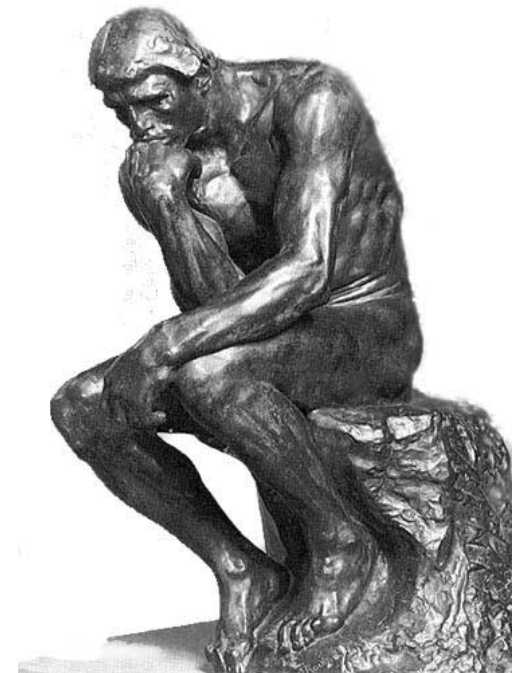


Linguagens de Programação

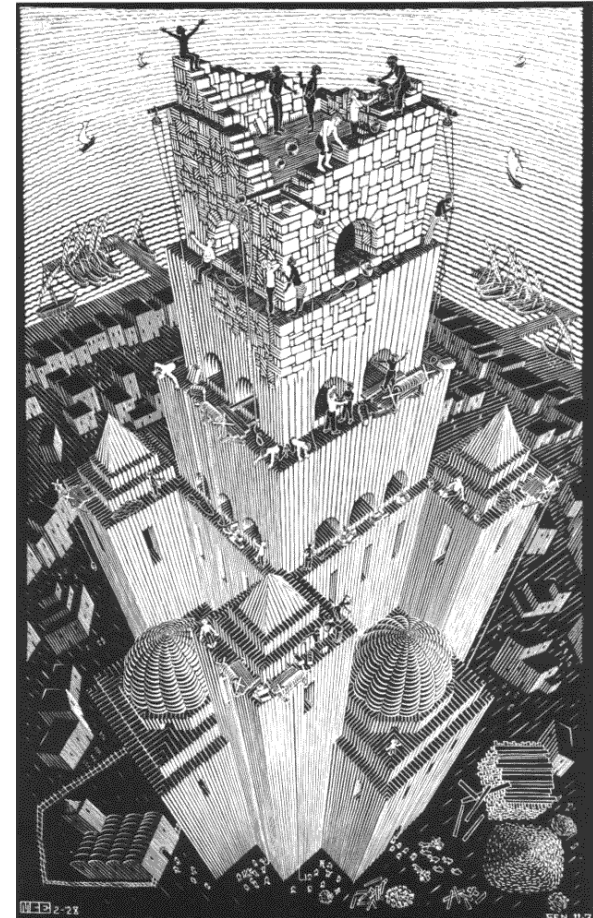
Fernando Magno Quintão Pereira

- O que são linguagens de programação?
- Por que elas existem?
- Como computadores eram programados antes das linguagens de programação?



A Torre de Babel

- Existem entre 5.000 e 6.000 línguas faladas em nosso planeta.
- Cerca de 200 idiomas possuem mais de um milhão de falantes.
- Como descrever um idioma? Que elementos estão presentes na descrição de uma linguagem?



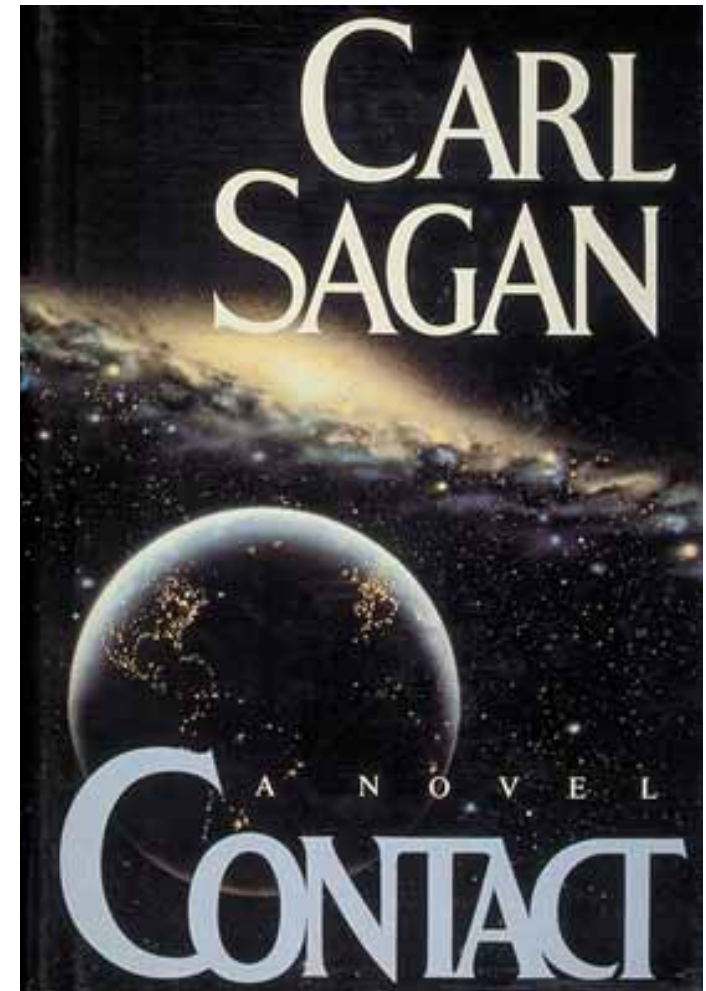
Computadores também *conversam*

- Como é a linguagem *falada* pelos computadores?
 - Que símbolos ela usa?
 - Quais palavras?
 - Como seria a gramática dessa língua eletrônica?



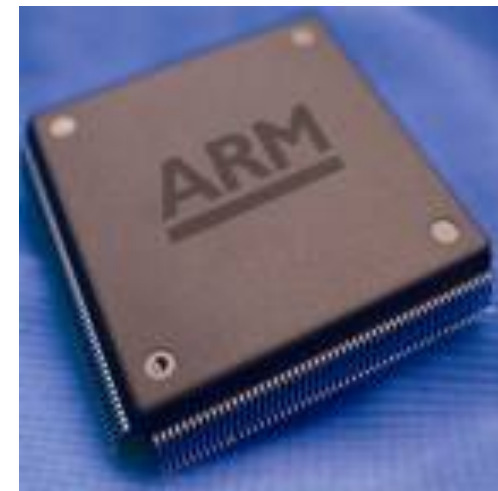
Vamos falar zero-um-nês?

- Computadores possuem cordas vocais muito simples: ou emitem som, ou não emitem
- É possível haver uma linguagem com apenas dois símbolos?
- Porque somente dois símbolos?



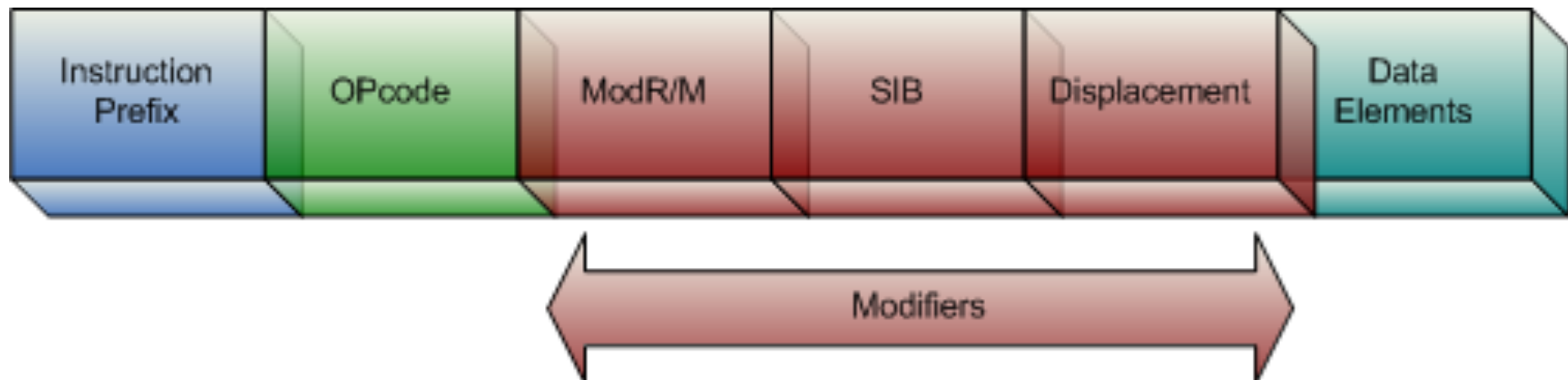
Dialetos do zero-um-nês

- Há muitas linguagens de zeros e uns diferentes, assim como há muitas linguagens diferentes usando caracteres latinos: inglês, português, espanhol, etc.
- Quem me dá exemplos de zero-um-nês diferentes?



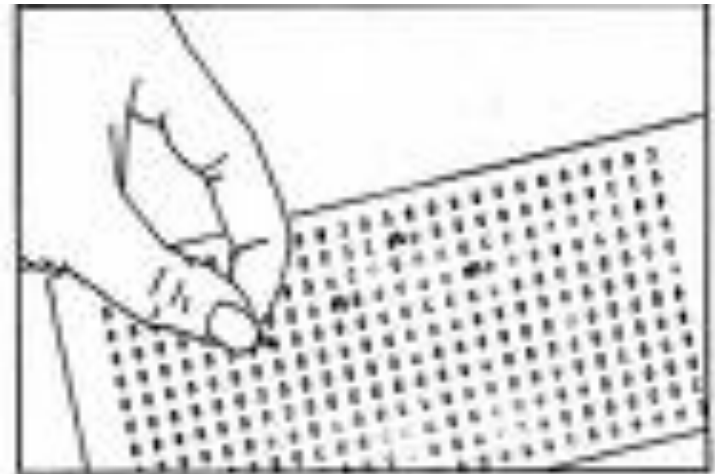
“The book is on the table”

- Cada instrução em zero-um-nês possui um nome, chamado *opcode*, e operandos.
- Instruções mudam o estado do computador.
- Que tipos de instruções poderiam existir?
- Falar zero-um-nês deve ser fácil, não é?



Mas não é não...

- Antigamente programar computadores era muito difícil.
- Qual o problema com zero-um-nês?
- Alguém aí conhece cartões perfurados?
- Como deixar zero-um-nês mais fácil de usar?



E veio a Deusa

- Palavras são mais fáceis de lembrar que sequências de zeros e uns.
- Por exemplo: qual instrução é mais fácil de ler: **mov \$1, AL**, ou
10110000
01100001?



O Que este programa faz?

```
    movl $5, %eax  
    movl $1, %edx  
.L4:  
    imull %eax, %edx  
    decl %eax  
    testl %eax, $0  
    jg .L4
```

O Que este programa faz?

```
movl $5, %eax
```

Coloque 5 em %eax

```
movl $1, %edx
```

Coloque 1 em %edx

```
.L4:
```

```
imull %eax, %edx
```

Multiplique %eax por %edx, e coloque o resultado em %edx

```
decl %eax
```

Subtraia 1 de %eax

```
testl %eax, $0
```

Teste se %eax é Zero

```
jg .L4
```

Se Zero, então vá para .L4

O Montador

- As pessoas falavam *assembly*, mas os computadores ainda falavam zero-um-nês.
 - Era preciso um tradutor.
- O que um tradutor deste tipo deveria ser capaz de fazer?

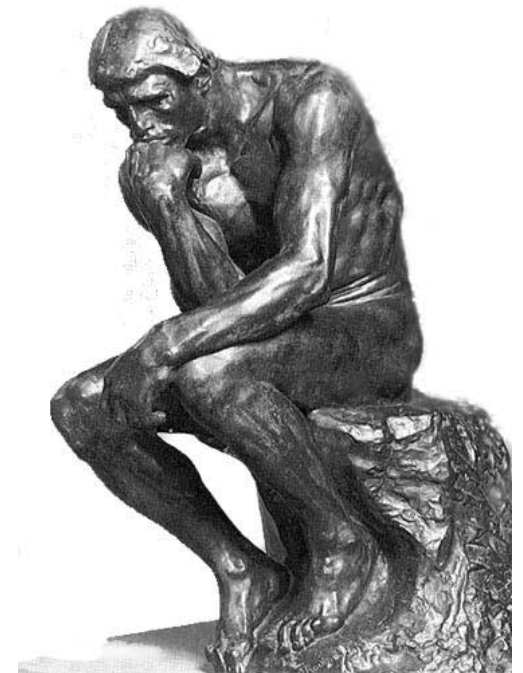


A Deusa não foi suficiente

- Programar em *assembly* ainda era difícil.
- Os programadores queriam que os computadores fossem capazes de falar línguas ainda mais parecidas com linguagens humanas.

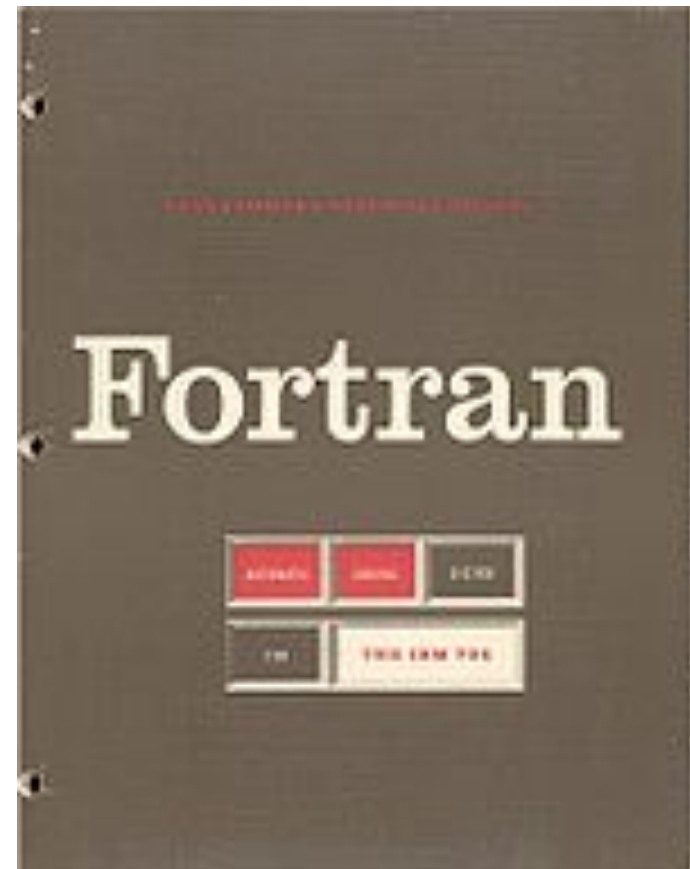


- Quais foram as primeiras linguagens de programação?
- Quem foram os pais dessas linguagens?



Surge Fortran

- *John Backus* estava com preguiça de escrever programas em *assembly*.
- IBM 1953/54
- Programar ficou umas 20 vezes mais fácil
 - Mas as pessoas ainda estavam relutantes...
Porque?



Exemplo de programa em Fortran

Fortran

```
nfact=1  
do i=1, 5  
    nfact = nfact*i  
enddo
```

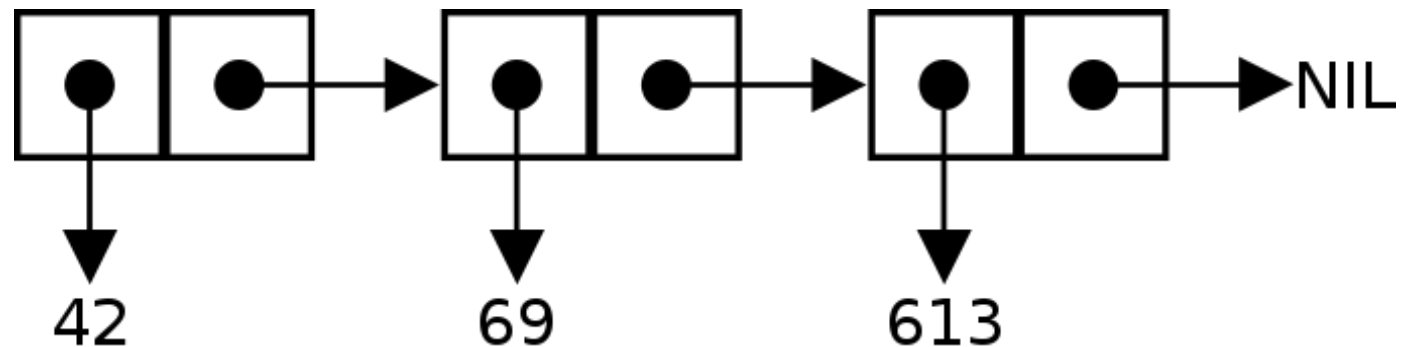
Que novidades surgiram
com Fortran?

Assembly

```
movl $5, %eax  
movl $1, %edx  
.L4:  
imull %eax, %edx  
decl %eax  
testl %eax, $0  
jg .L4
```


E Surge LISP

- 1958, *Massachusetts Institute of Technology*
- Professor *John McCarthy*.
- Uma notação simples, baseada em funções matemáticas.
- Muitos parênteses,
- E listas...



Exemplo de Programa em LISP

LISP

```
(defun factorial (n)
  (if (<= n 1)
    1
    (* n (factorial (- n 1)))))
```

Recursão!



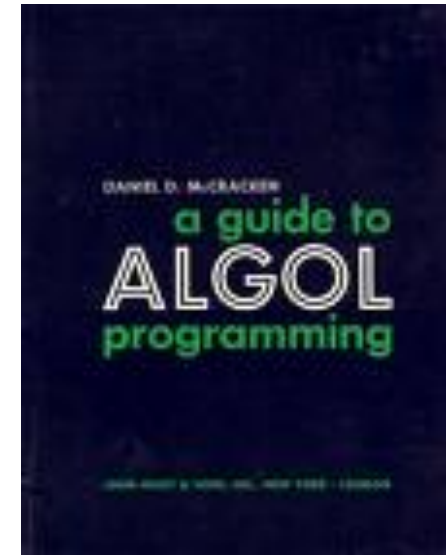
E quando, nos anos 70,
os soviéticos conseguiram
as últimas 500 linhas do
sistema de mísseis
americanos...

Fortran

```
nfact=1
do i=1, n
    nfact = nfact*i
enddo
```

ALGOL – um time de estrelas

- Precisava-se de um padrão para algoritmos.
- Um comitê foi formado em 1958.
 - John Backus
 - C. A. R. Hoare
 - John McCarthy, etc
- Deste comitê nasceu ALGOL 58.
- Talvez a mais influente linguagem de programação.



ALGOL – exemplo

```
integer procedure Factorial(m); integer m;  
Begin  
    integer F;  
    F := if m=1 then 1 else m*Factorial(m-1);  
    Factorial := F  
end;
```

- Vocês já viram algo parecido com isto?

E COBOL

- COBOL foi feita para negócios:
 - Contadores, economistas, etc
 - Como deveria ser uma linguagem assim?
- 1958: COBOL foi criada por um comitê.
 - Indústria, governo e academia
- Ainda usada em muitas companhias, até em BH!



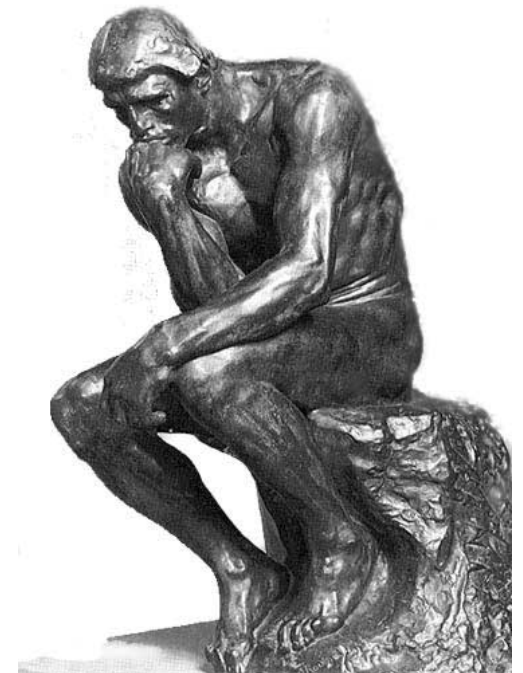
Exemplo de programas em COBOL

ADD YEARS **TO** AGE.

MULTIPLY PRICE **BY** QUANTITY **GIVING**
COST.

SUBTRACT DISCOUNT **FROM** COST **GIVING**
FINAL-COST.

- Quantas linguagens de programação existem?
- Quais as linguagens mais populares?



Quantas são?

- A editora O'Reilly diz que existem 2.500 linguagens de programação documentadas.
- A wikipédia documenta 650.
- Existem muitas...
- Mas, porque tantas?



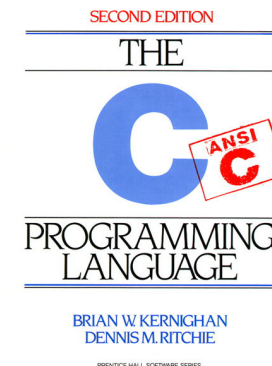
Propósitos diferentes

- Fortran servia para cálculos científicos.
- Lisp era usada em teoria da computação.
- COBOL foi feita para aplicações comerciais.
- Algol é uma linguagem acadêmica.
- E as outras linguagens que conhecemos?



Quais são as linguagens pop?

- Dados retirados de www.tiobe.com
 - Java: 18.71%
 - C: 16.89%
 - PHP: 10.39%
- Google code: C, Java, C++, PHP
- Craigslist: PHP, C, SQL
- Que outras medidas?



Alguém aí fala Javanês?

- De acordo com muitos critérios, Java é a a linguagem mais popular.
- Para que serve Java?
- Como esta linguagem surgiu?
- O que ela tem de mais?



Um exemplo de javanês:

```
public class Fact {  
    public static void main(String a[]) {  
        int n = 5;  
        int fact = 1;  
        while (n > 1) {  
            fact *= n;  
            n--;  
        }  
        System.out.println(fact);  
    }  
}
```

é A, é B, é C...

- C surgiu em 1972, e foi, durante muitos anos, a linguagem de programação mais popular.
- Porque C tem este nome?
- O que a gente faz com C?
- Porque C foi tão popular?
- Quais os problemas com C?
- C teve grande influência...



Falando em C...

```
int main() {  
    int n = 5;  
    int fact = 1;  
    while (n > 1) {  
        fact *= n;  
        n--;  
    }  
    printf("%d\n", fact);  
}
```

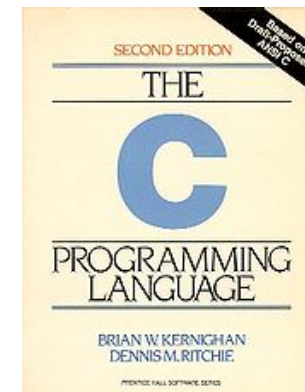
- Alguém já viu isto antes?

C teve grande influência...

```
int n = 5;  
int fact = 1;  
while (n > 1) {  
    fact *= n;  
    n--;  
}
```

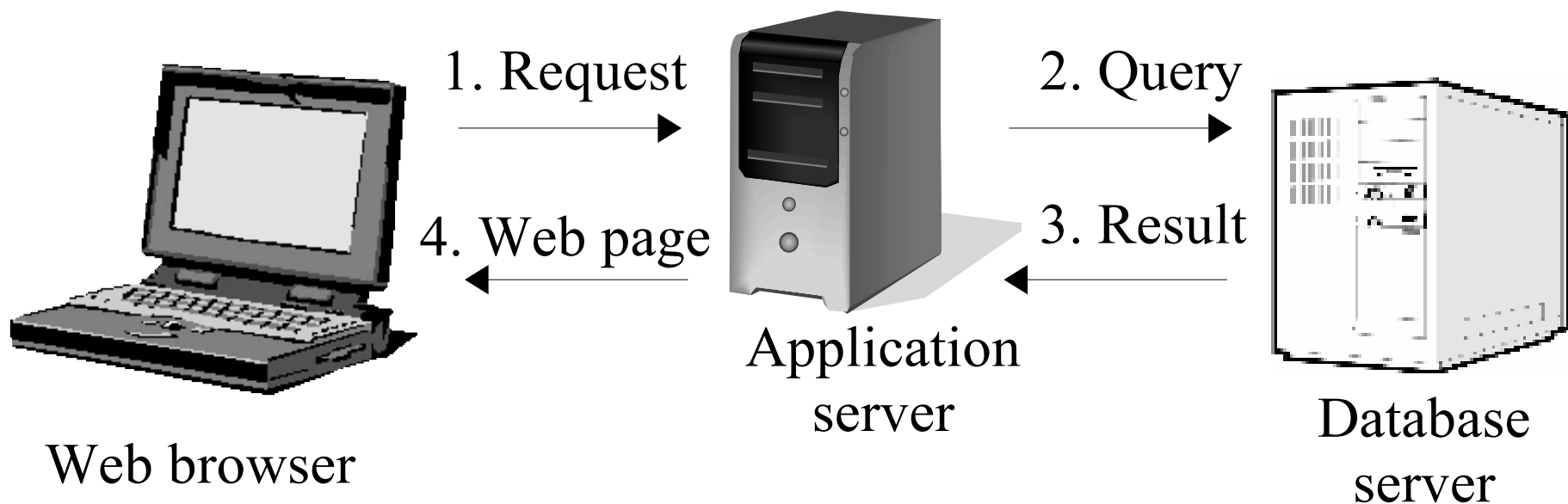


```
int n = 5;  
int fact = 1;  
while (n > 1) {  
    fact *= n;  
    n--;  
}
```



A Internet respira PHP

- Alguém aqui já programou em PHP?
- O que este nome quer dizer?
- Como deve ser uma linguagem para desenvolvimento web?



Um exemplo de PHPês:

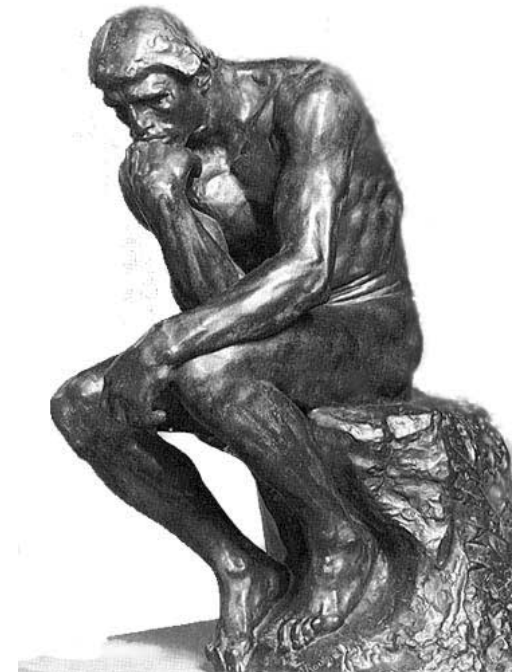
```
$id = $_GET["user"];

if ($id == '') {
    echo "Invalid user: $id"
} else {
    $getuser = $DB->query
        ("SELECT * FROM 'table' WHERE id='$id'");
    echo $getuser;
}
```



- Alguém notou um pouquinho de C aí?
- Qual o tipo da variável `$id`?

- Computadores falam zero-um-nês, nós falamos linguagens de programação... quem traduz estas coisas?
- E como esta tradução é feita?

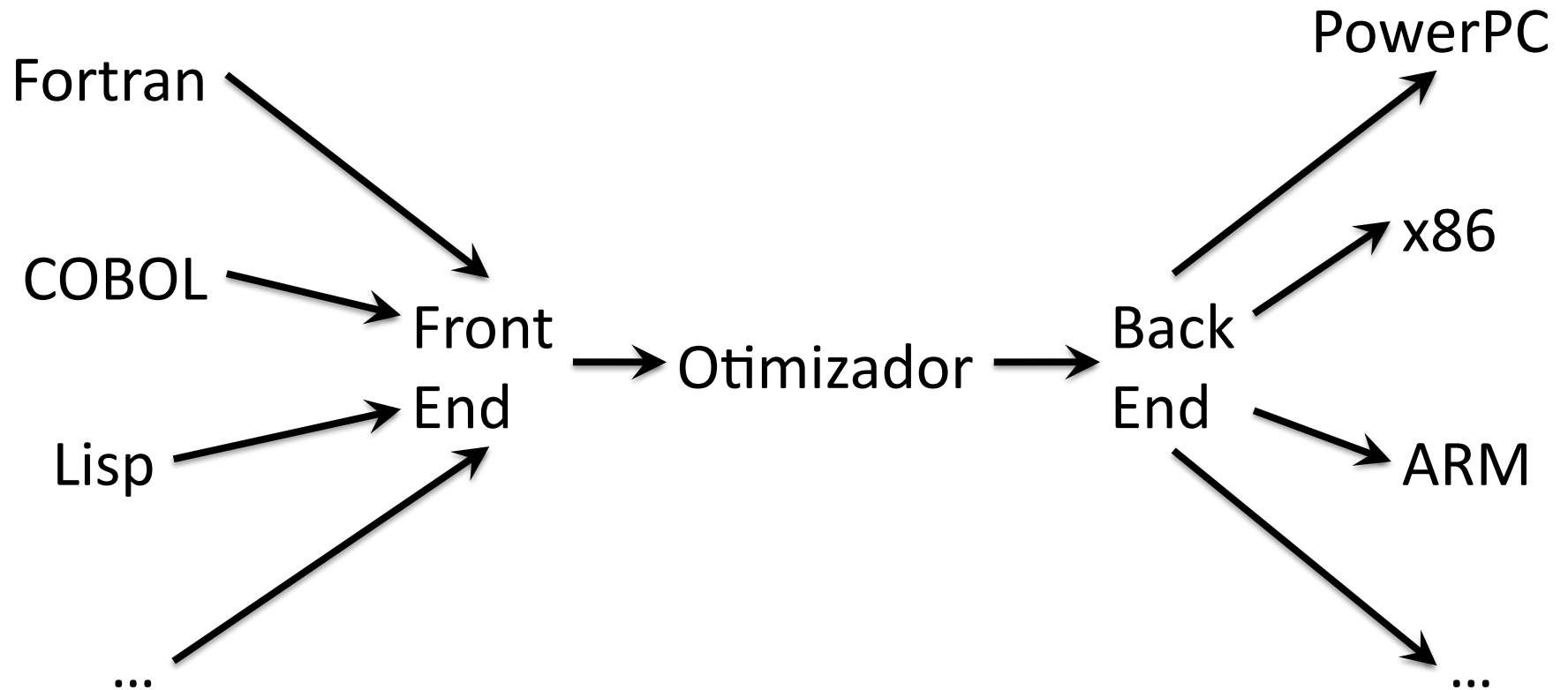


Compiladores são pontes

- O primeiro compilador foi, provavelmente, o **A-0** de *Grace Hopper* (1949).
- Linguagens de programação diferentes possuem diferentes compiladores.
- Mas o mesmo compilador também pode compilar linguagens diferentes.



Anatomia de um compilador



Máquinas Virtuais

- Uma máquina virtual é um *hardware* implementado em *software*.
- Porque isto é interessante?
- Que linguagens executam em máquinas virtuais?
- Ainda é necessário um tradutor?



Às vezes, tudo é interpretado

- Um interpretador não produz código de máquina. Ao contrário, ele lê o código do programa fonte, e interpreta cada comando encontrado.
- Quais as vantagens de um interpretador?
- Quais linguagens são interpretadas?
- Será que há alguma linguagem que necessariamente tenha de ser interpretada?
- Essas coisas são eficiente?

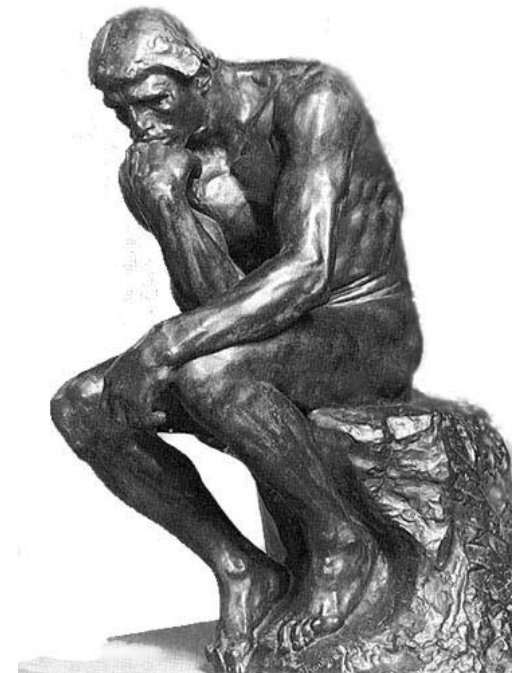


Fazemos *just-in-time*

- Algumas linguagens são compiladas enquanto estão sendo interpretadas.
 - *JavaScript*, por exemplo.
- E de onde vem a eficiência?
- Será que dá para fazer melhor que um compilador tradicional?



- Existe uma linguagem de programação “mais poderosa” que todas as outras?
- Se existe, que linguagem é esta?
- Mas como medir este “poder”?



Fácil ou Difícil

1. Encontre a rede de estradas mais curta que liga todas as cidades de Minas Gerais.
2. Encontre a menor rota passando por todas as cidades, sem repetir.
3. Dado um programa P para resolver (2), verifique se a primeira coisa que P imprime é *Nova Era*.



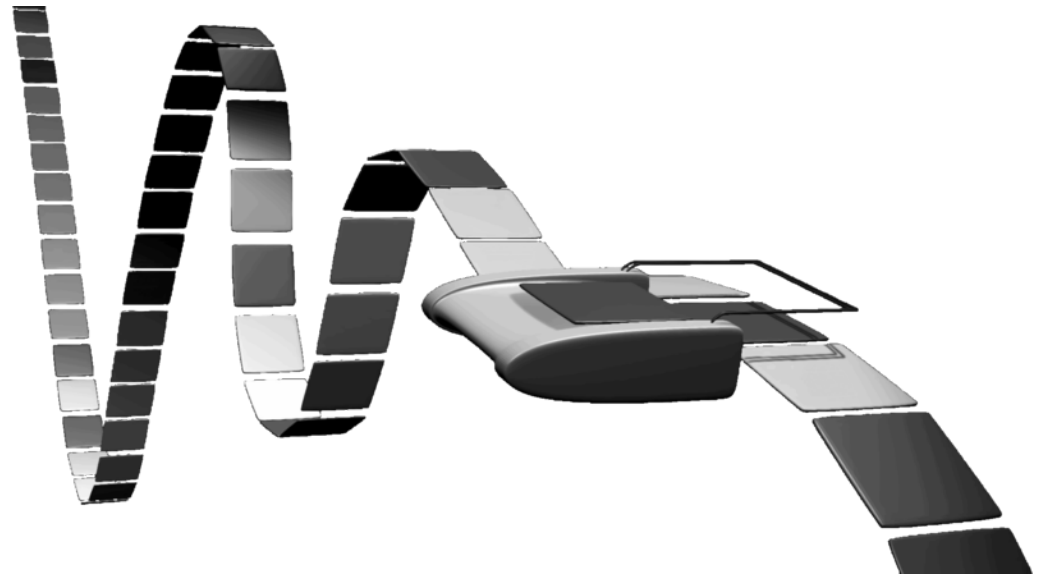
Há que sermos humildes

- A máquina de Turing é um modelo teórico que define todos os problemas que são computáveis.
 - Estado, fita, leitor, símbolos, instruções.
- Se não há solução na Máquina de Turing, então não tem jeito mesmo...



Linguagens Turing-Completas

- Se uma linguagem é equivalente à Máquina de Turing, então ela é *Turing-Completa*.
- Quase toda LP é Turing-Completa.
- Mas existem linguagens que não o são. Algum exemplo?



Brain-fuc*

Um arranjo muito grande, contendo números.

Oito comandos:

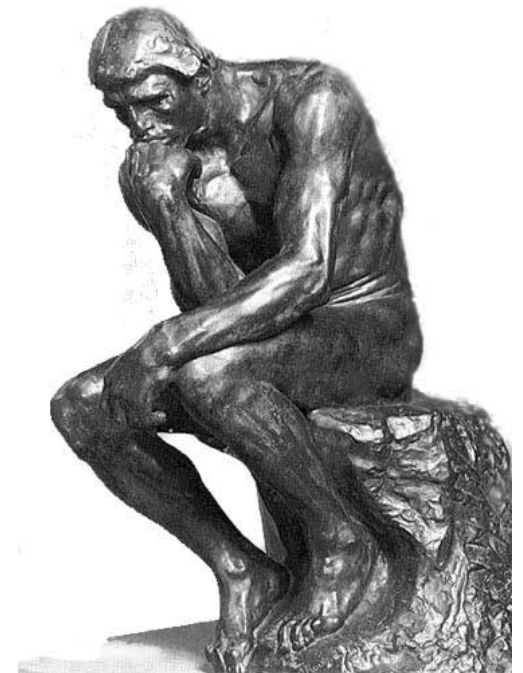
- > move uma posição para direita
- < move uma posição para esquerda
- + soma um à posição corrente (PC)
- subtrai um da PC
- . imprime conteúdo da PC
- , lê entrada e armazena na PC
- [vai para comando após] se PC é zero
-] volta para comando após [se PC não é zero.

O que estes programas fazem?

[-] ou [> + < -]



- Essas linguagens todas que a gente viu... Java, PHP, C, Fortran, COBOL, Algol, etc, etc... elas são muito parecidas: variáveis, loops, comandos... Será que não existe nenhum outro paradigma não?



Linguagens Imperativas e Declarativas

- Linguagens imperativas:
 - O programa são instruções.
 - Atribuições, *loops*, sequências.
 - Efeitos colaterais e estado.
- Linguagens declarativas:
 - O programa descreve uma verdade.
 - Ausência de efeitos colaterais.
 - *Loops* via chamada de funções recursivas



SML

- O programa é um conjunto de funções.
 - Programas são provas por indução.
- Principais estruturas de dados são listas e tuplas.

```
fun sum [] = 0
    | sum (h::t) = h + sum t
```

```
fun filter [] _ = []
    | filter (h::t) f =
        if (f h)
        then h :: (filter f t)
        else (filter f t)
```



Sorting

```
fun leq a b = a <= b
```

```
fun grt a b = a > b
```

```
fun filter _ nil = nil  
  | filter f (h::t) =  
    if f h then h :: filter f t else filter f t
```

```
fun qsort nil = nil  
  | qsort (h::t) =  
    (qsort (filter (grt h) t))  
    @ [h] @  
    (qsort (filter (leq h) t))
```



Prolog

- O programa é um conjunto de restrições:
 - Se A é verdade, e $A \rightarrow B$ é verdade, então B é verdade.

```
parent(kim, holly).  
parent(margaret, kim).  
parent(margaret, kent).  
parent(esther, margaret).  
parent(herbert, margaret).  
parent(herbert, jean).
```

```
bisavo(GGP, GGC) :-  
    parent(GGP, GP), parent(GP, P), parent(P, GGC).
```

```
ancestor(X, Y) :- parent(X, Y).  
ancestor(X, Y) :- parent(Z, Y), ancestor(X, Z).
```

- O que produzirá
bisavo(X, Y) ?

Um problema NP-completo

Dada uma lista L de números inteiros, existe uma sublista S cuja soma seja N?

```
sum([], 0).  
sum([Head|Tail], X) :-  
    sum(Tail, TailSum),  
    X is Head + TailSum.
```

```
subList([], []).  
subList([H|T], [H|R]) :- subList(T, R).  
subList([_|T], R) :- subList(T, R).
```

```
intSum(L, N, S) :- subList(L, S), sumList(S, N).
```

Por que saber mais sobre LPs?

- Porque elas estão aí!
- Algumas disputas são fascinantes.
- A história delas é incrível.
- Diferentes problemas pedem diferentes soluções.

