Introdução ao GNU/Linux

Curso de introdução ao GNU/Linux
Abordagem Prática para HPC

Aula 04

Programação bash

Linha de comando e script

- O bash interpreta comandos em linha de comando e salvos em arquivos similarmente.
- echo Hello

Hello

Linha de comando e script

No arquivo hello.sh

#!/bin/bash

echo Hello

Verificar permissão de execução e PATH (ou executar passando o caminho)

Variáveis

- Para utilizar variáveis no script, apenas atribua um valor a ela
  - #!/bin/bash
  - VAR=valor
  - echo VAR = $VAR
- Variáveis internas do bash
  - Parâmetros: $0, $1, $2 ….
  - Todos os parâmetros: $@
  - Quantidade de parâmetros: $#

Variáveis

- #!/bin/bash
- echo Quantidade de parametros $#
- echo Todos os parametros “$@”
- echo Parametro0 $0
- echo Parametro1 $1
- echo Parametro2 $2
- echo Parametro3 $3
- echo Parametro4 $4

Variáveis

- $? resultado do último comando
- ls
- echo $?
- ls zzzz
- echo $?
- $! PID do processo em background
- $IFS separador de linhas

Variáveis

- Variáveis
  - echo $var
  - echo ${var}
- Operações matemáticas
  - let x=10
  - let y=$x*2+1
  - echo $x e $y
  - c=$(($x*$y))
  - echo $c

Comando interação

- Loop for executa comandos percorrendo lista de itens separado por espaço (padrão $IFS)
- for arg in [list]
- do
- comandos ; comandos ;
- done
- help for

Comando interação

- Separador de variáveis é o IFS
- Arquivos
  - for var in “*”
- Parâmetros
  - for var in “$@”
- Variáveis
  - for var in $PATH (Tem que alterar $IFS)
- Sub-shell
  - for var in $(comando)

Condições de testes

- Condições de testes no bash
  - man test
  - info test
  - help [
  - [-z “$VAR”]
  - [“$VAR” = “texto”]
  - [“$VAR” -eq 10]
  - [-f “$arquivo”]
  - (($VAR == 20))
  - (($# > 3))

Comando if

- help if
- if [teste]
- then …
- elif [teste]
- then ...
- else
- fi

Comando while

- help while
- while: while COMMANDS; do COMMANDS; done
- #!/bin/bash
- while [teste]
- do
- comandos
- done

Dicas de Bash

- Ctrl+X Ctrl+E Editor de linha de comando. Configurar $EDITOR
- var=$(comando)
- comando $(comando)
- VAR=valor comando
- VAR=valor ; comando
- comando_para_shell ; echo !! >> script.sh (!n)
- comando arg ; comando $_ (ou primeiro !^, ou !comando:n)

Exercícios

1) Escreva um programa parametros.sh que execute o comando:
./parametros.sh -in entrada.txt -out saida.txt

Internamente cada parâmetro referenciar como $1,$2,$3 e $4

O programa executa o cat no parâmetro entrada e redireciona para saída

Considerar os parâmetros fixos $2 como entrada e $4 como saída

Exercícios

2) Escreva um programa loop.sh para executar:

./loop.sh 10

Ele deve imprimir

Hello 1
Hello 2
Hello 3
...
Hello 10

Exercícios

3) Desafio, alterar o parametro.sh para o -in e o -out não ter ordem na linha de comando. Poder executar:

./parametro.sh -in arquivo -out arquivo
./parametro.sh -out arquivo -in arquivo

Exibir erro se não for essas duas opções

Usar comandos if
IF [$# ...]
IF [$1 = ...]