Automatizando testes

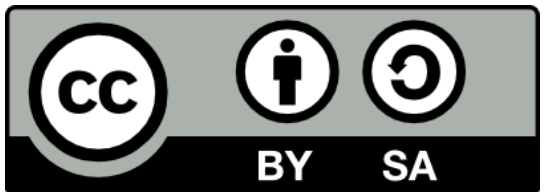
Raniere Gaia Costa da Silva[1]

3 de maio de 2013

[1] ra092767@ime.unicamp.br

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

Licença



Funções e Binário

Funções

Situação muito comum de sua implementação no GNU Octave/MATLAB.

Binário

Situação muito comum de "solvers" famosos.

Abordagem

Funções

Chamada da função desejada utilizando a CLI.
Utilizar um laço que chama a função desejada várias vezes.

Abordagem

Funções

Chamada da função desejada utilizando a CLI.
Utilizar um laço que chama a função desejada várias vezes.

Binários

Similar a funções mas precisando utilizar o Bash ou outra linguagem de script.

Linguagens

Sugeridas:

- Bash: Executar binários é (estupidamente) simples.
- Python: Alto nível e ótimo parse de argumentos.

Possíveis:

- GNU Octave/MATLAB
- Julia
- Perl
- Ruby

Lista de Dependências

- BASH,
- GAWK,
- GLPK,
- Netlib LP (`http://www.netlib.org/lp/data/`),
- MIPLIB (`http://miplib.zib.de/`).

Resolução das Dependências

Debian, Ubuntu, Mint, ...

```
# apt-get install glpk
```

Fedora, ...

```
# yum install glpk
```

Souce

```
$ ./configure
$ make
# make install
```

Variáveis Locais

Declaração

```
$ primeiro_nome=Maria
$ ultimo_nome=Silva
```

Variáveis Locais

Declaração

```
$ primeiro_nome=Maria
$ ultimo_nome=Silva
```

Uso

```
$ echo $primeiro_nome
Maria
$ echo $primeiro_nome$ultimo_nome
MariaSilva
$ echo $primeiro_nome $ultimo_nome
Maria Silva
```

Variáveis Globais

Uso

```
$ cat print_nome.sh
#!/bin/bash
echo $primeiro_nome
$ ./print_nome.sh
```

Variáveis Globais

Uso

```
$ cat print_nome.sh
#!/bin/bash
echo $primeiro_nome
$ ./print_nome.sh
```

Declaração e Uso

```
$ export primeiro_nome=Maria
$ ./print_nome.sh
Maria
```

Vetores

declaração

```
$ vetor=(1 2 3 4)
```

Vetores

declaração

```
$ vetor=(1 2 3 4)
```

Uso

```
$echo $vetor[0]
1[0]
$ echo ${vetor[0]}
1
$ echo ${vetor[*]}
1 2 3 4
$ echo $((${vetor[1]} + ${vetor[2]}))
5
```

Aspas

Simples

```
$ echo '$primeiro_nome'
$primeiro_nome
```

Aspas

Simples

```
$ echo '$primeiro_nome'
$primeiro_nome
```

Duplas

```
$echo "$primeiro_nome"
Maria
```

Expansão

Substituição em strings

```
$ echo $primeiro_nome
Maria
$ echo ${primeiro_nome/a/o}
Moria
$ echo ${primeiro_nome/%a/o}
Mario
$ echo ${primeiro_nome//a/o}
Morio
```

Expansão

Substituição em strings

```
$ echo $primeiro_nome
Maria
$ echo ${primeiro_nome/a/o}
Moria
$ echo ${primeiro_nome/%a/o}
Mario
$ echo ${primeiro_nome//a/o}
Morio
```

Comandos

```
$ echo $(ls /)
bin boot dev etc home lib lib64 lost+found mnt opt proc root run sbin srv sys
    tmp usr var
$ echo `ls /`
bin boot dev etc home lib lib64 lost+found mnt opt proc root run sbin srv sys
    tmp usr var
```

For (1)

Síntaxe

```
for NAME [ [in [WORDS ...] ] ; ] do COMMANDS; done
```

For (1)

Síntaxe

```
for NAME [ [in [WORDS ...] ] ; ] do COMMANDS; done
```

Exemplo 1

```
$ for i in 1 2 3 4; do echo $i; done
1
2
3
4
```

For (2)

Exemplo 2

```
$ for i in Joao Maria Pedro Ana; do echo $i; done
Joao
Maria
Pedro
Ana
```

For (2)

Exemplo 2

```
$ for i in Joao Maria Pedro Ana; do echo $i; done
Joao
Maria
Pedro
Ana
```

Exemplo 3

```
$ for d in `ls / | head -n 4`; do echo $d; done
bin
boot
dev
etc
```

Adquirir testes

Baixar

```
$ wget -nd -nH -np -r -l 1 http://www.netlib.org/lp/data/
```

Adquirir testes

Baixar

```
$ wget -nd -nH -np -r -l 1 http://www.netlib.org/lp/data/
```

Descomprimir

```
$ gcc -o emps emps.c
$ cat netlib-lp.awk
BEGIN {
    process = 0;
}
/Name/ {process = 1; next;}
/BOUND-TYPE/ {if (process) exit 0;}
{if (NF && process) print tolower($1);}
$ for i in `awk -f netlib-lp.awk readme`; do ./emps -s $i; done
```

GLPK

Opções

```
--mps             read LP/MIP problem in fixed MPS format
--tmlim nnn       limit solution time to nnn seconds
--memlim nnn      limit available memory to nnn megabytes
--log filename    write copy of terminal output to filename
```

Exemplo

```
$ ./glpsol --mps afiro.mps --log afiro.log
```

Rodando testes

Laço

```
for f in `ls *.mps`; do glpsol --mps $f --log ${f/.mps/.log}; done
```

Roteiro

- Baixar os testes da MIPLIB.
- Resolver os problemas para mais de uma combinação de opções do GLPK.

```
--first         branch on first integer variable
--last          branch on last integer variable
--mostf         branch on most fractional variable
--drtom         branch using heuristic by Driebeck and Tomlin
                (default)
--pcost         branch using hybrid pseudocost heuristic (may be
                useful for hard instances)
--dfs           backtrack using depth first search
--bfs           backtrack using breadth first search
--bestp         backtrack using the best projection heuristic
--bestb         backtrack using node with best local bound
                (default)
```

Obrigado!

r.gaia.cs@gmail.com