

# DESENVOLVIMENTO DE UM SCRIPT EM BASH QUE ENCONTRA TODOS OS ARQUIVOS SEM USO HÁ UM TEMPO DEFINIDO E ENVIA E-MAIL PARA O PROPRIETÁRIO E PARA UMA CONTA ESPECIAL DEFINIDA PARA ADMINISTRAÇÃO

Daniel Darlen Corrêa Ribeiro, Eduardo Campos dos Santos

ARL – Universidade Federal de Lavras (UFLA)  
Lavras – MG - Brasil

daniel\_darlen@yahoo.com.br, edu@edusantos.eti.br

***Resumo.** Este artigo foi criado para servir de documentação sobre a implementação de um aplicativo destinado a uma auditoria sobre a utilização dos arquivos em um dado sistema em plataforma Linux. Espera-se que os estudos aqui apresentados possam servir tanto como uma solução simples para um problema típico de administração de sistemas Linux, e também como material de apoio em estudos sobre a utilização de “bash-scripting” na administração de sistemas Linux. Este artigo elaborado como projeto de conclusão do módulo de Administração de Sistemas Linux do curso de Administração de Redes Linux “Lato Sensu” à Distância da Universidade Federal de Lavras. Professor Msc. Joaquim Uchôa.*

## 1. Introdução

O objetivo deste trabalho, conforme definido pelo tutor do módulo de Administração de Sistemas Linus, o Sr. Joaquim Uchôa no enunciado para o problema é apresentar uma implementação acompanhada de documentação no formato de artigo de um aplicativo em qualquer linguagem padrão em Linux - preferencialmente C/C++, PHP, Java ou Bash que atenda à seguinte característica: “Listar todos os arquivos sem uso há mais de xx meses e enviar e-mail para proprietário e uma conta especial.”

O trabalho iniciou-se pela escolha das linguagens a serem utilizadas observando-se as características das linguagens e o conhecimento dos autores sobre as mesmas. Assim, optou-se por utilizar a linguagem de script bash por considerar-se que implementação de um script em uma linguagem mais flexível e de tipagem fraca apresentaria-se mais simples e suficientemente atrativo para implementação desejada, sobretudo de forma didática. O fato de ser uma linguagem interpretada não traz problemas sobre a performance uma vez que o problema não requer uma execução extremamente rápida e constante.

A solução foi originalmente implementada sobre uma distribuição do Slackware. Na fase de testes percebeu-se a necessidade de alteração de parte do código para que a solução se tornar-se mais portátil. Manteve-se aqui as duas versões por acreditar-se que elas permitem uma ilustração didática de possíveis ocorrências no desenvolvimento de uma implementação para a plataforma Linux.

## **2. Análise do algoritmo**

O algoritmo consiste-se basicamente em:

- 1) Fornecer uma interface para o administrador definir certas características sobre a pesquisa a ser realizada: número de dias ou meses que os arquivos não são acessados; o diretório a partir do qual será efetuado o rastreamento e; a opção quanto a inclusão ou não de arquivos pertencentes a usuários especiais, tais como o “root”, o “bin”, etc.
- 2) Efetuar os devidos testes sobre os argumentos passados.
- 3) Efetuar os devidos testes sobre eventuais diretórios e arquivos temporários.
- 4) Rastrear o sistema de arquivos em busca de arquivos que atendam aos critérios passados, ou seja, localização e tempo de não utilização.
- 5) Construir para cada usuário, uma lista com os arquivos encontrados.
- 6) Enviar uma mensagem por correio eletrônico para cada usuário que teve algum arquivo encontrado no rastreamento incluindo a lista dos arquivos encontrados que lhe pertencem. Enviar uma cópia de cada mensagem passada para uma conta especial definida pelo administrador.
- 7) Apagar eventuais arquivos e diretórios temporários.

### 3. Análise dos “scripts”

A implementação está constituída em dois arquivos: busca.sh e arquivo.sh. A execução é efetuada através da linha de comando apresentada na Figura 3.1. O arquivo busca.sh faz então um rastreamento por arquivos não acessados no período de tempo definido pelo administrador ao executar o script. O segundo arquivo consiste-se numa rotina que é chamada para cada arquivo encontrado que atenda aos critérios definidos. Essa rotina retorna o dono do arquivo e armazena seu nome em uma lista de arquivos encontrados. É criada uma lista para cada usuário.

```
./busca.sh -[d|m] <valor> <diretório> -[e|n]
```

**Figura 3.1: Executando o script.**

Ao definir como o script trataria o período de tempo especificado, optou-se por trabalhar com dias como unidade de tempo. Apesar disso, o script apresenta a possibilidade de especificar o período de tempo desejado tanto em dias como também em meses. Caso o administrador opte pela opção de pesquisa por meses, o script faz a conversão assumindo meses de 30 dias. O parâmetro “dias” é iniciado com o valor “0” (zero). A conta especial que receberá uma cópia de todas as mensagens enviadas aos usuários do sistema é definida no parâmetro “usuário\_especial”. Seu valor deve ser definido diretamente no script, não pode ser passado como argumento. O valor do parâmetro “diretorio\_base” deve ser passado como argumento. Os parâmetros “arq\_mensagem” e “diretorio\_auxiliar” definem os nomes de arquivos temporários que serão gerados, usados e, ao final, apagados pelo script. O primeiro define o arquivo que conterá a mensagem que será encaminhada e o segundo define o diretório que armazenará as listas de arquivos encontrados por usuário. Essas listas, por sua vez, são constituídas por arquivos-textos com o nome igual ao login do usuário. Os parâmetros e seus respectivos valores iniciais estão listados na Tabela 3.1.

Qualquer erro na passagem dos argumentos resulta na chamada à função mensagem\_erro() demonstrada na Figura 3.2. Quanto aos argumentos passados, o script realiza os seguintes testes: se foram passados quatro argumentos; se o terceiro argumento é um diretório; se o quarto parâmetro é “-n” ou “-e” e; se o administrador optou por uma pesquisa passando valor em dias ou em meses, o que é representado pelo valor do primeiro argumento, “-d” para dias ou “-m” para meses. Os dois primeiros testes devem ser satisfeitos, caso contrário a função mensagem\_erro ( ) é chamada. No caso do terceiro teste, se o argumento passado não for equivalente a uma das opções previstas, ou seja, “-e” ou “-n” a mensagem\_de erro ( ) também é chamada. No quarto teste, caso o argumento tenha sido “-m”, o valor passado em meses é convertido para dias imediatamente e, caso tenha sido passado algum outro argumento não previsto, i.e., diferente de “-d” ou “-m”, a função mensagem\_erro ( ) é chamada.

**Tabela 3.1: Parâmetros do script.**

Nome do parâmetro	Valor do parâmetro
usuario_especial	"root"
dias	0
diretorio_base	""
conta_especial	"-a"
arq_mensagem	"/tmp/mensagem_busca.txt"
diretorio_auxiliar	"/tmp/busca_arquivos"

```
mensagem_erro () {
echo "Utilize: $0 -[d|m] <valor> <diretorio> -[e|n]"
echo ""
echo "Onde: "
echo "    d - faz pesquisa por dias."
echo "    m - faz pesquisa por meses."
echo "    e - inclui contas especiais na pesquisa."
echo "        Por exemplo: root, bin, lp"
echo "    n - nao inclui contas especiais na pesquisa."
echo "    Valores: "
echo "        <valor>      - numero de dias ou meses."
echo "        <diretorio> - diretorio de onde partira"
echo "                    a pesquisa."
echo ""
}
```

**Figura 3.2: A função mensagem\_erro ().**

Após o tratamento dos argumentos passados, o script testa se já existe um diretório tal como definido no parâmetro "diretorio\_auxiliar". Caso já exista, o diretório atual é removido e um novo diretório é novamente criado. Isso é feito para garantir que este diretório esteja vazio. Como se trata de um diretório criado em /tmp, um diretório destinado geralmente para receber apenas diretórios temporários, a exclusão do diretório especificado por padrão "/tmp/busca\_arquivos" não deverá trazer problemas ao funcionamento de qualquer outro programa no sistema, sobretudo pelo fato de que a execução do script não constitui num processo demorado. A Figura 3.3 apresenta o trecho do script que trata do diretório temporário.

O rastreamento por arquivos que atendam aos critérios passados é então realizado. Isso é efetuado através do comando "find" conforme ilustrado na Figura 3.4.

```
find $diretorio_base -atime +$dias -exec ./arquivo.sh "{}"  
$diretorio_auxiliar $conta_especial \;
```

**Figura 3.3: Rastreamento dos arquivos**

Para cada arquivo encontrado satisfazendo as condições passadas, i.e., contidos a partir do diretório-base e não acessados dentro do período de tempo requisitado, a rotina “arquivo.sh” é chamada passando três argumentos: o arquivo; o diretório auxiliar e; a informação se é para incluir ou não os arquivos pertencentes a pseudo-usuários usuais, geralmente designados por uid menor que 99. O resultado de cada chamada à rotina “arquivo.sh” é a inclusão do nome do arquivo na lista de arquivos encontrados referente ao usuário dono. Essa lista é um arquivo-texto com o nome do login do dono do arquivo. Caso esta lista ainda não exista, ela é então criada. Isso ocorrerá quando se tratar do primeiro arquivo daquele usuário encontrado no rastreamento. O código do arquivo `arquivo.sh`, está disponibilizado no Anexo C.

Após o rastreamento e a criação dos arquivos contendo as listas respectivas a cada usuário, um arquivo de mensagem personalizada para cada usuário é criado a partir da listagem do diretório temporário. A Figura 3.4 apresenta o trecho referente a isso presente no arquivo `busca.sh` da versão compatível com o Conectiva e a Figura 3.5 apresenta o mesmo trecho presente no arquivo `busca.sh` da versão compatível com o Slackware. A diferença entre as duas versões consiste-se na forma em que a listagem é inserida na mensagem. A implementação foi originalmente criada e testada sobre o Slackware. Ao efetuar um teste na distribuição do Conectiva 7.0, detectou-se que a linha destinada ao envio do email não funcionava porque o parâmetro “-a” do “mail”, que no Slackware permite anexar um arquivo, não é reconhecido pelo comando “mail” da distribuição do Conectiva 7.0. Assim, um novo arquivo foi gerado contendo apenas uma pequena modificação: ao invés de anexar o arquivo contendo a lista de arquivos encontrados, o script efetua um comando “cat” sobre o arquivo contendo a lista e inclui seu conteúdo ao final da mensagem. Na verdade, a segunda solução é mais portátil e funciona igualmente no Slackware, mas optou-se por manter as duas versões aqui a título de ilustração do que ocorreu durante o desenvolvimento e depuração da implementação.

```

for usuario in `ls $diretorio_auxiliar`
do
echo "Usuario $usuario" > $arq_mensagem
echo "" >> $arq_mensagem
echo "Foram encontrados arquivos de sua propriedade" >>
$arq_mensagem
echo "que nao foram acessados ha mais de $dias dias." >>
$arq_mensagem
echo "Por favor verifique a necessidade da permanencia destes" >>
$arq_mensagem
echo "arquivos dentro do sistema." >> $arq_mensagem
echo "" >> $arq_mensagem
echo "Em anexo, segue a lista completa dos referidos arquivos."
>> $arq_mensagem
echo "___" >> $arq_mensagem
echo "Administrador do Sistema." >> $arq_mensagem
echo "" >> $arq_mensagem
echo "" >> $arq_mensagem
echo "+-----+"
>> $arq_mensagem
echo "|Esta eh uma mensagem automatica que propositalmente      |"
>> $arq_mensagem
echo "|nao possui acentos.                                     |"
>> $arq_mensagem
echo "|Uma copia foi enviada para o administrador do sistema. |"
>> $arq_mensagem
echo "+-----+"
>> $arq_mensagem
echo " " >> $arq_mensagem
echo " " >> $arq_mensagem
echo
"....."
....." >> $arq_mensagem
echo " " >> $arq_mensagem
echo "Lista dos arquivos encontrados: " >> $arq_mensagem
echo "" >> $arq_mensagem
cat $diretorio_auxiliar/$usuario >> $arq_mensagem

mail -s "Alerta do sistema: arquivos de $usuario sem acesso."
$usuario -c $usuario_especial < $arq_mensagem

```

**Figura 3.4: Construção e envio da mensagem na versão para o Conectiva**

```

for usuario in `ls $diretorio_auxiliar`
do
echo "Usuario $usuario" > $arq_mensagem
echo "" >> $arq_mensagem
echo "Foram encontrados arquivos de sua propriedade" >>
$arq_mensagem
echo "que nao foram acessados ha mais de $dias dias." >>
$arq_mensagem
echo "Por favor verifique a necessidade da permanencia destes" >>
$arq_mensagem
echo "arquivos dentro do sistema." >> $arq_mensagem
echo "" >> $arq_mensagem
echo "Em anexo, segue a lista completa dos referidos arquivos."
>> $arq_mensagem
echo ""
echo "___" >> $arq_mensagem
echo "Administrador do Sistema." >> $arq_mensagem
echo "" >> $arq_mensagem
echo "" >> $arq_mensagem
echo "+-----+"
>> $arq_mensagem
echo "|Esta eh uma mensagem automatica que propositalmente      |"
>> $arq_mensagem
echo "|nao possui acentos.                                     |"
>> $arq_mensagem
echo "|Uma copia foi enviada para o administrador do sistema. |"
>> $arq_mensagem
echo "+-----+"
>> $arq_mensagem
echo " " >> $arq_mensagem
echo " " >> $arq_mensagem
echo
"....."
....." >> $arq_mensagem
echo " " >> $arq_mensagem

mail -s "Alerta do sistema: arquivos de $usuario sem acesso." -a
$diretorio_auxiliar/$usuario $usuario -cc $usuario_especial <
$arq_mensagem

```

**Figura 3.5: Construção e envio da mensagem na versão para o Slackware.**

## Anexo A - O arquivo busca.sh versão para Conectiva

```
# CURSO DE POS-GRADUACAO EM ADMINISTRACAO DE REDES LINUX
# MODULO ADMINISTRACAO DE SISTEMAS LINUX
#
# DANIEL DARLEN CORRÊA RIBEIRO
# EDUARDO CAMPOS DOS SANTOS
#

# variavel usuario_especial armazena o usuario que possui a conta
# para onde os e-mails serao enviados. Por padrao coloquei root
# mas pode-se criar uma conta específica para esse fim, neste caso
# basta alterar o valor da variavel.
usuario_especial="root"

dias=0
diretorio_base=""
conta_especial="-a"
arq_mensagem="/tmp/mensagem_busca.txt"
diretorio_auxiliar="/tmp/busca_arquivos"

mensagem_erro () {
echo "Utilize: $0 -[d|m] <valor> <diretorio> -[e|n]"
echo ""
echo "Onde: "
echo "    d - faz pesquisa por dias."
echo "    m - faz pesquisa por meses."
echo "    e - inclui contas especiais na pesquisa."
echo "        Por exemplo: root, bin, lp"
echo "    n - nao inclui contas especiais na pesquisa."
echo "    Valores: "
echo "        <valor>      - numero de dias ou meses."
echo "        <diretorio> - diretorio de onde partira"
echo "                    a pesquisa."
echo ""
}

#testa se existe os quatro parametros
if test -z $4
then
echo "Uso incorreto. "
mensagem_erro
exit 1
else
#testa se o 3o. parametro passado eh um diretorio
if test -d $3
then
diretorio_base=$3
else
echo "Parametro passado nao e um diretorio: $3"
mensagem_erro
exit 2
fi

#testa 4o. parametro e se este eh "-e" ou "-n"
if [ $4 = "-e" ]
```



```

then
conta_especial=$4
else
if [ $4 = "-n" ]
then
conta_especial=$4
else
echo "Parametro incorreto: $4"
mensagem_erro
exit 4
fi
fi

#testa se o usuario referenciou dias ou meses
if [ $1 = "-d" ]
then
dias=$2
else if [ $1 = "-m" ]
then
let "dias= $2 * 30"
else
echo "Parametro $1 incorreto"
mensagem_erro
exit 3
fi
fi
fi

if test -d $diretorio_auxiliar
then
rm -R $diretorio_auxiliar
fi
mkdir $diretorio_auxiliar

# Faz o rastreamento por arquivos nao acessados ha mais de <$dias>
dias.
# Para cada arquivo encontrado a rotina ./arquivo.sh eh chamada para
# retornar o dono do arquivo e armazenar uma lista dos arquivos
encontrados
# por usuario.

find $diretorio_base -atime +$dias -exec ./arquivo.sh "{}"
$diretorio_auxiliar $conta_especial \;

# A lista com os arquivos de cada usuario estah armazenada em
# <$diretorio_auxiliar>. Cada arquivo deste diretorio recebe
# como nome o "dono" dos arquivos listados.
# A partir de cada arquivo contido no diretorio serah gerado
# e-mail.

for usuario in `ls $diretorio_auxiliar`
do
echo "Usuario $usuario" > $arq_mensagem
echo "" >> $arq_mensagem
echo "Foram encontrados arquivos de sua propriedade" >>
$arq_mensagem
echo "que nao foram acessados ha mais de $dias dias." >>
$arq_mensagem
echo "Por favor verifique a necessidade da permanencia destes" >>
$arq_mensagem

```

```

echo "arquivos dentro do sistema." >> $arq_mensagem
echo "" >> $arq_mensagem
echo "Em anexo, segue a lista completa dos referidos arquivos." >>
$arq_mensagem
echo "___" >> $arq_mensagem
echo "Administrador do Sistema." >> $arq_mensagem
echo "" >> $arq_mensagem
echo "" >> $arq_mensagem
echo "+-----+" >>
$arq_mensagem
echo "|Esta eh uma mensagem automatica que propositalmente      |" >>
$arq_mensagem
echo "|nao possui acentos.                                     |" >>
$arq_mensagem
echo "|Uma copia foi enviada para o administrador do sistema. |" >>
$arq_mensagem
echo "+-----+" >>
$arq_mensagem
echo " " >> $arq_mensagem
echo " " >> $arq_mensagem
echo
"....." >> $arq_mensagem
....." >> $arq_mensagem
echo " " >> $arq_mensagem
echo "Lista dos arquivos encontrados: " >> $arq_mensagem
echo "" >> $arq_mensagem
cat $diretorio_auxiliar/$usuario >> $arq_mensagem

mail -s "Alerta do sistema: arquivos de $usuario sem acesso."
$usuario -c $usuario_especial < $arq_mensagem

done

rm -R $diretorio_auxiliar
rm -R $arq_mensagem

```

## Anexo B - O arquivo busca.sh versão para Slackware

```
#!/bin/bash
#
# CURSO DE POS-GRADUACAO EM ADMINISTRACAO DE REDES LINUX
# MODULO ADMINISTRACAO DE SISTEMAS LINUX
#
# DANIEL DARLEN CORRÊA RIBEIRO
# EDUARDO CAMPOS DOS SANTOS
#

# variavel usuario_especial armazena o usuario que possui a conta
# para onde os e-mails serao enviados. Por padrao coloquei root
# mas pode-se criar uma conta específica para esse fim, neste caso
# basta alterar o valor da variavel.
usuario_especial="root"

dias=0
diretorio_base=""
conta_especial="-a"
arq_mensagem="/tmp/mensagem_busca.txt"
diretorio_auxiliar="/tmp/busca_arquivos"

mensagem_erro () {
echo "Utilize: $0 -[d|m] <valor> <diretorio> -[e|n]"
echo ""
echo "Onde: "
echo "    d - faz pesquisa por dias."
echo "    m - faz pesquisa por meses."
echo "    e - inclui contas especiais na pesquisa."
echo "        Por exemplo: root, bin, lp"
echo "    n - nao inclui contas especiais na pesquisa."
echo "    Valores: "
echo "        <valor>      - numero de dias ou meses."
echo "        <diretorio> - diretorio de onde partira"
echo "                    a pesquisa."
echo ""
}

#testa se existe os quatro parametros
if test -z $4
then
echo "Uso incorreto. "
mensagem_erro
exit 1
else
#testa se o 3o. parametro passado eh um diretorio
if test -d $3
then
diretorio_base=$3
else
echo "Parametro passado nao e um diretorio: $3"
mensagem_erro
exit 2
fi
fi
```

```

#testa 4o. parametro e se este eh "-e" ou "-n"
if [ $4 = "-e" ]
then
conta_especial=$4
else
if [ $4 = "-n" ]
then
conta_especial=$4
else
echo "Parametro incorreto: $4"
mensagem_erro
exit 4
fi
fi

#testa se o usuario referenciou dias ou meses
if [ $1 = "-d" ]
then
dias=$2
else if [ $1 = "-m" ]
then
let "dias= $2 * 30"
else
echo "Parametro $1 incorreto"
mensagem_erro
exit 3
fi
fi
fi

if test -d $diretorio_auxiliar
then
rm -R $diretorio_auxiliar
fi
mkdir $diretorio_auxiliar

# Faz o rastreamento por arquivos nao acessados ha mais de <$dias>
dias.
# Para cada arquivo encontrado a rotina ./arquivo.sh eh chamada para
# retornar o dono do arquivo e armazenar uma lista dos arquivos
encontrados
# por usuario.

find $diretorio_base -atime +$dias -exec ./arquivo.sh "{}"
$diretorio_auxiliar $conta_especial \;

# A lista com os arquivos de cada usuario estah armazenada em
# <$diretorio_auxiliar>. Cada arquivo deste diretorio recebe
# como nome o "dono" dos arquivos listados.
# A partir de cada arquivo contido no diretorio serah gerado
# e-mail.

for usuario in `ls $diretorio_auxiliar`
do
echo "Usuario $usuario" > $arq_mensagem
echo "" >> $arq_mensagem
echo "Foram encontrados arquivos de sua propriedade" >>
$arq_mensagem
echo "que nao foram acessados ha mais de $dias dias." >>
$arq_mensagem

```

```

echo "Por favor verifique a necessidade da permanencia destes" >>
$arq_mensagem
echo "arquivos dentro do sistema." >> $arq_mensagem
echo "" >> $arq_mensagem
echo "Em anexo, segue a lista completa dos referidos arquivos." >>
$arq_mensagem
echo ""
echo "__" >> $arq_mensagem
echo "Administrador do Sistema." >> $arq_mensagem
echo "" >> $arq_mensagem
echo "" >> $arq_mensagem
echo "+-----+" >>
$arq_mensagem
echo "|Esta eh uma mensagem automatica que propositalmente      |" >>
$arq_mensagem
echo "|nao possui acentos.                                     |" >>
$arq_mensagem
echo "|Uma copia foi enviada para o administrador do sistema. |" >>
$arq_mensagem
echo "+-----+" >>
$arq_mensagem
echo " " >> $arq_mensagem
echo " " >> $arq_mensagem
echo
"....." >> $arq_mensagem
echo " " >> $arq_mensagem

mail -s "Alerta do sistema: arquivos de $usuario sem acesso." -a
$diretorio_auxiliar/$usuario $usuario -cc $usuario_especial <
$arq_mensagem

done

rm -R $diretorio_auxiliar
rm -R $arq_mensagem

```

## Anexo C - O arquivo arquivo.sh (idêntico para as duas versões)

```
#!/bin/bash
#
# CURSO DE POS-GRADUACAO EM ADMINISTRACAO DE REDES LINUX
# MODULO ADMINISTRACAO DE SISTEMAS LINUX
#
# DANIEL DARLEN CORREA RIBEIRO
# EDUARDO CAMPOS DOS SANTOS
#

arquivo=$1
diretorio_auxiliar=$2
flag_busca=$3

if ! test -d $diretorio_auxiliar
then
echo "ERRO: $diretorio_auxiliar nao existe."
exit 98
fi

if [ $flag_busca = "-e" ]
then
    #inclui as contas especiais na busca --> uid < 99
    parametros=`find "$arquivo" -printf %u -printf "\t" -printf %U -
printf "\t" -printf "%p" `
    set $parametros
    echo $3 >> $diretorio_auxiliar/$1
else
    if [ $flag_busca = "-n" ]
    then
#nao inclui contas especiais na busca --> uid > 98
parametros=`find "$arquivo" -printf %u -printf "\t" -printf %U -
printf "\t" -printf "%p" `
set $parametros
if [ $2 -gt 98 ]
then
echo $3 >> $diretorio_auxiliar/$1
fi
else
"Echo: Erro não previsto - 97"
exit 97
fi
fi
```