***Apostila de Shell Script***

***Arquivos Bash no Sistema:***

**/etc/profile →** Esse arquivo é executado quando o login é efetuado no sistema. Ele contém variáveis de ambiente e progamas de inicialização.

**/etc/bashrc** ou **/etc/bash.bashrc →** Arquivo de configuração incializado ao iniciar um novo terminal bash.

**~/.bash\_profile**

**~/.bash\_login**

**~/.profile →** O linux tenta carregar um desses arquivos, fazendo a tentativa na sequência apresentada, e executando o primeiro que encontrar. O arquivo é carregado após o arquivo **“/etc/profile”**. São arquivos que inicializam as variáveis de ambiente e programas progamados para inicializar após o login.

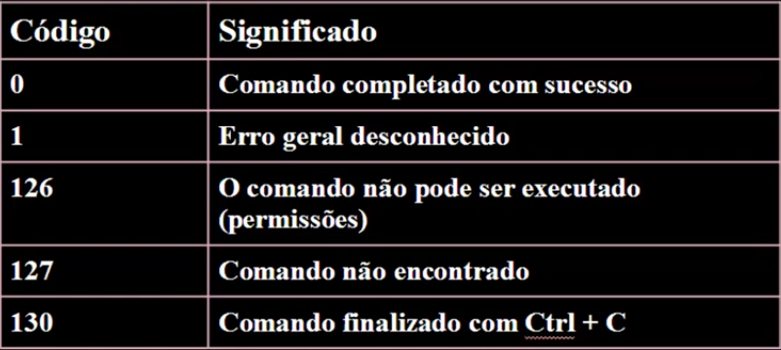
**~/.bashrc →** Esse arquivo de configuração é iniciado após iniciar uma nova sessão bash interativamente no terminal.

**~/.inputrc →** Esse arquivo é responsável por fazer as configurações do teclado a respeito das vincalações de teclas.

**~/.bash\_logout** → Esse arquivo de configuração é executado automaticamente quando é feito o logout do usuário no sistema.

***Comandos:***

No bash é possível exibir um status de saída para os comandos. Esse status é armezenado na variável “**$?**”.



**bash →** Comando utilizado para iniciar um novo processo bash interativo.

**ps →** Comando utilizado para mostrar os processos do bash que está em execução.

**ps aux →** Comando utilizado para mostrar todos os processos que estão execução no sistema, onde; **“a”** mostra os processos de todos os usuários; **“u”** mostra quem é o usuário de cada processo e mais detalhes como consumo de CPU e outros; e **“x”** que lista os processos não interativos que são processos que são iniciados ou pelo sistema na hora do boot por exemplo, ou pelo usuário ao abrir um progama como o navegador, sendo processos não iniciados pelo terminal.

**env** ou **printenv →** Utilizado para listar as variáveis de ambiente.

comando **| less →** Comando utilizado para abrir um scroll de rolagem. Geralmente utilizado na leitura de resultados que são extensos.

nome\_da\_variável**=**valor\_da\_variável → Comando utilizado para declarar uma variável local.

nome\_do\_array**=(**valor1 valor2 valor3**) →** Comando utilizado para criar um array local.

**echo →** Utilizado para imprimir informações no terminal.

**echo ${**nome\_do\_array**[\*]} →** Comando utilizado para imprimir todos os dados de um array.

**export** nome\_da\_variável → Comando utilizado para transformar uma variável local em uma variável global.

**export** nome\_do\_array\_global**=${**nome\_do\_array\_local**[\*]}**  → Comando utilizado para transformar um array local em um array global.

comando **>** nome\_do\_arquivo**.**extensão\_do\_arquivo **→** Comando utilizado para armazenar os dados de exibição de um comando em um arquivo. Geralmente usado em arquivos de log. Todo o contéudo do arquivo é substitúido.

comando **>>** nome\_do\_arquivo**.**extensão\_do\_arquivo **→** Comando utilizado para armazenar os dados de exibição de um comando em um arquivo. Geralmente usado em arquivos de log. O contéudo é acrecentado ao arquivo, mantendo os dados já salvos nele.

comando **<** nome\_do\_arquivo**.**extensão\_do\_arquivo **→** Comando utilizado para passar argumentos armazenados em um arquivo para execução de um comando.

**bc →** Comando utilizado para abrir a calculadora do bash.

**quit →** Comando utilizado para fechar a calculadora do bash.

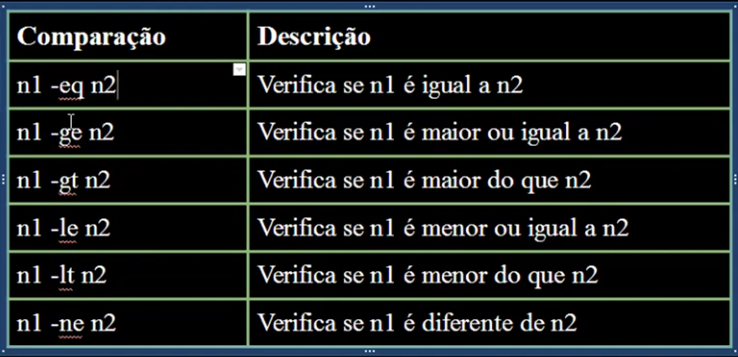
**exit** valor **→** Comando utilizado para exibir um valor no final do comando. Muito utilizado para exibir o status de retorno de um comando dentro de um script com a variável “**$?**”.

**printf “**interação\_com\_o\_usuário**” →** Comando utilizado para enviar um texto e receber como resposta uma interação do usuário.

**read** nome\_da\_variavel → Comando utilizado para armazenar em uma variável o contéudo enviado pelo usuário no comando **“printf”**. Podemos utilizar em uma estrutura condicional o parâmetro **“-t** tempo**”**, para finalizar o programa, e imprimir uma resposta, caso o usuário não digite nada no terminal durante um tempo determinado. Podemos também usar o parâmetro **“-s”**, geralmente utilizado em senhas, pois o contéudo não é mostrado na tela.

**eval "** array **=** **(** **$**array\_em\_string **) " →** Comando utilizado para transformar um Array em formato de string em um Array padrão.

***Tabela com os comparadores de String e Númericos:***



***Tabela com os comparadores de Diretórios:***



***Estrutura de Condição:***

**if [ $**var1 comparador **$**var2 **]**

**then**

Primeira condição.

**elif [ $**var1 comparador **$**var2 **]**

**then**

Segunda condição.

**else**

Terceira condição. **→** Exemplo simples de uma estrutura de condição.

***Estrutura de Condição Case:***

**case $**var

opção 1**)**

Primeira condição.

opção 2**)**

Segunda condição.

opção 3**)**

Terceira condição.

**\*)**

Opção caso todas as outras sejam falsas. →

Exemplo simples de uma estrutura de condição case.

***Estrutura de Repetição For:***

**for $**chave **in ${!**array**[@]}**

**do**

**echo “$chave = ${**array**[$chave]}”**

**done →** Exemplo simples de uma estrutura de repetião for.

***Estrutura de Repetição While:***

**while [ $**var condição **$**var **]**

**do**

Resultado…

**done →** Exemplo simples de uma estrutura de repetião while.

***Estrutura de Repetição Until:***

**ultil [ $**var condição **$**var **]**

**do**

Resultado…

**done →** Exemplo básico de uma esutrura de repetião until.

***While para leitura de arquivos:***

**cat** nome\_do\_arquivo **| while read** linha

**do**

**echo $linha**

**done →** Comando utilizado para fazer a leitura de cada linha em um arquivo.

***Funções:***

**function** nome\_da\_função **{**

Comandos...

**echo** Resultado

**} →** Exemplo de função sem passagem de parâmetros.

nome\_da\_função **(** parametro\_1 **,** paramêtro\_2 **, ...) {**

Comandos…

**echo** Resultado

} → Exemplo de função com passagem de parâmetros.

***Criando uma caixa de mensagem usando whiptail:***

**whiptail --title “**titulo\_da\_caixa**” --msgbox “**mensagem\_da\_caixa**” --fb** linhas colunas **→** Caixa de mensagem whiptail simples.

**whiptail --title “**titulo\_da\_caixa**” --yesno “**mensagem\_da\_caixa**” --fb** linhas colunas **→** Caixa de mensagem whiptail com opção de **sim** e **não**.

**whiptail --title “**titulo\_da\_caixa**” --yes-button “**texto\_da\_resposta\_sim**” --no-button “**texto\_da\_resposta\_não**” --yesno “**mensagem\_da\_caixa**” --fb** linhas colunas **→** Caixa de mensagem whiptail com opção de **sim** e **não** personalizadas.

**whiptail --titile “**titulo\_da\_caixa**” --inputbox “**mensagem\_da\_caixa**” --fb** linhas colunas **3>&1 1>&2 2>&3 →** Caixa de mensagem whiptail de interação com o usuário.

**whiptail --title “**titulo\_da\_caixa**” --checklist --fb \**

**“Quais cidades deseja visitar?”** linhas colunas numero\_de\_itens **\**

**“Londres” “Inglaterra” ON \**

**“Berlim” “Alemanha” OFF \**

**“Toronto” “Canadá” OFF \**

**“Abu Dhabi” “Emirados Árabes” OFF \**

**“Pequim” “China” OFF 3>&1 1>&2 2>&3 →** Caixa de mensagem com opções de seleção.

***Caixas de Diálogo Dialog:***

Para instalar o pacote **“dialog”** utilizando o comando…

**sudo apt-get install dialog**

**dialog --title “**titulo\_da\_caixa**” --msgbox “**texto\_da\_caixa**”** linhas colunas