

Processando Logs com GAWK

Raniere Gaia Costa da Silva¹

14 de maio de 2013

¹ra092767@ime.unicamp.br

Os arquivos desta apresentação encontram-se disponíveis em <https://gitorious.org/raniere-presentations/tech>.

Licença

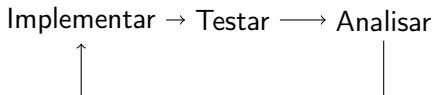
Salvo indicado o contrário, esta apresentação está licenciada sob a licença Creative Commons Atribuição-Compartilhagual 3.0 Não Adaptada. Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-sa/3.0/>.



- 1 Motivação
- 2 Dependências
- 3 Funcionamento
- 4 Expressões regulares
- 5 Atividade

Logs para CSV

Ao executar vários testes obtemos vários arquivos de logs. Para analisar os resultados é conveniente construir uma tabela (csv) com as informações relevantes presentes nos logs.



Linguagens

Sugeridas:

- AWK
- Python
- Perl

Possíveis:

- C/C++
- Fortran
- GNU Octave/MATLAB
- Julia
- Ruby

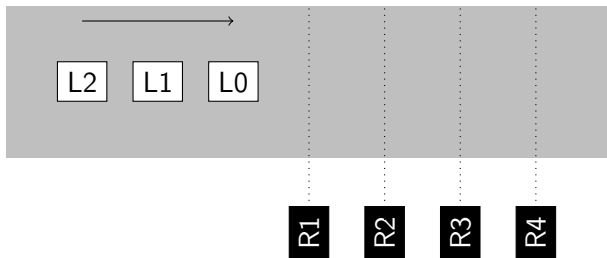
Lista de Dependências

- BASH,
- GAWK.

Alguns arquivos utilizados nos exemplos encontram-se como anexo deste pdf e também são disponibilizados no repositório desta apresentação.

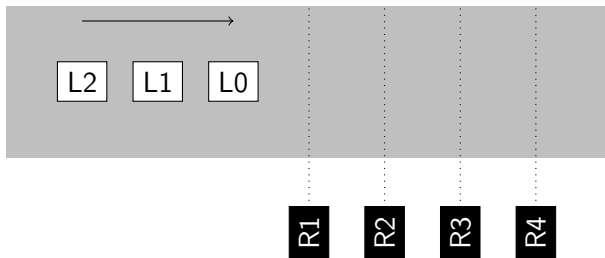
Lógica

Cada linha passa por várias regras que são aplicadas ou não.



Lógica

Cada linha passa por várias regras que são aplicadas ou não.



```
R1 { COMANDO }  
R2 { COMANDO }  
R3 { COMANDO }  
R4 { COMANDO }
```


Chamada

```
$ awk program [argument ...]  
$ awk -f progfile [argument ...]
```

Hello world

```
$ cat begin.awk  
BEGIN { print "Hello world" }  
$ echo | awk -f begin.awk  
Hello world
```

Hello world

```
$ cat begin.awk
BEGIN { print "Hello world" }
$ echo | awk -f begin.awk
Hello world
```

```
$ cat end.awk
BEGIN { print "Hello world" }
END { print "Don't Panic" }
$ echo | awk -f end.awk
Hello world
Don't Panic
```

Exemplo de regras (1)

Considerando o arquivo [sample](#).

```
$ cat feijoada.awk
/feijoada/ {print NR": " $0 }
$ awk -f feijoada.awk sample
20: sexta  arroz          feijoada  moyaje  uva
```

Exemplo de regras (1)

Considerando o arquivo [sample](#).

```
$ cat feijoada.awk
/feijoada/ {print NR": " $0 }
$ awk -f feijoada.awk sample
20: sexta  arroz          feijoada    moyaje uva

$ cat tomate.awk
/tomate/ {print NR": " $0 }
$ awk -f tomate.awk sample
17: terca  arroz feijao peixe    tomate acelora
```

Exemplo de regras (2)

```
$ cat fei.awk
/fei/ {print NR": " $0 }
$ awk -f fei.awk sample
2: segunda arroz feijao carne-moida repolho abacaxi
3: terca  arroz feijao frango      alface uva
4: quarta arroz feijao carne-assada rucula laranja
5: quinta arroz feijao peixe-frito pepino limao
6: sexta  arroz feijao carne-picada agriao tangerina
9: segunda arroz feijao carne-assada moyaje laranja
10: terca  arroz feijao peixe      rucula limao
11: quarta arroz feijao carne-moida alface acerola
12: quinta arroz feijao peixe-frito agriao tangerina
13: sexta  arroz feijao strogonofe repolho laranja
16: segunda arroz feijao frango-frito pepino uva
17: terca  arroz feijao peixe      tomate acelora
18: quarta arroz feijao carne-moida rucula limaola
19: quinta arroz feijao frango      alface abacaxi
20: sexta  arroz      feijoadada  moyaje uva
```

Exemplo de regras (3)

```
$ cat frango-uva.awk  
/frango uva/ {print NR": " $0 }  
$ awk -f frango-uva.awk sample
```

Exemplo de regras (3)

```
$ cat frango-uva.awk
/frango uva/ {print NR": " $0 }
$ awk -f frango-uva.awk sample

$ cat peixe-rucula-limao0.awk
/peixe rucula limao/ {print NR": " $0 }
$ awk -f peixe-rucula-limao0.awk sampl
```


Exemplo de regras (3)

```
$ cat frango-uva.awk
/frango uva/ {print NR": " $0 }
$ awk -f frango-uva.awk sample

$ cat peixe-rucula-limao0.awk
/peixe rucula limao/ {print NR": " $0 }
$ awk -f peixe-rucula-limao0.awk sampl

$ cat peixe-rucula-limao1.awk
/peixe      rucula limao/ {print NR": " $0 }
$ awk -f peixe-rucula-limao1.awk sample
10: terca  arroz feijao peixe      rucula limao
```

Letras

Considerando o arquivo `regex0`.

```
$ awk '/F/ {print $0}' regex0
```

```
F00
```

```
Foo
```

```
F00
```

```
F01
```

```
F10
```

Letras

Considerando o arquivo `regex0`.

```
$ awk '/F/ {print $0}' regex0
```

```
F00
```

```
Foo
```

```
F00
```

```
F01
```

```
F10
```

```
$ awk '/F0/ {print $0}' regex0
```

```
F00
```

```
F01
```

Curingas (1)

```
$ awk '/.0/ {print $0}' regex0
```

```
F00
```

```
F01
```

```
F10
```

```
f00
```

```
f01
```

```
f10
```

Curingas (1)

```
$ awk '/.0/ {print $0}' regex0
```

```
F00
```

```
F01
```

```
F10
```

```
f00
```

```
f01
```

```
f10
```

```
$ awk '/.[01]/ {print $0}' regex0
```

```
F00
```

```
F01
```

```
F10
```

```
f00
```

```
f01
```

```
f10
```

Curingas (2)

```
$ awk '/.0|1/ {print $0}' regex0
```

```
F00
```

```
F01
```

```
F10
```

```
f00
```

```
f01
```

```
f10
```

Curingas (2)

```
$ awk '/.0|1/ {print $0}' regex0
```

F00

F01

F10

f00

f01

f10

```
$ awk '/.[^01]/ {print $0}' regex0
```

F00

Foo

foo

f00

Repetição (1)

Considerando o arquivo `regex1`.

```
$ awk '/0/ {print $0}' regex1
```

F0

Fo0

F00

Fo0o

Fo00

Foo0

F0o0

F0oo

Repetição (1)

Considerando o arquivo `regex1`.

```
$ awk '/0/ {print $0}' regex1
```

F0

Fo0

F00

Fo0o

Fo00

Foo0

F0o0

F0oo

```
$ awk '/00/ {print $0}' regex1
```

F00

Fo00

Repetição (2)

```
$ awk '/0*/ {print $0}' regex1
```

f

fo

foo

fooo

F

Fo

Foo

Fooo

F0

Fo0

F00

Fo0o

Fo00

Foo0

F0o0

F0oo

Repetição (3)

```
$ awk '/0+o/ {print $0}' regex1
```

```
Fo0o
```

```
F0o0
```

```
F0oo
```

Repetição (3)

```
$ awk '/0+o/ {print $0}' regex1
```

```
Fo0o
```

```
F0o0
```

```
F0oo
```

```
$ awk '/0?o/ {print $0}' regex1
```

```
fo
```

```
foo
```

```
fooo
```

```
Fo
```

```
Foo
```

```
Fooo
```

```
Fo0
```

```
Fo0o
```

```
Fo00
```

```
Foo0
```

```
F0o0
```

```
F0oo
```

Explicação

Tomando o arquivo [grow22.log.glpk](#), queremos:

- nome do problema,
- número de linhas (iniciais),
- número de colunas (iniciais),
- número de não zeros (iniciais),
- número de iterações,
- valor ótimo da função objetivo.

Nome do problema

```
$ cat glpk-name.awk  
/Problem/ {print $2}  
$ awk -f glpk-name.awk grow22.log.glpk  
GROW22
```

Nome do problema

```
$ cat glpk-name.awk
/Problem/ {print $2}
$ awk -f glpk-name.awk grow22.log.glpk
GROW22
```

```
$ cat glpk-lname.awk
/Problem/ {print tolower($2)}
$ awk -f glpk-lname.awk grow22.log.glpk
grow22
```

Tamanho do problema

```
$ cat glpk-size0.awk
/* rows, .* columns,.* non-zeros/ {print $0}
$ awk -f glpk-size0.awk grow22.log.glpk
441 rows, 946 columns, 8318 non-zeros
441 rows, 946 columns, 8318 non-zeros
440 rows, 863 columns, 8169 non-zeros
```


Tamanho do problema

```
$ cat glpk-size0.awk
/* rows, .* columns,.* non-zeros/ {print $0}
$ awk -f glpk-size0.awk grow22.log.glpk
441 rows, 946 columns, 8318 non-zeros
441 rows, 946 columns, 8318 non-zeros
440 rows, 863 columns, 8169 non-zeros

$ cat glpk-size1.awk
BEGIN {size = 0}
/* rows, .* columns,.* non-zeros/ {if (size == 0) print $0; size = 1;}
$ awk -f glpk-size1.awk grow22.log.glpk
441 rows, 946 columns, 8318 non-zeros
```

Tamanho do problema

```
$ cat glpk-size0.awk
/* rows, .* columns,.* non-zeros/ {print $0}
$ awk -f glpk-size0.awk grow22.log.glpk
441 rows, 946 columns, 8318 non-zeros
441 rows, 946 columns, 8318 non-zeros
440 rows, 863 columns, 8169 non-zeros
```

```
$ cat glpk-size1.awk
BEGIN {size = 0}
/* rows, .* columns,.* non-zeros/ {if (size == 0) print $0; size = 1;}
$ awk -f glpk-size1.awk grow22.log.glpk
441 rows, 946 columns, 8318 non-zeros
```

```
$ cat glpk-size2.awk
BEGIN {size = 0}
/* rows, .* columns,.* non-zeros/ {if (size == 0) print $1 " ", " $3 ", " $5;
    size = 1;}
$ awk -f glpk-size2.awk grow22.log.glpk
441, 946, 8318
```

Número de Iterações

```
$ cat glpk-it0.awk
/*: obj/ {it = $2}
END {print it};
$ awk -f glpk-it0.awk grow22.log.glpk
587:
```

Número de Iterações

```
$ cat glpk-it0.awk
/*: obj/ {it = $2}
END {print it};
$ awk -f glpk-it0.awk grow22.log.glpk
587:
```

```
$ cat glpk-it1.awk
/*: obj/ {it = $2}
END {sub(/:/, "", it); print it};
$ awk -f glpk-it1.awk grow22.log.glpk
587
```

Valor ótimo

```
$ cat glpk-obj.awk  
/.*: obj/ {obj = $5}  
END {print obj};  
$ awk -f glpk-obj.awk grow22.log.glpk  
-1.608343365e+08
```

Tudo como entrada de CSV

```
$ cat glpk-all.awk
BEGIN {size = 0}
/Problem/ {name = tolower($2)}
/.* rows, .* columns,.* non-zeros/ {if (size == 0) row = $1; col = $3; nnz = $5;
    size = 1;}
/.*: obj/ {it = $2; obj = $5;}
END {printf "%s,%d,%d,%d,%d,%s\n", name, row, col, nnz, it, obj};
$ awk -f glpk-all.awk grow22.log.glpk
grow22,441,863,8169,587,-1.608343365e+08
```

Obrigado!

`r.gaia.cs@gmail.com`