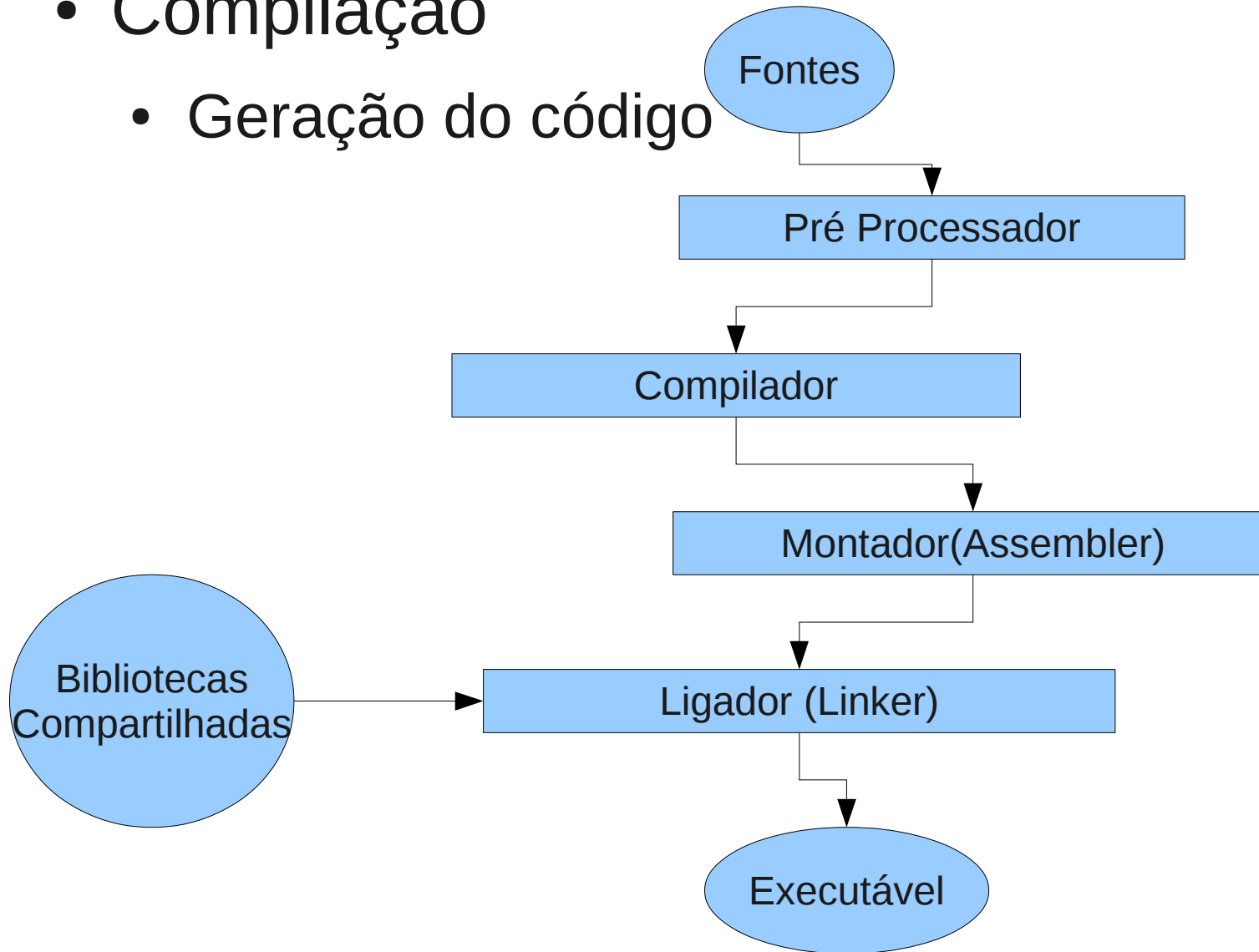


105 – Shell Script

João Bosco Teixeira Junior

Compilação x interpretação

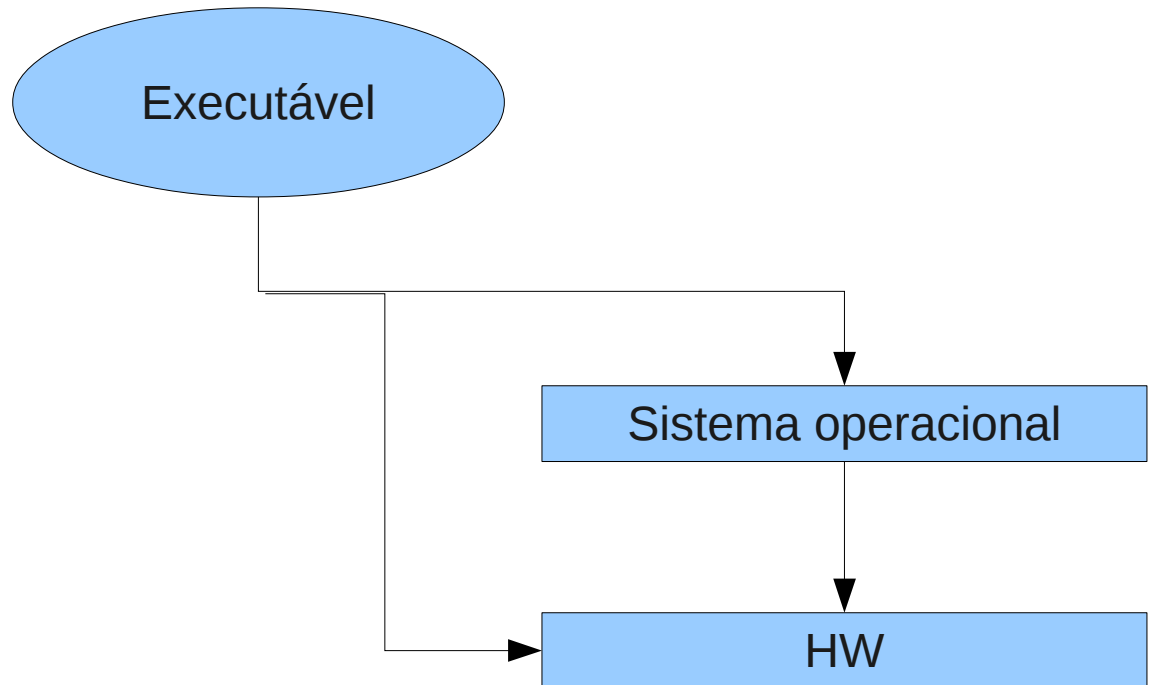
- Compilação
 - Geração do código



Compilação x interpretação

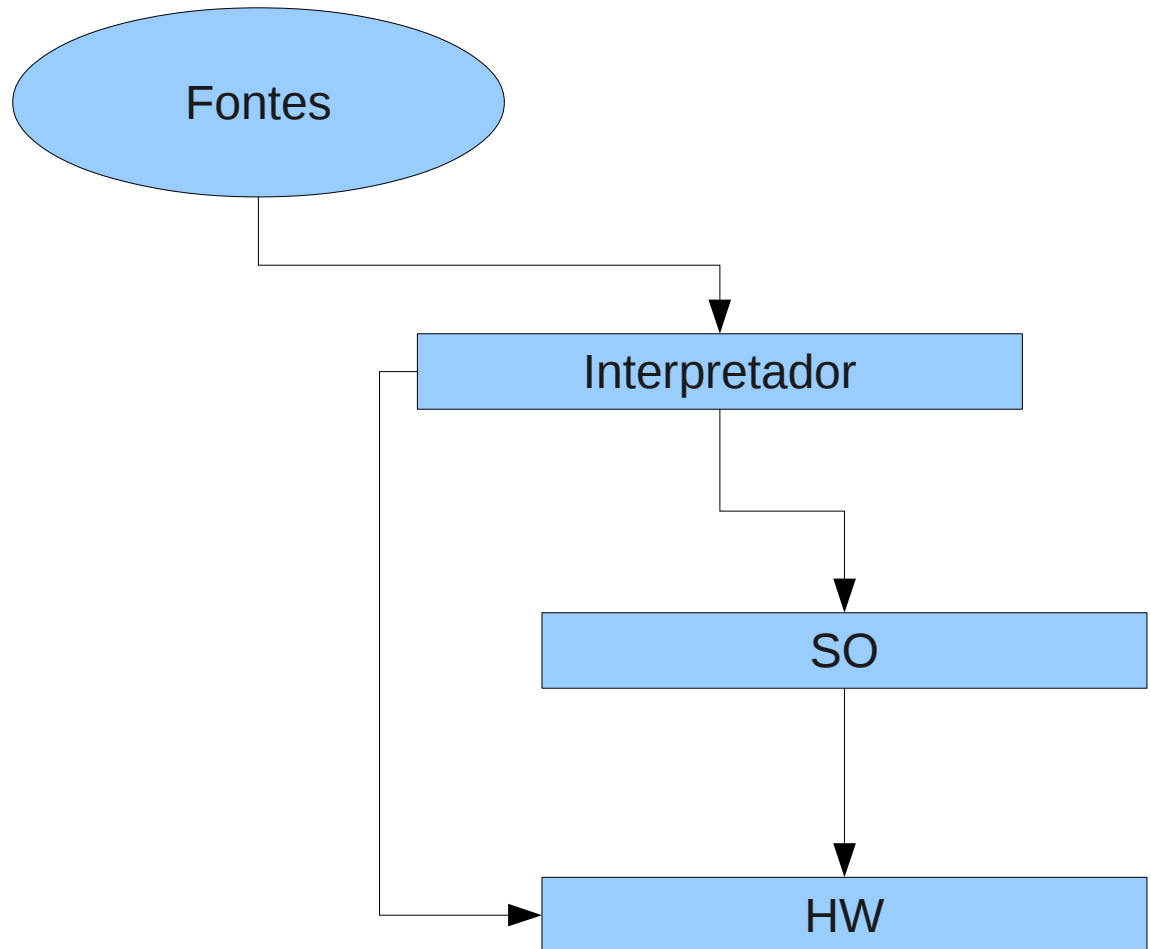
(cont...)

- Compilação
 - Execução



Compilação x interpretação (cont...)

- Interpretação
 - Execução



Scripts - Vantagens

- Não precisa compilar ;)
- Orientado a comandos (Bash)
- Todo SO tem uma linguagem !

Scripts

- Interpretado
 - Interpretador de comandos
- Interativos
 - Requer resposta
- Não interativos
 - Rodam sem intervenção
 - Podem ser agendados. Ex.:
 - Backup
 - Monitoramento (Email, SMS)
 - Instalação automática de aplicativos

Shell Scripting

- Interpretadores de Comandos
 - Tcsh
 - Ash
 - Csh
 - Bsh
 - Dash
 - **Bash**

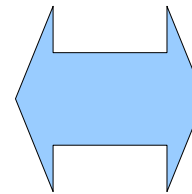
Shell Scripting (cont...)

- Bash
 - Projeto Livre
 - Padrão na maioria das distribuições Linux
 - Linguagem de Programação embutida
- Conceitos
 - Interpretador de Comandos
 - Prompt
 - Working Directory
 - Aliases
 - Variáveis de Ambiente
 - Comandos Embutidos

Shell Scripting (cont...)

- Prompt do Interpretador de comando
 - Formato
 - [`aluno@debian`: Diretório_corrente]\$
 - usuário@maquina diretório corrente
 - Acesso após o login
 - Usado para fazer requisições ao SO

- Criar ou remover arquivos
- Acessar um recurso da rede
- Criar ou remover usuário
- Iniciar um aplicativo



Criar
processos
diversos

Shell Scripting (cont...)

- Working directory
 - Diretório de Trabalho
 - Usado em caso de omissão do caminho
 - Comando:
 - pwd

```
aluno@debian:~ $ pwd  
/home/aluno
```

Shell Scripting (cont...)

- Aliases (Apelido)
 - Variáveis especiais criadas para substituir comandos
 - Comando
 - alias copy = 'cp'

```
aluno@debian:~ $ copy a.txt b.txt
bash: copy: comando não encontrado
aluno@debian:~ $ alias copy = 'cp'
aluno@debian:~ $ copy a.txt b.txt
aluno@debian:~ $
```

Shell Scripting (cont...)

- Variáveis de Ambiente
 - Variáveis usadas para fornecer um profile ao usuário.
 - Ajustadas normalmente em tempo de login
 - Arquivos
 - /etc/bashrc
 - ~/.bashrc
 - /etc/profile
 - ~/.bash_profile

```
aluno@debian:~ $ echo $PATH
:/bin/:/usr/bin/:/usr/local/bin
aluno@debian:~ $ echo $PWD
/home/aluno
```

Shell Scripting (cont...)

- Estrutura de um script
 - Definição: Sequência de comandos em um arquivo texto

Interpretador ↔
Comentário ↔
Comando ↔

```
#!/bin/bash  
# Isso é um comentário  
echo "isso é um teste"
```

Shell Script (Cont...)

- Interpretador de Comandos
 - Interface entre o usuário e o SO
 - Sessão
 - Preferencias do usuário
 - Informações
 - Acesso a variáveis de ambiente

```
aluno@teste:~$ echo $USER
aluno
aluno@teste:~$ TMOUT=10
aluno@teste:~$ echo $TMOUT
10
```



Shell Script

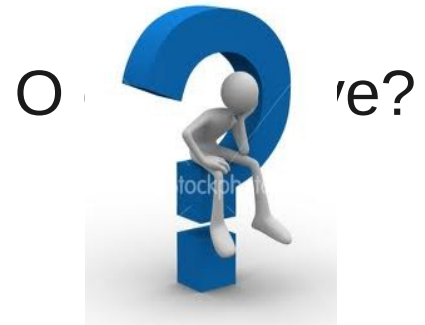
- Estrutura de script

```
aluno@debian# pico script.sh
```

Interpretador ↔
Comentário ↔
Comando ↔

```
#!/bin/bash  
# Isso é um comentário  
echo "isso é um teste"
```

```
aluno@debian# script.sh
```



Shell Script

- Lendo entrada do teclado

```
#!/bin/bash
echo "entre um número"
read VALOR
echo o valor foi $VALOR
```



```
aluno@debian# ./script.sh
```

- Atividade: Faça um script que leia do teclado o seu nome e depois e seu sobrenome, depois imprima: ola [nome] [sobrenome]
- Tempo 5 min

Shell Script

- Passando parâmetros para um script

```
aluno@debian# ./script par1 par2 ... parN
```

\$0

\$1

\$2

\$N

\$* ou \$@

```
#!/bin/sh
```

```
echo comando: $0
```

```
echo Parametro 1: $1
```

```
echo Parametro 2: $2
```

```
echo no total foram passados $# parametros
```

```
aluno@debian# chmod u+x ./script2.sh
```

```
aluno@debian# ./script2.sh
```

Shell Script

- Expressões Aritméticas

```
aluno@debian# A=10  
aluno@debian# B=20
```

- Formas de avaliação

- expr
- let
- $\$(())$

```
aluno@debian# expr $A + $B  
aluno@debian# let C="$A+$B"  
aluno@debian# echo $C  
aluno@debian# echo $(( $A+$B ))
```

Shell Script

- Variáveis Revisitadas
 - Direcionando a saída de um comando para uma variável

```
aluno@debian# A=`ls`  
aluno@debian# A=$(ls)
```

Shell Script

- Fazendo testes com números, textos e arquivos
 - Comando: test

```
$ test 0 -lt 1
```

```
$ test -z $VAR
```

```
$ test -x /etc/hosts
```

	Testes em variáveis
-lt	Núm. é menor que (LessThan)
-gt	Núm. é maior que (GreaterThan)
-le	Núm. é menor igual (LessEqual)
-ge	Núm. é maior igual (GreaterEqual)
-eq	Núm. é igual (Equal)
-ne	Núm. é diferente (NotEqual)
=	String é igual
!=	String é diferente
-n	String é não nula
-z	String é nula

	Testes em arquivos
-d	É um diretório
-f	É um arquivo normal
-r	O arquivo tem permissão de leitura
-s	O tamanho do arquivo é maior que zero
-w	O arquivo tem permissão de escrita
-nt	O arquivo é mais recente (NewerThan)
-ot	O arquivo é mais antigo (OlderThan)
-ef	O arquivo é o mesmo (EqualFile)
-a	E lógico (AND)
-o	OU lógico (OR)

Shell Script

- Fazendo testes com números, textos e arquivos
 - Atalho pro comando test

```
$ [ 0 -lt 1 ]
```

```
$ [ -z $VAR ]
```

```
$ [ -x /etc/hosts ]
```

Obs.: Sempre deixar espaços depois do “[“ e antes de “]”

Shell Script

- Tomada de decisão

- Comando: if

```
if comando
then
    ...
elif comando
then
    ...
else
    ...
fi
```

- Condição

- Teste ex: `if [-x /bin/ls] then ... ; fi`

- Comando ex: `if ls ; then ... ; fi`

Shell Script

- Tomada de decisão
 - Comando case

```
case "$VAR" in
    Valor1)
        ...
        ;;

    Valor2)
        ...
        ;;
    *)
        ...
        ;;
esac
```

Shell Script

- Laços
 - Comando for
 - Percorre uma lista pegando os elementos um a um

```
for VAR in LISTA
do
...
comandos
...
done
```

- Comando seq
 - Gera a lista em caso de necessidade

```
for numero in seq 1 10
do
    echo "Contando: $numero"
done
```


Shell Script

- IFS - Internal Field Separator
 - Variável de Ambiente que define o caractere de separação dos campos

```
#set | grep IFS
```

Shell Script

- Laços
 - Comando while

```
While comando
do
...
comandos
...
done
```

- A mesma coisa que foi feita com o comando for

```
i=0
while test $i -le 10
do
    i=$((i+1))
    echo "Contando: $i"
done
```

Shell Script

- Funções
 - Por que usar?
 - Modularização
 - “Reuso”
- Declaração

```
nome_da_funcao() {  
    ...  
    Comandos;  
    ...  
}
```

Shell Script

- Funções (Cont...)
 - Passagem de parâmetros

```
...  
nome_da_funcao par1 par2 par3  
...
```

- Declaração

```
nome_da_funcao() {  
    ...  
    Comando $1  
    Comando $2 $3  
    ...  
}
```