# Linguagens de Programação

UNIVERSIDADE FEDERAL FLUMINENSE
PREFEITURA MUNICIPAL DE RIO DAS OSTRAS

## Linguagens de Script

Carlos Bazilio
carlosbazilio@id.uff.br
http://www.ic.uff.br/~bazilio/cursos/lp

# Motivação

- Linguagens de programação, usualmente, são projetadas para criação de aplicações completas e auto-contidas
- Sua estruturação visa eficiência em compilação e na execução dos seus programas
- Linguagens de script, por sua vez, estão voltadas para tarefas em menor escala:
  - Criação de rotinas de execução corriqueiras (macros)
  - Integração entre aplicações existentes: a saída de uma aplicação servir de entrada para outra

# Exemplo de Script

- Exemplo que combina shell script e uma aplicação em Scala

*– Conteúdo do arquivo teste.sh*

```
for f in *
do
      if [ ${f#*.} == "scala" ]; then
            scala arquivos.scala $f;
      fi
done
```

*// Conteúdo do arquivo arquivos.scala*

```
import scala.io.Source

object Test {
   def main(args: Array[String]) {
      Source.fromFile(args(0)).foreach{
                  print
      }
   }
}
```

```
$ ./teste.sh
```

# Motivação

- Linguagens script são mais flexíveis e permitem a criação de programas através da combinação de componentes já existentes
- São, às vezes, chamadas de *glue languages*, pois permitem a junção de programas menores para se construir um grande sistema
- Também são vistas como linguagens de extensão, pois permitem que algumas ferramentas sejam estendidas com estas linguagens

# Exemplos de Linguagens de Script

- Propósito geral
  - TCL, Python, Lua, Ruby, Perl, VBScript (Windows), AppleScript (Mac), ...
- Propósito específico
  - PHP, ASP, JSP, CGI, JavaScript, Awk, APL, …
- As linguagens script tiveram como ancestrais
  - Os interpretadores de comandos (*shells*) de ambientes computacionais (terminais e em lote)
  - Ferramentas para processamento de texto e geração de relatórios

# Características Comuns

- Uso interativo e em lote
  - Poucas linguagens de script possuem um compilador que lê o código inteiro antes de executá-lo (Perl é um exemplo). A maioria realiza esta tarefa linha a linha
- Economia de expressões

```
// Java
public class Principal {
    public static void main(String[] args) {
      System.out.println(“Alô Mundo!”);
    }
}
```

```
// Perl, Python, Ruby, Lua ...

print “Alô Mundo!”\n”
```

# Características Comuns

- Ausência de declaração, regras de escopo simples
  - A maioria destas linguagens dispensam a declaração de variáveis.
  - Quanto ao escopo, são adotadas regras simples. Perl, por exemplo, tudo é global por padrão, mas há como limitar o escopo. Em PHP, é exatamente o contrário. Em Python, as variáveis são locais ao bloco onde são atribuídas.

# Características Comuns

- Tipagem
  - A maioria possui tipagem dinâmica flexível. Em algumas linguagens o tipo de uma variável é checado imediatamente antes do seu uso (PHP, Python, Ruby, Scheme, ...). Em outras, a variável será interpretada dependendo do contexto onde aparece

```
// Em Lua, Perl ...

a = "4"
print a . 3 . "\n"; // . é concatenação
print a + 3 . "\n" // + é adição
```

```
// Saída

43
7
```

# Características Comuns

- Facilidade de Acesso à Funcionalidades do Sistema
  - Funcionalidades de sistema são muito mais facilmente acessadas por estas linguages (em comparação com linguagens de programação tradicionais, como C): entrada e saída, manipulação de arquivos, gerenciamento de processos, proteção e autorização, relógio, acesso à rede, etc.
- Manipulação de strings sofisticada
  - Normalmente baseada em expressões regulares
  - Herança das linguagens de processamento de texto

# Características Comuns

- Tipos de dados de alto-nível
  - Estruturas como conjuntos, dicionários (mapas), são extremamente comuns. Várias são definidas usando a própria linguagem. Em todas há a implementação de coleta automática de lixo

# Domínios
# Linguagens Shell

- Fundamentalmente voltadas para manipulação do sistema de arquivos
- Possuem comandos simples, como a listagem de um diretório: ls (dir)
  - Variações com o uso de caracteres coringa (? e *), paginação (more), listagem de atributos, etc
- Possuem comandos compostos como for
- Permitem redirecionamento da saída ( > )
- Permitem o encadeamento de comandos - *pipe* ( | )

# Domínios
# Linguagens Shell

- Exemplos:
  - grep main *.c > arquivos-principais.txt
  - grep main *.c | wc -l

# Domínios
# Processamento de Texto

- Linguagens exemplo
  - Sed, Awk (Aho, Weinberger e Kernighan) e Perl
- Permitem a criação de scripts de manipulação de arquivos texto
- Suponha a tarefa de extração de cabeçalhos de uma página html (<h1></h1>, <h2></h2>, etc)
- É possível a utilização de um editor de texto comum para tal. Entretanto, certamente será tedioso, pois este procedimento não é geral

# Domínios
# Matemática e Estatística

- Sed e Awk estão para processamento de texto, assim como APL está para expressões matemáticas
- Sucessores desta são os pacotes matemáticos comerciais Maple, Mathematica e Matlab
- Similarmente, para estatística temos as linguagens S e R

# Domínios WWW

- Podem ser divididas em scripts que executam no servidor ou no cliente
- Exemplos
  - Servidor: CGI, Perl, PHP, ASP, JSP, …
  - Cliente: JavaScript
- Applets são programas Java hospedados num servidor e que são executados num cliente