


LPOO - Rede de Colaboração


101 - Linux

Última atualização: 14 de maio de 2013

 **0003.** Considerando o arquivo `example1.apps` cujas linhas são apresentadas abaixo:

```
1 @ "Linear"
2 "Upper" vector 2 1 1
3 "Lower" vector 2 -1 -1
4 @@
5
6 @ "Evaluator"
7 "Executable Name" string "example1"
8 "Input Prefix" string "ex1_in"
9 "Output Prefix" string "ex1_out"
10 @@
11
12 @ "Solver"
13 "Debug" int 4
14 "Initial X" vector 2 2e-1 0.3
15 @@
```

- Escreva uma chamada do `gawk` que imprima a linha em que **Upper** é informado.
- Escreva uma chamada do `gawk` que imprima a linha em que **Lower** é informado.
- Escreva uma chamada do `gawk` que imprima apenas o inteiro correspondente ao nível do Debug.

 **0004.** Considerando o arquivo `example3.apps` cujas linhas são apresentadas abaixo:


```
1 @ "Bounds"
2 "Upper" vector 3 30 30 30
3 "Lower" vector 3 DNE 1 DNE
4 "Scaling" vector 3 1 1 1
5 @@
6
7 @ "Evaluator"
8 "Executable Name" string "example3"
9 "Input Prefix" string "example3_input"
```

```

10 "Output Prefix" string "example3_output"
11 @@
12
13 @ "Solver"
14 "Debug" int 3
15 "Bounds Tolerance" double 1e-8
16 "Initial Step" double 1
17 "Step Tolerance" double 1e-5
18 "Sufficient Decrease Factor" double 0
19 @@

```

- Escreva uma chamada do `gawk` e outra do `sed` que mude o nível de `Debug` de 3 para 4.
- Escreva uma chamada do `gawk` e outra do `sed` que mude a tolerância do passo de $1e-5$ para $1e-8$.

 **0005.** Considerando o arquivo `example3.apps` cujas linhas são apresentadas abaixo:

```

1 [19:16:29]Running PSPDOC
2 sifdecode PSPDOC
3
4 Problem name: PSPDOC
5
6 Double precision version will be formed.
7
8 The objective function uses          2 nonlinear groups
9
10 There are          3 free variables
11 There is          1 variable  bounded only from above
12
13
14 *****
15 Problem name: PSPDOC
16
17 Number of Variables: 4
18 Number of Constraints: 0
19             Equality: 0
20             Inequality: 0
21
22 EXIT: The Algorithm has Converged
23 f(x) = 2.41421
24 |c(x)| = 0
25 |g(x) + J(x)'*y| = 2.11481e-08
26 y offset = 0
27 BFGS? no
28 Number of Iterations = 9
29 Elapsed Time = 0.00047493 s

```

```

30 x:
31   -1
32   2.75217e-13
33   1.17866e-07
34   1.17866e-07
35   *****
36 Setup time: 0
37 Execution time: 0

```

- Escreva uma chamada do **gawk** que informe o nome do problema apenas uma vez.
- Escreva uma chamada do **gawk** que informe o número de variáveis.
- Escreva uma chamada do **gawk** para determinar se o algoritmo convergiu.
- Escreva uma chamada do **gawk** que informe o número de iterações.


 **0006.** Considerando o arquivo `rhomax-table.tex` cujas linhas são apresentadas abaixo:

```

1 problem & nvar & ncon & f & |gp| & |c| & iter & time &
   res & \\ \hline
2 HS99EXP & 31 & 21 & -1.13865e+12 & 0.000885729 & 2.4544e
   -09 & 1416 & 0.386497 & con & \\ \hline
3 GRIDGENA & 6218 & 0 & 6.43184e+09 & 6.59871e+12 & 0 & 54
   & 0.87214 & unc & \\ \hline
4 RAYBENDL & 2050 & 0 & 7.50112e+10 & 1.3646e+16 & 0 & 57 &
   1.98024 & unc & \\ \hline
5 RAYBENDS & 2050 & 0 & 2.54583e+16 & 1.16334e+22 & 0 & 74
   & 55.056 & unc & \\ \hline
6 LUKVLE2 & 10000 & 4999 & -1e+20 & 2.43431e+09 & 2.41308e
   -07 & 60 & 12.7858 & con & \\ \hline
7 LUKVLE4 & 10000 & 4999 & 6.39713e+09 & 5.78738e+13 &
   1.29647e-11 & 91 & 2.33251 & con & \\ \hline
8 LUKVLE8 & 10000 & 9998 & 1.06096e+06 & 0.00129202 &
   1.22916e-08 & 8731 & 155.004 & con & \\ \hline
9 LUKVLI10 & 10000 & 9998 & 1e+20 & 6.04144e+59 & 9.34885e
   -07 & 81 & 2.50585 & con & \\ \hline
10 LUKVLI4 & 10000 & 4999 & 7.80472e+09 & 7.04007e+13 &
   3.75377e-11 & 4256 & 114.12 & con & \\ \hline
11 COSINE & 10000 & 0 & -9999 & 0.700572 & 0 & 65 &
   0.640611 & unc & \\ \hline
12 SCOSINE & 5000 & 0 & 668.412 & 7.38587e+06 & 0 & 133 &
   1.87982 & unc & \\ \hline

```

Escreva um programa utilizando o **gawk** que informa o número de problemas em que o campo `res` é `con` e `unc`.

 **0007.** Considerando o arquivo `dcicpp.col` cujas linhas são apresentadas abaixo:

```
1 3PK Converged .01
2 AGG of 382.49
3 AIRCRFTA Converged .01
4 AIRCRFTB Converged .01
5 AIRPORT Converged .45
6 AKIVA Converged .01
7 ALJAZZAF Converged 28.64
8 ALLINIT Converged 0
9 ALLINITC Converged .01
10 ALLINITU Converged 0
11 ALLINQP unlimited .45
12 ALSOTAME Converged 0
13 ANTWERP Converged .65
14 ARGAUSS stationary 0
15 ARGLALE stationary 1.00
16 ARGLBLE the 7193.96
17 ARGLCLE the 7189.76
18 ARGLINA Converged .28999999
19 ARGLINB Converged .26999999
20 ARGLINC Converged .26999999
21 ARGTRIG Converged .53
22 ARTIF of 1226.28
23 ARWHDNE the 7191.05
24 ARWHEAD Converged .44
25 AUG2D Converged .56
```

Escreva um programa utilizando o `gawk` que informa o número de problemas que convergiram e a soma do tempo de execução dos mesmos.