

AWK TUTORIAL

Introdução

O awk trabalha sempre dessa forma:

```
awk padrão ação
```

Quando o awk encontra um padrão na busca ele executa a ação padrão que é imprimir o registro inteiro que é \$0

```
awk '/pattern/'
```

O padrão é um termo buscado, pode ser imprimir as linhas que contém a palavra sérgio do arquivo de grupos:

```
awk -F: '/sergio/ {print $1}' /etc/group
```

O awk considera o espaço como separador de campos padrão, devido à estrutura do arquivo "/etc/group" modificamos o separador de campos usando:

```
-F:
```

O que está entre barras "/" é o padrão a ser buscado e o "\$1" diz ao awk para imprimir o primeiro campo.

Caso não seja informado nenhum padrão o awk imprime a linha inteira, a menos que peçamos para imprimir um campo, veja:

```
awk -F: '{print $1}' /etc/group
```


No comando acima não foi informado nenhum padrão, apenas foi informado ao awk qual o separador de campos e qual campo queríamos como saída.

Obs: Sempre que trabalhar com o awk use aspas simples pois o cifrão "\$" representa para o awk indicador de campos (começado em 1, se for zero é a linha toda), mas se não forem usadas aspas simples o shell tentará interpretar primeiro buscando ali uma variável que na verdade não existe.

Variáveis do awk

NR	Contém o número do registro atual
NF	contém o número de registros por campo
FS	indica o separador de campos
RS	indica o separador de registros
OFS	indica o separador de campo do arquivo de saída
ORS	indica o separador de registros do arquivo de saída

Problema resolvido com operador ternário

- fonte: Lista brasileira sobre Linux Shell Script 

Pessoal, boa tarde.

Estava estudando aqui um pouquinho sobre awk, e cheguei no seguinte problema:

```
$ cat -E tmp/arquivos.txt
#/home/odin/shell/tmp/arquivo1.txt$
#/home/odin/shell/tmp/arquivo2.txt$
/home/odin/shell/tmp/arquivo3.txt$
#/home/odin/shell/tmp/arquivo4.txt$
/home/odin/shell/tmp/arquivo5.txt$
```

o que eu quero é exibir isso tudo em 1 linha apenas, ignorando as linhas que começam com #, exibindo apenas o nome do arquivo e separando-os por ; (obs, não é para ter ; no final do texto, apenas entre os nomes).

Solução

```
awk -F '/' ' !/#/{s=(s)?s";"$NF:$NF}END{print s}' arquivos.txt
```

Ainda com relação ao operador ternário podemos fazer coisas como

```
seq 1 30 | awk 'ORS = NR % 5 ? FS:RS'
```

No código acima o separador de registros de saída 'ORS' será igual ao separador de registros (quebra de linha) caso o resto da divisão seja zero, caso contrário o separador de registro de saída será um espaço. O resultado é que os valores serão agrupados de 5 em 5.

```
awk 'ORS=/foo$/?FS:RS'
```

joins the following line to the current one if the current one ends in foo.
You are encouraged to find your own variations.

Listar usuários alfabeticamente

```
awk -F ':' '{print $1 | "sort";}' /etc/passwd
```

como receber variáveis do sistema no awk

```
awk -v n=${var} '$0 ~ n {print $0}' teste.txt  
awk -v sign="$var" '{if ($1 == sign) print$1}' teste.txt
```

como imprimir os registros que contenham um determinado padrão?

Digamos que você tem um arquivo assim

```
HostName: Empresa1-AD1  
TaskName: Veracid  
Last Result: 1  
Start Time: 07:00:00  
Start Date: 18/12/2011  
Power Management: No Start On Batteries, Stop On Battery Mode  
  
HostName: Empresa2-AD1  
TaskName: Veracid-01  
Last Result: 0  
Start Time: 08:00:00  
Start Date: 18/12/2011  
Power Management: No Start On Batteries, Stop On Battery Mode  
  
HostName: Empresa1-AD2  
TaskName: Veracid-02  
Last Result: 01d  
Start Time: 18:00:00  
Start Date: 17/11/2011  
Power Management: No Start On Batteries, Stop On Battery Mode
```

Você deseja tratar cada bloco como um registro e imprimir os registros em que o campo "Last Result:" não termine em zero

```
awk 'BEGIN {RS="";FS="\n"} {if ($3 ~ /^[^0]$/) print $0,"\n"}' teste.txt
```

Ao alterar o separador de Registros RS e o Separador de campos (Field Separator) tratamos cada bloco como um registro, basta agora fazer um if, no caso como expressões regulares que diz "não zeros [^0]" no final da linha \$
OBS: o padrão da busca em expressões regulares fica entre barra /pattern/

o \$0 faz referência ao registro todo

Como deletar linhas alternadas

- [stackoverflow](#)

```
echo -e "um\\ndo\\ntres\\nquatro\\ncinco\\nseis\\n" | awk 'NR%2'
```

Usando awk com dois padrões

```
awk '/Apples/ && !/Red Apples/' file
```

O comando acima pode ser usado no lugar de um duplo grep em que se busca 'Apples' mas não se busca 'Red Apples'.

ignore case - ignorando maiúsculas e minúsculas

- [source: StakOverflow](#)

```
echo "Create" | awk 'IGNORECASE = 1;/^create/;
```

separadndo dados - awk range

Nossa missão, separar países de cidades

```
Country1  
city1  
city2
```

```
Country2  
city3  
city4
```

```
awk 'BEGIN {RS="";FS="\n"} {print $1 > "countries"} {for (i=2;i<=NF;i++) print $i > "cities"}'  
  
# imprime do campo 2 em diante  
awk 'BEGIN {RS="";FS="\n"} {print substr($0,match($0,$2))}' teste.txt
```

source: <http://stackoverflow.com/questions/6436170/extracting-line-after-new-line>

imprimir ultimo campo

```
awk '{print $NF}' file  
# imprimir o ultimo campo 'juntado em uma só linha'  
awk -F":" '{printf $NF " "}' file  
  
# somando o ultimo campo  
awk '{sum += $NF} END {print sum}' downloads.txt
```

buscar ocorrência exata com awk

```
awk '$2=="172.17.4.1" {print $1}' file.txt
```

numerar linhas

```
awk '{print FNR "\t" $0}' files*
```

imprimir numero da linha de um padrão

```
awk '/enable bash completion/ {print NR}' /etc/bash.bashrc
```

imprimindo tudo menos o 5º campo

```
awk '{ $5=""; print }' file
```

imprimindo o primeiro e o último campos

- <http://br.groups.yahoo.com/group/shell-script/message/31590> ↗
- <http://br.groups.yahoo.com/group/shell-script/message/33109> ↗

Tenho uma coluna que contém nomes, exemplo:

```
Ciclano Fulano da Silva Soares  
José dos Anzóis Pereira  
Juca da Silva  
Mateus Marcus Lucas e João  
Benedita Maria da Silva  
Santa Maria
```

estou precisando formatá-los pegando apenas o primeiro e o último nome, o único problema é que eles não tem o mesmo tamanho... como devo fazer? alguém pode me ajudar?

JJ.Santanna

```
awk '{print $1,$NF}' teste.txt
```

Pegando o número ip

```
ifconfig eth0 | awk '/inet add/ {print $3}'  
awk '/inet addr/ {print $2}' <(ifconfig eth0)  
#obs: a opção --> hostname -I é mais prática  
  
# o comando netstat pega a interface padrão  
awk '/inet end./ {print $3}' <(ifconfig $(netstat -i | awk '/eth/ {print $1}'))
```

No caso acima pedimos para listar o endereço atualmente atribuído à interface de rede eth0 (lembre-se que em alguns sistemas a interface padrão pode ser eth1). Em seguida a saída que deveria ser mostrada na tela é redirecionada para o awk. O awk por sua vez procura pelo padrão "inet end" e imprime o terceiro campo retornando somente o ip.

Pegando o endereço do roteador

```
awk 'NR==2 {print $1}' /proc/net/arp
```

A explicação: Mostre o campo 1 '\$1' do registro 2 'NR==2' do arquivo /proc/net/arp

Removendo linhas duplicadas

```
awk '!($0 in a) {a[$0];print}' file  
  
awk '!a[$0]++' file
```

Imprimindo linhas de um intervalo

```
awk 'NR >= 27 && NR <= 99' file.txt > /tmp/file3
```

fonte: <http://br.groups.yahoo.com/group/shell-script/message/31705>

Srs.,

Boa tarde!

Gostaria de uma ajuda para solucionar este desafio(pra min..rs).
Desejo capturar um intervalo de linhas separadas por chaves.

ex:

```
#arquivo teste
primeira_linha
teste{
linha1
linha2
linha3
}
ultima_linha
```

Gostaria de saber qual comando posso utilizar para que seja possível jogar o que está entre chaves para alguma variável:

```
linha1
linha2
linha3
```

desde já agradeço atenção!

att,
Cleberson.

Solução

```
cat teste.txt | awk '/teste{/,/}/' | grep -v '{\|}'
```

Imprimir linhas em intervalos específicos (neste caso a cada 3 linhas)

```
awk 'NR%3==1' file
```

um comentário a cada 5 linhas

```
ls -l | awk '{if (NR % 5 == 0) print "-- COMMIT --"; print}'
```

imprimindo um espaço a cada duas linhas

Bash insert newline after every 2 lines

source: <http://unstableme.blogspot.com/2011/05/bash-insert-newline-after-every-3-lines.html> ↗

```
awk '!( NR % 2 ) {$0 = $0"\n"} {print}' file
```

juntando linhas de 3 em 3

você tem isto

```
field1  
field2  
field3  
field1  
field2  
field3
```

e quer isto

```
field1,field2,field3  
field1,field2,field3
```

deve então fazer isto

```
awk '{if (NR%3==0){print $0;}else{printf "%s,", $0;}}' file.txt  
  
# ou  
awk '{ORS=NR%3?" ":"\n";print}' file.txt
```

zerando a primeira coluna

```
awk '{$1=""; print}'
```

fazendo substituição em coluna específica

```
awk '{gsub("foo","bar",$5)}1' file
```

apagando a primeira e terceira colunas

```
awk '{$1=$3=""}1' file
```

Para o awk o '1' é visto como true e portanto equivale a imprimir

Substituição de um padrão - usando a função sub

```
echo "isto aquilo" | awk '{ sub(/aquilo/, "aquele"); print }'
```

Substituindo o terminador de linha

Convert Windows/DOS newlines (CRLF) to Unix newlines (LF) from Unix.

```
awk '{ sub(/\r$/, ""); print }'
```

Convert Unix newlines (LF) to Windows/DOS newlines (CRLF) from Unix.

```
awk '{ sub(/$/, "\r"); print }'
```

Substitute “foo” with “bar” only on lines that contain “baz”.

```
awk '/baz/ { gsub(/foo/, "bar") }; { print }'
```


imprimir um intervalo

```
awk '/start_pattern/,/stop_pattern/' file.txt
```

colocando em maiúsculas

```
echo "um dois" | awk '{print $1, toupper($2)}'
```

convertendo tags html/xml para minúsculas

- <http://gutocarvalho.net/wordpress/2008/08/07/dica-convertendo-uppercase-to-lowercase/> 

referência inicial

```
cat usuarios.txt | awk '{print tolower($1)}'
```

Concatenando linhas

Concatenate every 4 lines of input with a comma.

```
awk 'ORS=NR%4?" , ":"\n"'
```

O comando acima resolve questões como esta:

<http://br.groups.yahoo.com/group/sed-br/message/4369> 

Senhores,
Ainda não consegui achar uma solução, tem um arquivo que ele assim:

```
dn: uid=xx,ou=Usuarios,dc=bbbb,dc=com,dc=br
cn: fulano de tal
homePhone: 053922
mail: fulano@...
dn: uid=yy,ou=Usuarios,dc=bbb,dc=com,dc=br
cn: beltrano de tal
homePhone: 053923
mail: beltrano@...
```

Isso se repete, sempre, dn:, cn: homePhone: e mail:, gostaria de deixar ele dessa forma:

```
dn: uid=xx,ou=Usuarios,dc=bbbb,dc=com,dc=br;cn: fulano de tal;homePhone: 053922
```

em coluna para cada 4 linhas do arquivo. Alguém tem alguma idéia de como fazer isso?

Calculadora com awk

```
# coloque no seu ~/.bashrc
calc(){ awk "BEGIN{ print $* }" ;}
```

Exibindo os maiores arquivos

Insira estas linhas no bashrc

```
biggest () {
du | sort -r -n | awk '{split("k m g",v); s=1; while($1>1024){$1/=1024; s++} pr
}
```

em seguida pode abrir outro terminal e fazer

```
biggest | head -20
```

retirar ultimo diretório de uma lista

```
echo /teste/teste1/teste2/teste3 | awk -F"/" '{print $NF}'
```

-F"/" = especifica / como divisor de campo
\$NF = último campo

Ordenando strings por tamanho

```
awk '{ print length($0),$0 | "sort -n"}' file
```

Imprimindo linhas maiores que

```
awk 'length>72' file
```

imprimir linhas com mais de três campos

```
awk 'NF >= 3' filename
```

Somando listas

```
ls -la | awk 'BEGIN{count=0}; {count++; print $5} END{ print "Total:", count}'
```

outro exemplo

```
awk '{sum+=$2} END {print sum}' arquivo
```

Somando os números de um arquivo

```
34
42
11
6
2
99
...
```

```
awk '{ sum += $1 } END { print sum }' file
```

Trocar de posição os campos 1 e 2 de um arquivo e imprimir a linha

```
{ temp = $1 ; $1 = $2 ; $2 = temp ; print }
```

Um outro jeito mais inteligente!

```
'{print $2,$1}'
```

imprimindo linhas impares

```
cat -n /etc/passwd | awk '$1 % 2 == 1 {print}'
```

Combinando com ls

```
ls -l | awk '$6 == "Nov" { sum += $5 }  
END { print sum }'
```

Localizando processos por usuário

```
ps aux | awk '!/root|nobody/ { if ($4>2) {print $2,$11}}'
```

Pegar coluna de linha especifica

```
#exemplo da coluna 2 na linha 5  
awk 'NR == 5 {print $2}'
```

imprimir primeira coluna da linha 2

```
awk 'NR==2 {print $1}' arquivo
```

somando números em um arquivo

```
awk '{s+=$0} END {print s}' /tmp/file.txt
```

somando os 10 primeiros números

```
$ seq 10 | awk '{ i+=$1 } END { print i }'  
55
```

Opção com o paste + bc para somar numeros em um arquivo

```
paste -sd+ /tmp/file.txt | bc  
  
awk '{ sum += $1 } END { print sum }' file
```

Calculando sua fatura do com o awk

Eu tenho uma fatura assim:

```
12/01/2007 INSINUANTE_LJ_187(12/12) 187,91
04/09/2007 RABELO_SOM_E_IMAGEM(04/12) 51,75
03/11/2007 EXTRA_1375_PARANGABA(03/04) 41,75
03/11/2007 EXTRA_1375_PARANGABA(03/04) 11,75
13/11/2007 INTUICAO(02/02) 34,50
21/11/2007 ADAUTO_PECAS(02/06) 200,00
17/12/2007 ANUIDADE(12/12) 7,00
26/12/2007 VIACAO_ITAPEMIRIM(01/03) 259,34
```

```
awk '{ total = total + $3} END {print total}' fatura.txt
```

```
awk '{s+=$3} END {print s}' fatura.txt
```

A variável total receber o valor dela + o acumulado do terceiro campo do arquivo "fatura.txt", mas só é impresso a soma no final graças ao uso do END

Resultado:

```
794
```

somando campos de arquivos csv

```
awk -F ',' '{ x = x + $4 } END { print x }' test.csv
```

procura e mostra linhas com o terceiro campo igual a "peter" ou que contenham o string "marie":

```
cat nomedoarquivo | awk '$3 == "peter" || $3 ~ /marie/'
```

Imprimir o tamanho de uma linha

```
awk '/root/ {print NR " - " length}' /etc/passwd
```

imprimir linhas menores que

```
awk 'length<30' /etc/passwd
```

Imprimir uma linha específica

a ação padrão do awk é imprimir portanto não precisa fazer 'nr=1234 {print}' basta fazer o teste comparando o número do registro (linha)

```
awk 'NR==1234' arquivo
```

Imprimir o nome dos usuários acima de um determinado GID

```
awk -F":" '{if ($4 >= 1000) print $1}' /etc/passwd
```

Imprimir nome de usuário o id

```
awk -F":" '{ print "username: " $1 "\t\tuid:" $3 }' /etc/passwd
```

Programa que imprime o tamanho da maior linha

```
awk '{ if (length($0) > max) max=length($0) } END { print max }' texto.txt
```

Imprimir linhas maiores que 80 caracteres

```
length($0) > 80
```

visualizando logs do squid

```
tail -f /var/log/squid/access.log | awk '{print "Host: "$3 "site: "$7}'
```

Imprimindo um trecho de arquivo no awk

```
awk '/start_pattern/,/stop_pattern/' file.txt
```

Mostrar usuários com ID acima de 1000

```
awk -F":" '$3 > 1000 {print $0}' /etc/passwd
```

Modificando o separador de registros e de campos

Se você tem o arquivo...

```
1
2
3
4
5
teste
6
7
8
9
10
teste
11
12
13
14
15
teste
16
17
18
19
20
teste
```

... e quer definir como delimitador de registros a palavra "teste" e o delimitador de campos para quebra de linha faça:

```
awk 'BEGIN { FS="\n"; RS="teste"} {if (NR=="2") print }' teste.txt
```

Imprimir valores negativos

Como imprimir somente os valores negativos usando o awk?

Queremos imprimir somente os valores negativos do seguinte arquivo

```
-5232,-92338,84545,34  
-2233,25644,23233,2  
6211,-1212,4343,43  
-2434,621171,9121,-33
```

```
awk 'BEGIN {RS="",|\n"}/^-/ {print}' file.txt
```

Indicamos duas alternativas para o Separador de Registros, vírgula ou quebra de linha ", |\n" assim ele considera cada valor

como sendo uma linha, em seguida usamos uma expressão regular para pegar o começo de linha com o sinal de menos

Remover linhas em branco

```
awk 'NF>0' texto.txt
```

Modificar campo 2 mediante uma condição

```
awk '{print $1,($2<0.1) ? 0.0 : $2}' file1.txt
```

Imprimir somente usuários com UID igual a 1.000 ou maior

```
awk -F: '$3 > 999 {print $0}' /etc/passwd
```

Imprimir a última linha

```
awk 'END {print $0}' /etc/passwd
```

contar o número de linhas em um arquivo

```
awk 'END { print NR }' <arquivo>
```


Problemas da lista shell script

Olá pessoal... tenho o seguinte arquivo:

```
cat file.txt
```

```
bdacp tty1 Thu Nov 13 12:16 - down (00:31)
bdacp tty3 Thu Nov 13 12:53 - down (00:13)
bdacp tty3 Thu Nov 13 13:09 - down (02:34)
reboot system boot 2.4.31 Thu Nov 13 12:15 (00:32)
reboot system boot 2.4.31 Thu Nov 13 12:48 (00:18)
reboot system boot 2.4.31 Thu Nov 13 13:07 (02:35)
teste pts/0 Thu Nov 13 12:53 - 12:53 (00:00)
teste tty2 Thu Nov 13 12:51 - 12:55 (00:04)
teste tty2 Thu Nov 13 13:09 - 13:31 (00:22)
usuario pts/0 Thu Nov 13 12:49 - 12:50 (00:00)
usuario tty1 Thu Nov 13 12:49 - 13:04 (00:14)
usuario tty1 Thu Nov 13 13:09 - 13:23 (00:14)
```

e preciso dessas informações

```
bdacp 12:16 - down (00:31)
bdacp 12:53 - down (00:13)
bdacp 13:09 - down (02:34)
reboot 12:15 (00:32)
reboot 12:48 (00:18)
reboot 13:07 (02:35)
teste 12:53 - 12:53 (00:00)
teste 12:51 - 12:55 (00:04)
teste 13:09 - 13:31 (00:22)
usuario 12:49 - 12:50 (00:00)
usuario 12:49 - 13:04 (00:14)
usuario 13:09 - 13:23 (00:14)
```

solução

```
awk '{$1, $6, $8, $9}' arquivo
```


print the total number of lines that contain "Beth"

```
awk '/Beth/{n++;} END {print n+0}' file
```

sequencia de fibonacci em awk

```
seq 50 | awk 'BEGIN {a=1; b=1} {print a; c=a+b; a=b; b=c}'
```

Awk - replace blank spaces with single space

Fonte: <http://unstableme.blogspot.com/2009/11/awk-replace-blank-spaces-with-single.html> 

Input file 'file.txt' contains fields separated with uneven spaces (or tabs)

```
$ cat file.txt
6767      1212   9090 12
657676 1212  21212      21232
76767      12121 909090   121212
12      9090      1212      21
```

Required: Replace one or more space with single space or single tab or comma.

The solutions using awk:

```
$ awk -v OFS="," '$1=$1' file.txt
Output:
6767,1212,9090,12
657676,1212,21212,21232
76767,12121,909090,121212
12,9090,1212,21
```

```
$ awk -v OFS="\t" '$1=$1' file.txt
Output:
6767    1212    9090    12
657676  1212    21212   21232
76767   12121   909090   121212
12      9090    1212    21
```

```
$ awk -v OFS=" " '$1=$1' file.txt
Output:
6767 1212 9090 12
657676 1212 21212 21232
76767 12121 909090 121212
12 9090 1212 21
```

Recebendo variáveis do shell

```
read_val=0301; awk -v var=${read_val} '{sub(/0019/,var); print}' infile
```

```
echo "Enter ID: "  
read ID  
awk -v ID=$ID 'NR==14{ print ID}{print}' "file"  
  
# assim dá pra usar o argv para passar alguns parâmetros no script  
  
awk 'BEGIN {print ARGV[5]}' 1 2 3 4 5 6 7  
  
awk 'BEGIN {print ARGV[5] "\n" ARGV[3]}' 1 2 3 4 5 6 7  
  
awk -v sq=`pwd` 'BEGIN {print "você está em " sq}'
```

extraíndo dados de um arquivo

Você tem um arquivo assim:

```
a.txt  
{some data}  
success!!  
  
b.txt  
{some data}  
success!!  
  
c.txt  
{some data}  
error!!
```

deseja a seguinte saída:

```
a.txt: success  
b.txt: success  
c.txt: error
```

com awk dá pra fazer assim:

```
awk 'BEGIN {RS="\n\n"} {gsub("!!","", $NF); print $1": "$NF}' teste.txt  
  
# um jeito bem mais simples  
awk '{print $1": "$4}' RS="\n\n" results.txt
```

juntando linhas a cada 5 linhas

- fonte: <http://br.groups.yahoo.com/group/shell-script/message/31367> 

Pessoal,

tenho uma comando que gera a seguinte saida, onde a unica padronagem é uma sequencia de 5 linhas:

```
0
r5
751625160
601300096
391584768
1
r5
1401393800
1121115008
621056
12
r5
1401393800
1121115008
68344320
45
r5
1401393800
1121115008
1235456
223
r5
2522508840
2242230016
388346880
```

Eu precisava gerar a saida neste formato, em 5 colunas:

```
0 r5 751625160 601300096 391584768
1 r5 1401393800 1121115008 621056
12 r5 1401393800 1121115008 68344320
45 r5 1401393800 1121115008 1235456
223 r5 2522508840 2242230016 388346880
```

Como poderei fazer?

Obrigado,
Alessandro Almeida.

```
awk '{printf("%s%s", $0, (NR%5 ? " " : "\n"))}'
```

como saber se seu sistema é 32 ou 64 bits?

```
file /bin/bash | awk '{print $3}'
```

Listar os 10 comandos mais usados

```
history | awk '{print $2}' | sort | uniq -c | sort -rn | head
```

processando html com o awk

- <http://www.levlafayette.com/node/128>
- <https://www.readability.com/articles/ugx7y6nv>

```
#!/bin/bash
awk ' {
print $0
if(NR % 8 == 0 )
print "<tr>"
} ' file.csv > file1.csv

#Or, directly on the command line:

awk ' {print $0; if(NR % 8 == 0 ) print "<tr>"} ' file.csv > file1.csv
```

Fonte:

<http://linuxdicas.wikispaces.com/awk>

<http://matheuslion.blogspot.com/2012/11/awk-tutorial-o-melhor.html>