Frase de Um Japonês:
Quanto menor, melhor!

O Americano chamaria de
One-Liners

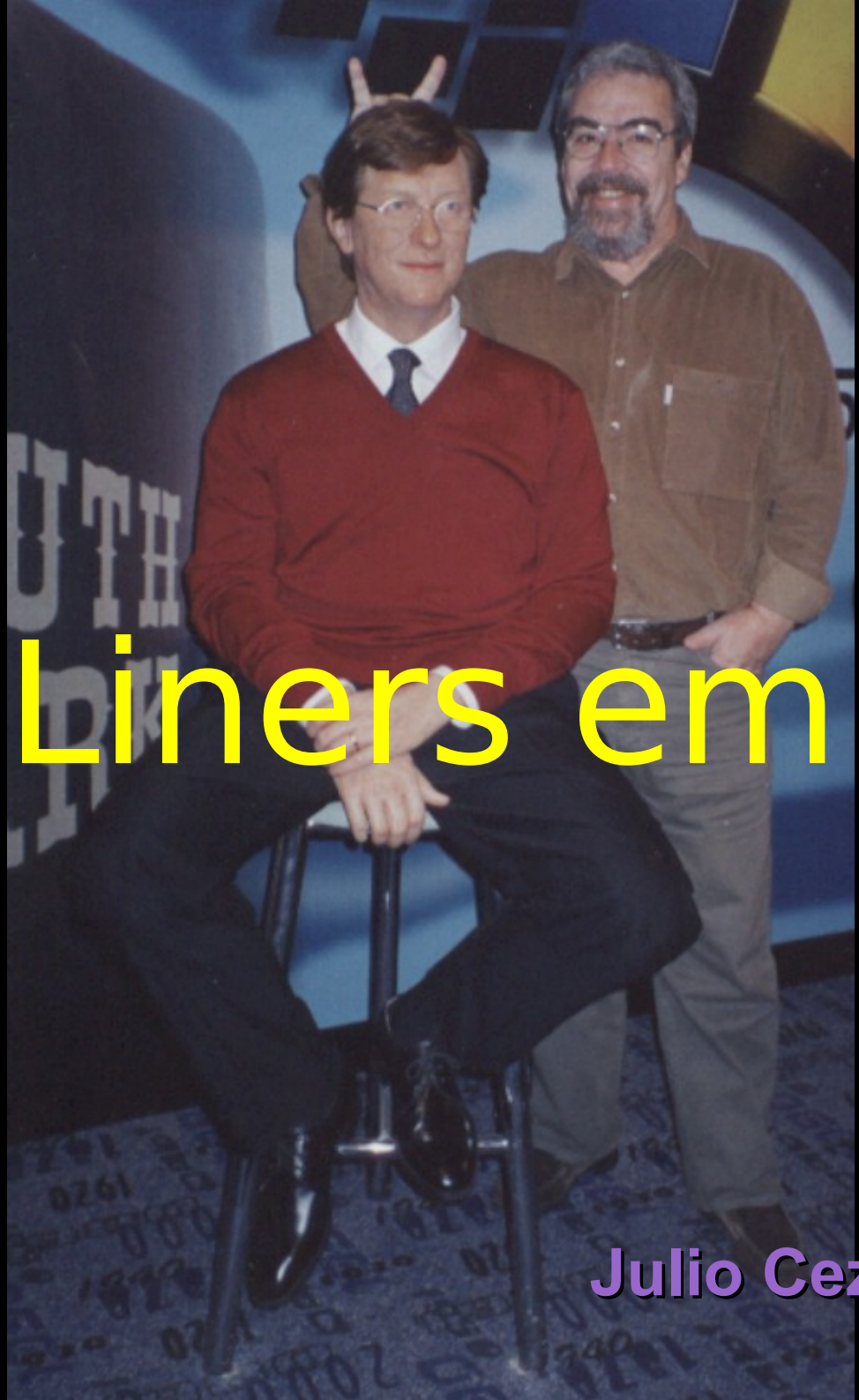
Eu prefiro chamá-lo de

Método KISS

(Keep It Simple Stupid)

Julio Cezar Neves

OneLiners em Shell



Julio Cezar Neves

Como isso começou

O Desafio era fazer o menor programa possível para contar a quantidade de arquivos com cada extensão

```
$ ls | cut -sf2 -d. | sort | uniq -c
      2 log
      1 ods
      1 odt
      3 rtf
      1 sxc
      1 sh
      5 sxi
```

Como isso começou

Logo depois, na excelente lista Shell-Script do Yahoo (http://br.groups.yahoo.com/group/shell-script), aparece a seguinte dúvida:

- Qual a melhor forma de dizer quantas vezes cada vogal aparece em uma frase.

A melhor resposta foi a seguinte:

```
$ echo "uma FrasE muitissimo legAL" |
      tr '[:upper:]' '[:lower:]' |
      grep -o -E '[aeiou]' | sort | uniq -c
3 a
2 e
3 i
1 o
2 u
```

Tiago Barcellos Peczenyj

Substituição de Processos

Você sabe que **só** se pode passar a saída de um comando para a entrada de outro com um *pipe* (|), né?

ERRADO, veja isso:

```
$ cat <(ls)
arq1
arq2
arq3
```

Agora veja esta maluquice:

```
$ ls -l >(cat)
l-wx------  1 jneves jneves 64 2006-05-12 11:34 /dev/fd/63 -> pipe:[15199]
```

Isto é um *file descriptor* de *named pipe*

É um link simbólico, mas não é 777

Aponta para um pipe temporário

Substituição de Processos

E pra que serve esta traquitana? Ahh! Para um monte de coisas, veja:

```
$ ls | while read arq; do let i++; echo $i $arq; done
1 arq1
2 arq2
3 arq3
```

Porém veja só:

```
$ echo $i

$
```

Agora veja com substituição de processos:

```
$ while read arq; do let i++; echo $i $arq; done < <(ls)
1 arq1
2 arq2
3 arq3
$ echo $i
3
```

Substituição de Processos

Então vamos ver um one-liner poderoso usando substituição de processos:

```
$ diff <(ls -l dir) <(ls -l dir.bkp)
```

Ahh! Não tá satisfeito? Então vamos ver de outra forma:

```
$ cmp <(cat dir/*) <(cat dir.bkp/*)
```

Substituição de Processos

Na lista de Shell do Yahoo, um colega mandou a seguinte solicitação: "Eu consigo ver se um arquivo tem uma determinada palavra com o **`grep -i 'palavra chave' arquivo`**. Porém eu queria fazer de forma automática, com que todos os arquivos que **não** contém a **`palavra chave`** fossem excluídos. valew"

Vi a oportunidade do *one-liner* usando substituição de processos e mandei de bate-pronto:

```
$ rm -i $(comm -13 <(grep -li 'palavra chave' *) <(ls))
```

Pensei comigo: "Que maravilha este *one-liner*! Rebentei!" Não passaram 10 minutos até o Tiago Barcellos Peczenyj, um colega de lista, mandasse essa:

```
$ grep -iL 'palavra chave' *
```

O Comando xargs

O comando `xargs` foi feito para remediar um erro comum nos UNIXES antigos que era "*Too many arguments*" gerados pelo comando `find`.

Forma careta:

```
$ find $1 -type f -exec grep -l "$2" {} \;
```

Forma "envenenada":

```
$ cat grepr
#
# Grep recursivo
# Pesquisa a cadeia de caracteres definida em $2 a
# partir do diretorio $1
#
find $1 -type f -print | xargs grep -l "$2"
```

O Comando xargs

O cara resolveu remover todos os seus arquivos do diretório corrente. Como ficaria melhor?

Forma 1:

```
$ find . -user cara -maxdepth 1 -exec rm -f {} \;
```

Forma 2:

```
$ ls -l | grep " cara " | cut -c55- | xargs rm
```

Agora olha esse para remover todos com extensão `.txt`, mostrando seus nomes:

```
$ find . -type f -name "*.txt" | \
  xargs -i bash -c "echo removendo {}; rm {}"
```

A opção `-i` do `xargs` troca pares de chaves (`{}`) pela cadeia que está recebendo pelo *pipe* (`|`).

O Comando xargs

O default do `xargs` é o comando `echo` e a opção `-n` diz a quantidade de argumentos que serão tratados de cada vez.

```
$ seq 5 | xargs -n 2
1 2
3 4
5
```

Olha o *one-liner* aí, gente!!!

```
$ ls dir
arq1.bug
arq1.ok
  ...
arq9.bug
arq9.ok
$ ls | xargs -p -n2 diff -c
diff -c arq1.bug arq1.ok ?...y
  ...
diff -c arq9.bug arq9.ok ?...y
```

A opção `-p` **P**ergunta se você realmente deseja fazer a operação

O Comando paste

Para gerar uma saída tabulada:

```
$ ls arq[1-8] | paste -s -d'\t\t\n'
arq1    arq2    arq3
arq4    arq5    arq6
arq7    arq8
```

Mas... Dá para encolher um pouco?

Claro que dá, veja só:

```
$ ls arq? | paste - - -
arq1    arq2    arq3
arq4    arq5    arq6
arq7    arq8
```

O Comando paste

Para fazer um fatorial:

```
$ seq 4 | paste -sd\*
1*2*3*4
$ seq 4 | paste -sd\* | bc
24
```

Mas... Dá prá encolher um pouco?

Claro que dá, veja só:

```
$ seq -s\* 5
1*2*3*4*5
$ seq -s\* 5 | bc
120
```

Sites imperdíveis:

http://www.julioneves.com
http://www.aurelio.net
http://www.thobias.org
http://br.groups.yahoo.com/group/shell-script

Dúvidas

Meu e-mail
julio.neves@openoffice.org