Caixa de ferramentas

A linguagem AWK

http://www.inf.ufpr.br/marcos/caixa_de_ferramentas

Marcos Alexandre Castilho

DInf UFPR, Curitiba PR

19 de maio de 2021

Sobre a caixa de ferramentas

- Todo mundo deveria ter uma em casa!
- Martelos, chaves, parafusos variados, furadeira, ...
- Nesta série quero mostrar o básico de coisas que existem e eu gosto
- O objetivo é fazer videos curtos e deixá-los interassados a buscar mais informações
- A web é repleta de tutoriais sobre tudo que eu apresento

AWK

- AWK é uma poderosa linguagem de programação criada nos anos 70 mais ou menos em conjunto com o próprio sistema operacional Unix e a linguagem de programação C
- O nome vem das iniciais dos sobrenomes de seus criadores, que dispensam maiores apresentações:
  - Alfred **A**ho
  - Peter **W**einberger
  - Brian **K**ernighan
- É uma linguagem voltada para processamento de textos, manipulação de dados e serve como uma excelente ferramenta para criação de protótipos (que as vezes se tornam programas definitivos)

Motivação

- AWK pode ser usada em scripts shell (tipo bash) em meio a uma sequência de pipes, como por exemplo (um exemplo bobo):
  - A saída de um `ls -l` é chata de manipular com um `cut` por causa dos espaços em branco
  - AWK é mais elegante!

Exemplo bobo

```
marcos@macalan:/home/bcc$ ls -l | head
total 2044
drwx------ 18 abc76     bcc  4096 mai 19 18:30 abc76
drwx------ 76 aafj16     bcc  4096 mai 19 15:29 aafj16
drwx------ 21 aam12      bcc  4096 mai 19 15:29 aam12
drwx------ 53 abw17      bcc  4096 mai 19 18:22 abw17
drwx------ 33 acfm16     bcc  4096 mai 19 17:57 acfm16
drwx------ 36 acgbas19   bcc 36864 mai 19 18:52 acgbas19
drwxr-xr-x 50 ach13      bcc  4096 mai 19 15:29 ach13
drwx------ 44 acq14      bcc  4096 mai 19 15:29 acq14
drwx------ 34 advc19     bcc  4096 mai 19 16:37 advc19
```

- Se fizer um `ls -l | head | cut -d\ -f 4` a saída mostra várias linhas em branco
- AWK lida melhor com estes espaços em branco, conforme veremos

Motivação

- Como dito, este exemplo é bobo
- AWK é uma linguagem de programação extremamente elegante, poderosa e simples
- Pode ser usada para processar arquivos obtendo resultados complexos com base em uma estrutura muito simples
- É uma linguagem interpretada

O básico

- AWK processa textos com base em expressões regulares
- Estas expressões regulares são tratadas por comandos que podem ser grandes programas com tudo o que uma linguagem tem direito: desvios, repetições, etc
- Possui estruturas de dados do tipo vetor, por exemplo
- Mas é tudo natural, sem complicações, embora a semântica dos vetores seja diferente, aliás, é bem legal!
- As noções fundamentais são os registros e os campos

Registros e campos

- Registros são tipicamente linhas
- Campos são tipicamente colunas
- Mas isto pode ser alterado
- As linhas são processadas uma de cada vez a partir de um padrão que casa (como se fosse um grep)
- Para os padrões que casam, os comandos operam sobre as colunas
- Estes comandos podem ser meros prints ou podem ser programas complexos

Estrutura

- Para cada linha lida, um padrão é procurado
- Uma vez encontrado, o comando executa para esta linha
- O comando na verdade é uma ação, quer dizer, o que fazer com a linha que casa com o padrão

```
condição { ação }
condição { ação }
...
condição { ação }
```

Alguns exemplos simples

```
marcos@macalan:/home/bcc$ ls -l | head
total 2044
drwx------ 18 abc76     bcc  4096 mai 19 18:30 abc76
drwx------ 76 aafj16     bcc  4096 mai 19 15:29 aafj16
drwx------ 21 aam12      bcc  4096 mai 19 15:29 aam12
drwx------ 53 abw17      bcc  4096 mai 19 18:22 abw17
drwx------ 33 acfm16     bcc  4096 mai 19 17:57 acfm16
drwx------ 36 acgbas19   bcc 36864 mai 19 18:52 acgbas19
drwxr-xr-x 50 ach13      bcc  4096 mai 19 15:29 ach13
drwx------ 44 acq14      bcc  4096 mai 19 15:29 acq14
drwx------ 34 advc19     bcc  4096 mai 19 16:37 advc19
```

- `ls -l | head | awk '/15:/ {print $0}'` (grep)

```
marcos@macalan:/home/bcc$ ls -l | head | awk '/15:/ {print $0}'
drwx------ 76 aafj16     bcc  4096 mai 19 15:29 aafj16
drwx------ 21 aam12      bcc  4096 mai 19 15:29 aam12
drwxr-xr-x 50 ach13      bcc  4096 mai 19 15:29 ach13
drwx------ 44 acq14      bcc  4096 mai 19 15:29 acq14
```

Alguns exemplos simples

```
marcos@macalan:/home/bcc$ ls -l | head
total 2044
drwx------ 18 abc76     bcc  4096 mai 19 18:30 abc76
drwx------ 76 aafj16     bcc  4096 mai 19 15:29 aafj16
drwx------ 21 aam12      bcc  4096 mai 19 15:29 aam12
drwx------ 53 abw17      bcc  4096 mai 19 18:22 abw17
drwx------ 33 acfm16     bcc  4096 mai 19 17:57 acfm16
drwx------ 36 acgbas19   bcc 36864 mai 19 18:52 acgbas19
drwxr-xr-x 50 ach13      bcc  4096 mai 19 15:29 ach13
drwx------ 44 acq14      bcc  4096 mai 19 15:29 acq14
drwx------ 34 advc19     bcc  4096 mai 19 16:37 advc19
```

- `ls -l | head | awk '/15:/ {print $4}'` (grep | cut)

```
marcos@macalan:/home/bcc$ ls -l | head | awk '/15:/ {print $4}'
bcc
bcc
bcc
bcc
```

O princípio

- Uma linha é dividida em campos
- Os campos são variáveis cujos nomes são posicionais ($1, $2, $3, $4, ...)
- Algumas variáveis controlam quantas colunas (ou linhas) existem
- A linha inteira é $0

drwx——	18	abc76	bcc	4096	mai	19	18:30	abc76
$1	$2	$3	$4	$5	$6	$7	$8	$9

As consequências

- Você pode usar o poder das linguagens de programação, com comandos de desvio, repetição, ... para fazer o que quiser com a linha processada!
- Detalhe: você pode definir o separador de linhas, bem como o de colunas

Exemplo quase básico

- Script AWK para simular uma planilha
- No próximo slide, um trecho de uma planilha
- No seguinte, um trecho de um programa que processa a planilha

Exemplo de planilha (em formato CSV)

- Eu mantenho um arquivo CSV com os dados de alunos de uma turma
- O GRR, número de faltas, nota das provas 1, 2 e 3, nota da final, nome completo, curso, turma e alguma informação adicional, tipo este exemplo aqui:

```
GRR20159999:14:19:05:--::Fulano de Tal:21A:A:--
GRR20023357:22:13:00:--::Beltrano ben Truc:96A:K:--
GRR19985762: 2:78:70:100::Sicrano de Bergecroix:96A:K:OK
```

Trechos do script AWK

- Aqui trechos do script que processa as notas destes alunos
- Isto é, eu não preciso das malfadadas planilhas...

Trechos do script AWK

```
mat=$1
faltas=$2
p1=$3
p2=$4
p3=$5
final=$6
nome=$7
media=(p1 + 2*p2 + 3*p3)/6
freq=(60-faltas)
if (media < 40) situacao="R-nota"
else if (media >= 70) situacao="Aprovado"
     else
       if ( final != "" ) {
          media=(media+final)/2
          media=int(media)
          if (media >= 50) situacao="Aprovado"
          else situacao="R-nota"
       }
       else situacao="Final "
printf "%s  %2s %3s %3s %3s ", mat, faltas, p1, p2, p3
```

Enfim

- Pela definição da caixa de ferramentas, foi apresentado o mínimo do mínimo da linguagem AWK
- Sabendo o básico, já ajuda um monte!
- Estudando então...
- É uma linguagem muito bacana! Garanto pra vocês.