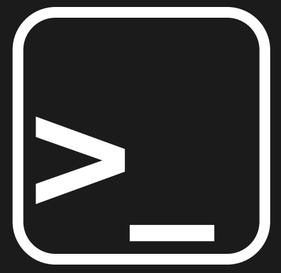
IFPB - REDES DE COMPUTADORES

//01



Shell Script.sh

Ayrton Macaulay
Bruno Ribeiro
Isa Polyana

Programação de Script - 2020.2

Sumário

O que é Shell Script

Shell scripting é uma ótima forma de automatizar tarefas repetitivas e pode poupar muito tempo quando se é um desenvolvedor.

SHELL

Todo script deve fazer uso de um interpretador, o mais comum é o Bash (que acompanha a maioria das distribuições Linux atuais).

PRATICIDADE

Qualquer comando possível de ser executado no terminal poderá ser usado em um shell script, inclusive comandos de programas externos (instalados na máquina) ou até mesmo outros scripts criados pelo usuário.

USABILIDADE

É comum a criação de scripts para quando utilizamos determinadas sequências de comandos com frequência.

Também acompanha muitos programas de código aberto para facilitar a instalação nas máquinas dos usuários.

Exemplos de uso

//04

> MONITORAMENTO_

- Script para monitorar um servidor Apache (Alura)
- Monitoramento do sistema Linux (BASHTOP)

> AUTOMATIZAR INSTALAÇÕES_

- Instalar e configurar NS-3 (ns3install)
- Instalar e configurar o Docker (getdocker)

> FACILITAR TAREFAS RECORRENTES_

- Script que copia arquivos de uma pasta para outra e dá um commit para o git.

```
#!/bin/bash
#
# This is a scripz file based by https://github.com/wyllianbs/ns3install
.
# It has been modified to download and build the (at the time) newest
# version of NS3 in a non-interactive manner
#
# Federal University of Santa Catarina - UFSC
# (c) Prof. Wyllian Bezerra da Silva <wyllian.bs@ufsc.br>
#
# Check out README.md for more details available at URL:
# https://github.com/wyllianbs/ns3install

export DEBIAN_FRONTEND=noninteractive

echo "Installing all dependencies for Debian-based Distro (via apt)"
sudo apt-get update;
sudo -E apt-get install -y g++ python3 python3-dev pkg-config sqlite3 git
 python3-setuptools qt5-default mercurial gdb valgrind gir1.2-goocanvas-2
.0 python-gi python-gi-cairo python3-gi python3-gi-cairo python3-pygraphv
iz gir1.2-gtk-3.0 ipython3 openmpi-bin openmpi-common openmpi-doc libopen
mpi-dev autoconf cvs bzr unrar uncrustify doxygen graphviz imagemagick te
xlive texlive-extra-utils texlive-latex-extra texlive-font-utils dvipng l
atexmk python3-sphinx dia  gsl-bin libgsl-dev libgsl23 libgslcblas0 tcpdu
mp sqlite sqlite3 libsqlite3-dev libxml2 libxml2-dev cmake libc6-dev libc
6-dev-i386 libclang-6.0-dev llvm-6.0-dev automake python3-pip libgtk-3-de
v vtun lxc uml-utilities libboost-dev libboost-filesystem-dev wget libyam
l-cpp-dev vim > apt.log 2>&1
python3 -m pip install --user cxxfilt > cxxfilt.log 2>&1
python3 -m pip install pyyaml > pyyaml.log 2>&1
python3 -m pip install nympy > numpy.log 2>&1
python3 -m pip install matplotlib > matplotlib.log 2>&1

# Ns3 Folder
URLfile="https://www.nsnam.org/releases/ns-allinone-3.32.tar.bz2"
file=$(echo "${URLfile##*/}")
NSallinonedir=$(echo "${file%.*.*}");
NSversion=$(echo "${NSallinonedir##*-}");
NSdir=ns-$NSversion;

echo -e "\n::: Downloading the NS3 version: \"$URLfile\"...";
wget $URLfile ;
```

Script para automatizar a instalação do NS-3 e de suas dependências

> Bash é a shell do 'GNU Project'—a Bourne Again Shell. É uma shell compatível com a antiga Bourne shell que incorpora recursos da Korn shell (ksh) e da C shell (csh).
As melhorias oferecidas pelo Bash incluem:

- *Edição em linha de comando,*
- *Tamanho ilimitado do histórico de comandos,*
- *Controle de processos,*
- *Funções shell e apelidos,*
- *Arrays indexados de tamanho ilimitado,*
- *Aritmética entre inteiros de qualquer base de 2 até 64._*

- GNU.org

#!/bin/bash

//05

BOURNE-AGAIN SHELL

> Em qualquer script shell é necessário, antes de qualquer coisa, definir qual interpretador será utilizado para ler e executar os comandos escritos no arquivo de script.
Por isso é comum haver a expressão ***#!/bin/bash*** na primeira linha, indicando que o arquivo deve ser executado pelo programa ***bash*** encontrado dentro da pasta ***/bin***._

› cat /etc/shells_

```
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
```

Checar quais shells estão instaladas no sistema.

Sintaxe Básica

> COMENTÁRIOS_

```
# Comentários são indicados
# pelo caractere
# de jogo da velha!
```

> EXEMPLO DE COMANDO_

```
echo "Opa meu bom, como vai?"
echo "Tô bem demais e tu?"
echo "Melhor agora!"
```

> VARIÁVEIS_

```
videncia="Flamengo vai ganhar
do Inter"
echo $videncia
read gols_flamengo
read gols_inter
echo "Placar: $gols_flamengo
x $gols_inter"
```

> FOR LOOP_

```
for i in 1 2 3 4 5 6 7 8 9 10
do
   expr $i \* 2
done
```

> CONDICIONAIS (CASE)_

```
read a
case $a in
   teste)
      echo "-1"
      ;;
   *)
      echo "1"
      ;;
esac
```

> WHILE LOOP_

```
a=0
while [ "$a" -lt 10 ]
do
   echo $a
   a=$((a+1))
done
```

> CONDICIONAIS (IF)_

```
read -p "Adivinha o número! " a
if [ "$a" -eq 10 ]
then
   echo "Yay"
else
   echo "Sem Yays pra você"
fi
```

Fish

FRIENDLY INTERACTIVE SHELL

> Fish é uma shell de linha de comando (como o bash ou zsh) que é inteligente e amigável. Fish tem suporte para incríveis ferramentas como syntax higlighting, auto-sugestão e auto-completar que funcionam desde o zero, sem nada para aprender ou configurar._

- fishshell.com

> Fish é uma shell com a proposta de ser mais amigável e de uso simplificado. Dando suporte para cores de 24bit, possui uma interface extremamente interativa e de fácil entendimento. Além de ter facilidades como uma sintaxe mais enxuta com relação ao bash e auíxilios na digitação de comandos._

› cat /etc/shells_

```
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
/usr/bin/fish
```

Fish shell está instalada no sistema

Sintaxe Básica

> COMENTÁRIOS_

```
# Comentários são indicados
# pelo caractere
# de jogo da velha!
```

> EXEMPLO DE COMANDO_

```
echo "Opa meu bom, como vai?"
echo "Tô bem demais e tu?"
echo "Melhor agora!"
```

> VARIÁVEIS_

```
set videncia "Flamengo vai
ganhar do Inter"
echo $videncia
read gols_flamengo
read gols_inter
echo "Placar: $gols_flamengo
x $gols_inter"
```

> FOR LOOP_

```
for i in 1 2 3 4 5 6 7 8 9 10
   expr $i \* 2
end
```

> CONDICIONAIS (SWITCH)_

```
read a
switch $a
   case teste
      echo "-1"
   case '*'
      echo "1"
end
```

> WHILE LOOP_

```
set a 0
while test "$a" -lt 10
   echo $a
   set a (math $a + 1)
end
```

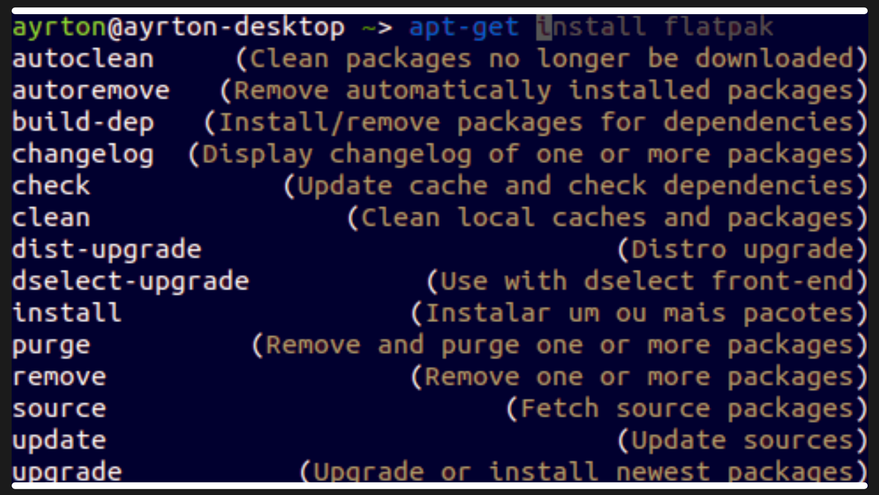
> CONDICIONAIS (IF)_

```
read -P "Adivinha o número! " a
if test "$a" -eq 10
   echo "Yay"
else
   echo "Sem Yays pra você"
end
```

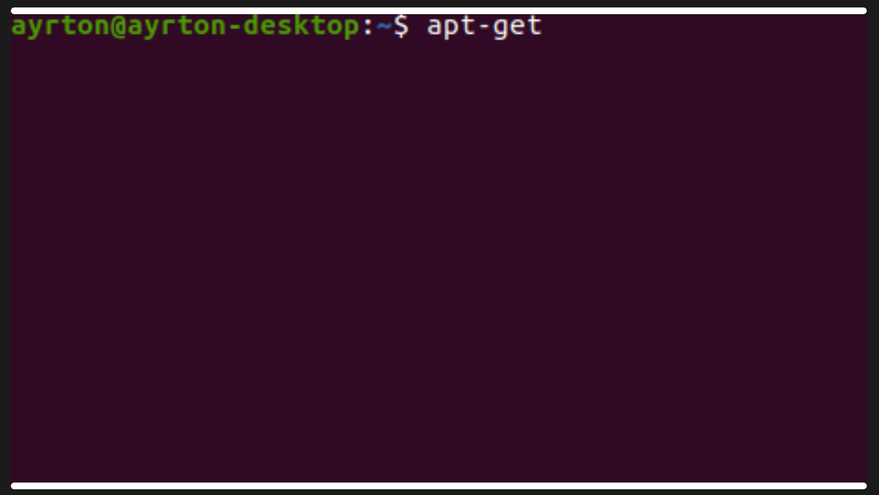
Fish vs Bash

Auto-completar e Auto-sugestão

> Fish sugere comandos enquanto o usuário digita o que deseja, mostrando as sugestões ao lado direito do cursor.
As sugestões vão desde caminhos de arquivos e pastas até opções e histórico de comandos.
Também é possível ver uma lista de sugestões para completar a linha que está sendo digitada apertando a tecla TAB._



fish

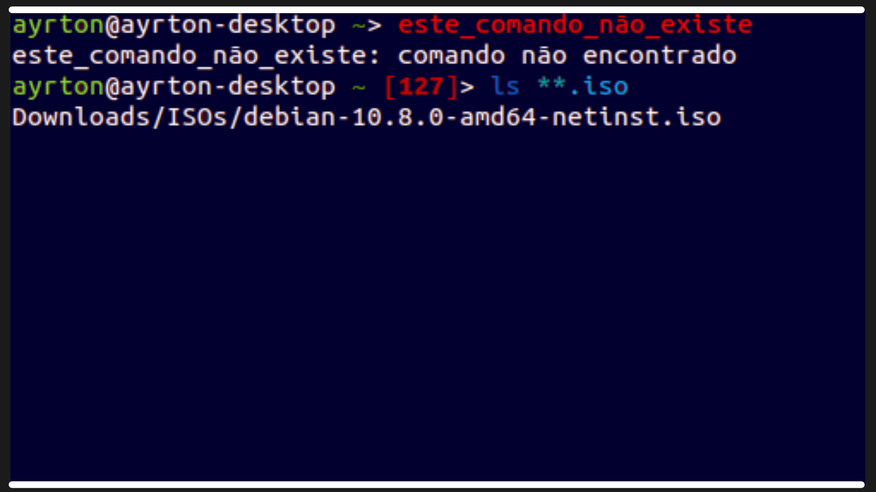


bash

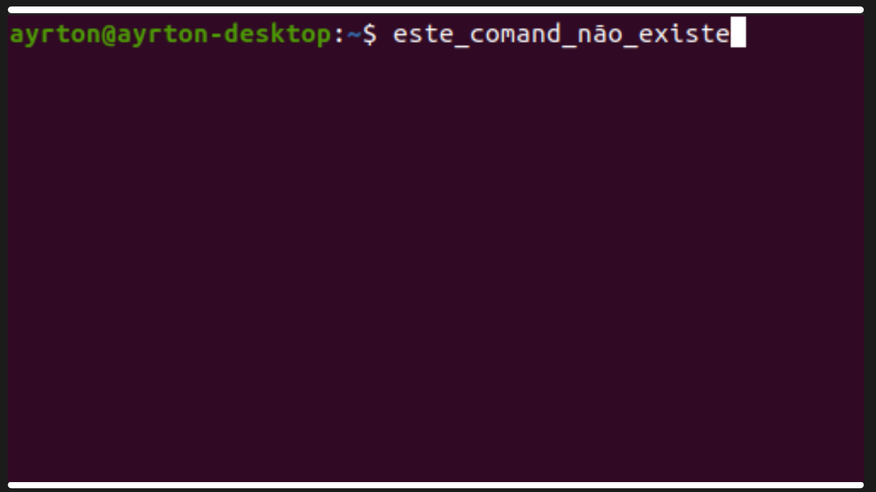
Fish vs Bash

Syntax Highlighting e Recursive Wildcard

> Fish possui a funcionalidade de indicar quando um comando está errado ou simplesmente não existe, identificando-o com a cor vermelha.
Além disso, um outro recurso interessante é o wildcard recursivo ' ** ' que itera recursivamente por todos os arquivos e diretórios do caminho especificado._



fish



bash

Programação de Script - 2020.2

Referências

Shell Scripting Tutorial

SHELLSCRIPT.SH

Codecademy

CODECADEMY.COM/LEARN/BASH-SCRIPTING

Terminal Root

YOUTU.BE/UCLMJMERXXU

GNU Bash

GNU.ORG/SOFTWARE/BASH

Fish Shell Docs

FISHSHELL.COM/DOCS/CURRENT