AWK = Aho, Weinberger and Kernighan

Programação III : Março 2002

Intro.

AWK é uma linguagem de programa para o processamento de texto. O nome do Unix utilitário que interprete este linguagem também se chame AWK. Não é apenas um filtro de Unix que transforme o seu input mas também consegue fazer cálculos e outros operações conforme o seu input, por exemplo

- procure de padrões
- transformação de texto
- produções de relatórios e estatísticas

Como a escrita dum programa em AWK é geralmente mais rápido do que num linguagem compilado como C ou em Pascal é muito utilizado como ferramenta de trabalho de dia a dia por administradores de sistemas e base de dados e para rápido prototyping duma aplicação. Obviamente não tem a mesmo segurança e poder duma linguagem de alto nível.

Utilização:

O texto é lido para registos (geralmente uma linha) e cada registos é dividido em campos (fields), a divisão de campos é normalmente um espaço ou tab (\t).
Para cada registo é feito o seguinte programa **Condições Acção**

Quer dizer o registo é comparada com as condições. Se for verdade é feito as acções

Condição , envolvendo possibilidade de testar:

- se o registo lido (ou cada um dos seus campos) inclui ou não certo padrão
- expressões envolvendo quantidades, comprimentos de campos, números de registos, etc
- condição por defeito: Verdadeira

Acção , por exemplo:

- cálculo de estatísticas
- escrita de texto
- substituição de texto
- atribuições, ciclos, condicionais, etc.
- acção por defeito: copiar registo de entrada para a saída (standard output)

Elementos de Linguagem

- Constantes numéricas
- Constantes strings: incluídas entre aspas; podem ter "\`n"
- Operadores (como C)
- Funções predefinidos (log, sqrt etc.)
- A função print "string" ou possibilidade de utilizar sintaxe de printf de C printf("%s",string)

Variáveis
Podem ser simples (inteiros, reais, strings) ou ARRAYS.
Os arrays podem ter índices inteiros ou strings!
Não são declaradas e são automaticamente inicializadas a 0, "" ou array vazio, conforme o respectivo tipo (automático).

Acesso aos Registos e Campos

- $0 - registo actual
- $1, $2 - campo1, campo2 do registo actual
- $NF - ultimo campo

Palavras reservadas e variáveis globais predefinidas:

- BEGIN - Padrão que é verdade no início (para inicializações de variáveis, etc.)
- END - Padrão que é verdade no fim do ficheiro (para fazer acções finais)
- FILENAME - Nome do ficheiro actualmente a ser processado
- NF - número de campos do registo actual
- NR - número do registo actual
- FS - O separador de campos do registo de entrada (por defeito uma sequência de 'espaços ' " ou '\t' pode ser mudado por exemplo FS=":")
- RS - Separador de registos na entrada (por defeito a nova linha '\n')

Exemplos

- Imprimir pela ordem oposta os dois primeiros campos de cada registo lido menos o registo/linha 5.

```
awk   'NR!=5 { print $2, $1}'  dados.in
```

- Imprimir os campos pela ordem inversa

    awk -f invert dados.in
    sendo o ficheiro "invert": for (i = NF; i > 0; --i) print $i

- Calcular o somatório do primeiro campo de todos os registos do ficheiro "dados.in" e imprimi- lo assim como a média

    awk -f soma.awk dados.in
    sendo o ficheiro "soma.awk":
    { s += $1 }
    END { print "soma = ", s, " media = ", s/NR }

Para Mais Informações sobre AWK Ver

1) ciunix > man awk
2) pagina : http://www.dmi.ubi.pt/~crocker/prog3/docs.html
3) fazer pesquisa para awk em www.google.com
4) AWK Language Programming
http://www.cl.cam.ac.uk/texinfodoc/gawk_toc.html

Exercícios

1) rel.awk
Escrever um ficheiro awk com nome a partir dum ficheiro de texto com o seguinte formato

Title1 Title2
2 3
3 11
etc.

Produzir um relatório com seguinte formato

Title 1 [2] Title 2 [3]
Title 1 [3] Title 2 [11]
etc
Soma Title1 233
Media Title1 23.2

2) df.awk
Escrever um ficheiro awk que utilizando como input o output do comando Unix df produz um relatório com informação sobre o espaço em disco(s) do computador

    Output
    Numero Total de (512) Blocos xxxxx
    Espaço Total xx MB yy GB
    Total Utilizado yy GB
    Total Disponível xx GB yy %
    Numero de Mount Points xx

3) awk.cpu
Escrever um ficheiro awk que permite imprimir o % total de CPU do computador a ser utilizado juntamente com o processo que está a consumir a maior quantidade

    ps opções | awk -f cpu.awk
    Total CPU 12.2 %
    Processo Mais Pesado : CPU 2.2 Utilizador a1234 A Executar bigprog

4) txt.awk
Escrever um ficheiro awk que permite transformar um texto de seguinte maneira,

- Apagar linhas que contêm a palavra "Faro"
- Substituir todos os ocorrências de Palavra "Lisboa" por "Covilha"
- Duplicar linhas cujo segundo campo é igual a Fundao

4) nome.awk : No ficheiro de Passwords /etc/passwd apanhar as "Catarina"'s

    awk -f awk.nome /etc/passwd

    Output
    Nome [Catarina Tomas] Userid [ctomas]
    Nome [Ana Catarina ferreira] userid [a_14297]

5) Invente um exercício com AWK, não esquecendo a sua solução.

Fazer os Exercícios em grupos de Dois. Mandar as suas soluções em apenas um ficheiro de texto com os seus nomes nos primeiros duas linhas para o endereço crocker@ciunix.ubi.pt