

# Programação I

## Comandos de repetição

André Grégio, Fabiano Silva, Luiz Albini e Marcos Castilho

DInf UFPR, Curitiba PR

23 de julho de 2020

# Objetivos

- Apresentar os comandos de controle de fluxo em scripts
  - Desvios condicionais
  - (\*) Repetições
  - Seleção

# Repetições

- 2 tipos de *for*
- *while*
- *until*

## FOR, primeira forma

- Permite iterar sobre uma lista fixa de valores, é diferente do for do *Pascal*

```
1 for name [in lista]
2 do
3     list
4 done
```

ou se colocar em uma única linha:

```
1 for name [in lista] ; do list ; done
```

## FOR, segunda forma

- Parecido com a linguagem C

```

1 for (( expr1 ; expr2 ; expr3 ))
2 do
3     list
4 done
  
```

ou se colocar em uma única linha:

```

1 for (( expr1 ; expr2 ; expr3 )) ; do list ; done
  
```

# Exemplo

```
IFS=:
```

```
for dir in $PATH  
do  
    ls -ld $dir  
done
```

## Exemplo 2

```
for user in $(ls /home/bcc)
do
    finger $user | grep Name
done
```

## Exemplo 3

```
for i in 1 2 3 4 5  
do  
    mkdir teste_$(i)  
done
```

# Exemplo

```
for ((i=1 ; i<=10 ; i++))  
do  
    echo $i  
done
```

# WHILE

```
while condicao  
do  
    sentencas  
done
```

- *while* é útil quando combinado com aritmética de inteiros, entrada e saída de variáveis e processamento de linhas de comando
- *condição* é uma lista de comandos, o status de saída do último é utilizado para determinar a saída do laço
- executa enquanto a condição é verdadeira (zero)

# UNTIL

```
until condicao  
do  
    sentencas  
done
```

- mesmo princípio do *while*, mas o teste é ao contrário
- executa até que a condição seja verdadeira (zero)
- ao contrário de *Pascal* o teste é feito no início do laço

## Exemplo de WHILE

```
path=$PATH:

while [ $path ]; do
    ls -ld ${path%*:}
    path=${path#*:}
done
```

- copia \$PATH e adiciona um :
- faz ls no diretório
- remove o primeiro diretório da lista

## Exemplo de UNTIL

```
until cp $1 $2; do  
    echo 'tentativa de copia falou, esperando...'  
    sleep 5  
done
```

- tenta copiar um arquivo em outro até dar certo

## Mesma coisa com WHILE

```
while ! cp $1 $2; do
    echo 'tentativa de copia falou, esperando...'
    sleep 5
done
```

- Portanto, *until* é desnecessário...

# Licença

- Slides feitos em  $\text{\LaTeX}$  usando beamer
- Licença

*Creative Commons* Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>