

SEMINÁRIOS I

LINUX: SHELL SCRIPT

LINUX: AGENDAMENTO DE TAREFAS

Profs. Alexei Machado e João Caram

Shell script

2

- No Linux quase todas as tarefas podem ser automatizadas
- Podemos escrever arquivos de *script* que realizam ações em lote/sequência para não ser necessário repetir várias sequências de comandos

Shell script

3

- Um *script* é simplesmente um texto, um roteiro ou plano de ações
- Para o *shell*, um *script* é uma sequência de comandos que ele vai interpretar e executar
- Uso do *shebang* (`#!`) para identificação
- Ex: *bomdia.sh*, *bomdia2.sh*

Shell script


4

- O shell, na verdade, define um ambiente de programação com as características mais comuns de qualquer linguagem
 - Variáveis
 - Parâmetros
 - Estruturas de repetição e decisão
 - ...

Variáveis

5

- ❑ Nomes iniciados com letra ou *underscore*
- ❑ Atribuição direta
- ❑ Leitura a partir do teclado
- ❑ Uso precedido por \$
- ❑ Ex: *var.sh* e *apagaTemp.sh*



Atenção
para a
sintaxe!!!

Parâmetros

6

- Todo linha de comando tem pelo menos um parâmetro (o comando executado)
- Podemos verificar a *quantidade* de parâmetros, o comando e cada parâmetro passado.
- Ex: *param.sh* e *soma.sh*

Estrutura de decisão: if

7

```
if [ teste ]  
then  
    comandos  
else  
    comandos  
fi
```

□ Ex: *soma2.sh* e *temRC.sh*

Estrutura de decisão - if

8

- ❑ -lt: é menor que (less than)
- ❑ -gt: é maior que (greater than)
- ❑ -le: é menor ou igual (less or equal)
- ❑ -ge: é maior ou igual (greater or equal)
- ❑ -eq: é igual (equal)
- ❑ -ne: é diferente (not equal)
- ❑ =: String é igual
- ❑ !=: String é diferente
- ❑ -n: String é não nula
- ❑ -z: String é nula

Estrutura de decisão - if

9

- -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 (newer than)
- -ot: O arquivo é mais antigo (older than)
- -ef: O arquivo é o mesmo (equal file)
- -a: E lógico (AND)
- -o: OU lógico (OR)

Estrutura de repetição - for

10

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

Estrutura de repetição - for

11

- A lista pode ser composta de:
 - Parâmetros
 - Resultado de execução de um comando
 - Redirecionamento de entrada

- Ex: *for1.sh* e *for2.sh*

Estrutura de repetição - while

12

```
while [ teste ]  
do  
    comandos  
done
```

□ Ex: *guess.sh* e *guess2.sh*

Agendamento de tarefas

13

- Sistemas Linux permitem que
 - Ações sejam configuradas para executar a cada login
 - Ações sejam configuradas para executar a cada abertura de terminal
 - Ações sejam configuradas para executar periodicamente, esteja o usuário logado ou não

Configuração do perfil - login

14

- `/etc/profile`: global
- `~/.bash_profile`: pessoal
- `~/.bash_login`: caso não exista o *profile*
- `~/.profile`: caso não existam o *profile* e o *login*

Configuração do perfil - sessão

15

- `/etc/bash.bashrc`: global
- `~/.bashrc`: pessoal

Agendamento de tarefas

16

- *Cron e crontab*
- Opções do crontab
 - -l
 - -r
 - -e

Crontab

17

- Uma linha do crontab:

```
30 2 * * 7 ~/rmTmp.sh
```

Crontab

18

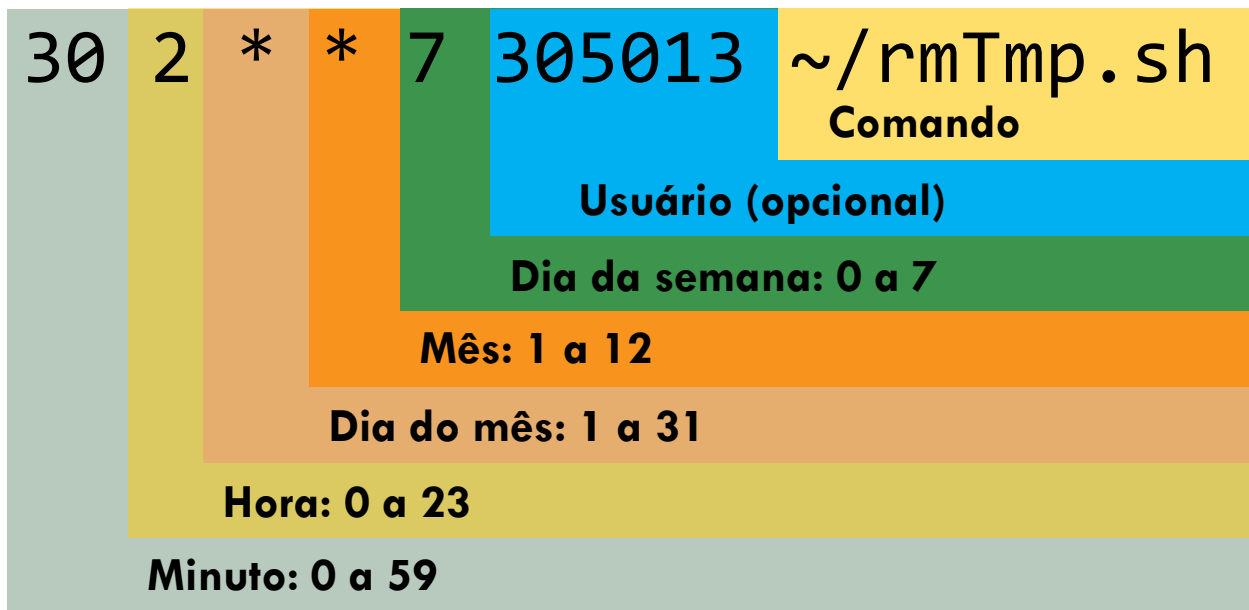
- Uma linha do crontab:

```
30 2 * * 7 305013 ~/rmTmp.sh
```

Crontab

19

- Uma linha do crontab:



OBRIGADO.

Dúvidas?