

**UFABC – Universidade Aberta do Brasil**  
**Curso de Especialização em Tecnologias e Sistemas de Informação**  
Disciplina: Software Livre  
Profa. Denise Goya  
Aluno: Eder Vieira Cavalcante  
Aula 05 – Prática Ambiente Linux

**Exercício 01**

```
#!/bin/bash
# consultar informações pelo nome armazenado no arquivo agenda.dat
if [ $# -eq 0 ]
then
echo "Erro - Você não inseriu um valor"
echo "consultar informações por nome no arquivo agenda.dat"
exit 1
fi
grep -i "$1" agenda.dat
exit 0
```

**Exercício 02**

```
#!/bin/bash
# inserir as informações nome e telefone no arquivo agenda.dat
echo "Entre com o nome:"
read nome
echo "Entre com o numero"
read numero
echo "$nome $numero" >> agenda.dat
cat agenda.dat
exit 0
```

**Exercício 03**

```
#!/bin/bash
# analisar se é dia, tarde ou noite e saudar o usuário.
hora=`date +%H`
# echo "Hora = $hora"
if [ $hora -lt 6 ]; then
echo "Boa madrugada!"
elif [ $hora -lt 12 ] ; then
echo "Bom dia!"
elif [ $hora -lt 18 ] ; then
echo "Boa tarde!"
elif [ $hora -lt 24 ] ; then
echo "Boa noite!"
else
echo "Nada"
fi
exit 0
.
```

#### Exercício 04

```
#!/bin/bash
numero=-1
echo "Entre um número entre 0 e 9"
read num
if [ "$num" -lt 0 ] || [ "$num" -gt 9 ]; then
echo "Digite um número entre 0 e 9."
exit 1
fi
case $num in
0) echo "par" ;;
1) echo "impar" ;;
2) echo "par" ;;
3) echo "impar" ;;
4) echo "par" ;;
5) echo "impar" ;;
6) echo "par" ;;
7) echo "impar" ;;
8) echo "par" ;;
9) echo "impar" ;;
*) echo "Numero invalido!" ;;
esac
exit 0
```

#### Exercício 05

```
#!/bin/bash
echo "Entre com um nome de diretorio"
read diretorio

if [ "$diretorio" = "" ]; then
echo " Argumento inválido"
elif [ ! -d "$diretorio" ]; then
echo " Diretório inexistente"
else
#Listando do mais recente para o menos recente ls -t
echo " Diretorio Valido" $ diretorio
ls $diretorio -t
fi
```

## Exercício 06

```
#!/bin/bash
echo "Iniciando backup de arquivo diretório atual"

#Diretorio a ser utilizado para backup

bkpdir="./bkp"

#Verifica se existe senão cria
if [ ! -d "$bkpdir" ]; then
mkdir -p $bkpdir

fi

#concatenar nome do arquivo com o dia de hoje
day=$(date +%A)

#Fazer o backup usando tar

tar -cf "backup-$day.bak" $(find -name "*.txt")

#mover o arquivo bkp para o diretório bkp

mv -f "backup-$day.bak" bkp/backup-$day.bak

echo "Backup finalizado"
```

## Exercício 07

O script faz um teste simples de verificação de espaço no disco.

Foi criada uma variável `space` primeiramente.

Como principal e primeiro comando: **`df -h`**

O comando `df` mostra as partições e quanto cada uma ocupa no disco. O parâmetro `h` imprime as informações de forma que as pessoas possam ler mais facilmente.

Depois daí para frente os comandos vão filtrando as informações da tabela de partições de forma a chegar na quantidade ocupada por cada disco. Por exemplo utilizando o comando **`grep`** que encontra texto, **`sort`** para fazer a ordenação, **`cut`** para extrair informação. E um dos mais interessantes o `awk` `{print 5}` que imprime a 5ª coluna que representa a porcentagem.

Assim é definido um valor de alerta de 80% que se algum disco estiver acima disso é enviado uma mensagem alertando sobre a situação.

## Exercício 08

O script faz uma cópia dos arquivos no diretório `myhomedirectory` para uma pasta dentro de um diretório no webserver.

Um novo diretório é criado a cada hora.

Se as fotos estão ocupando muito espaço as fotos mais antigas são descartadas.

Logo no início temos a abertura de um laço de repetição utilizando a estrutura `while`.

Logo em seguida é criada uma variável `DISKFUL`, semelhante ao exercício 07 é executado o comando `df -h` para listar a tabela de partição. Porém aqui temos um limite de 90% de ocupação.

Quando o disco alcança essa porcentagem é criado um diretório com a data e hora e as imagens do formato `jpg` são movidas através do comando `mv` para o diretório destino, o sistema aguarda (`sleep`) e retorna a verificar.

A variável `TORREMOVE` tem o arquivo que será encontrado através do comando `find` utilizando o parâmetro `-mtime +30` que encontra arquivos com tempo de modificação de +30 dias atrás.

O script aparente ser de um serviço de vigilância onde precisamos guardar as imagens que foram registradas e quem não podem ser excluídas e sim movidas para um lugar seguro, porém o disco da máquina que executa o serviço deve ser limpo.