Série
#LivingInTheShell

Parte #1

Pipes em Pre-Processadores

A Arquitetura do Unix

O Unix é um conjunto de sistemas operacionais que surgiu na década de 60 nos estados unidos sendo um dos primeiros sistemas operacionais
a suportar múltiplas operações em conjunto e paralelo

Alguns exemplos de Sistemas Operacionais que sao derivados do Unix são: MacOS, Linux, FreeBSD, UnixV entre outros.

A Filosofia do Unix

O Unix tem uma filosofia bem clara de desenvolvimento de software e uma dela é a de que:

Os Programas devem se especializar em fazer apenas uma coisa e fazê-la bem, porem, deve ser fácil fazer com que um programa use a saída de outro programa como entrada e vice-versa.

Essa capacidade de se usar programas em conjunto, em que a saída de um eh a entrada de outro, eh uma das maiores vantagens do unix e sera o tema dessa apresentação.

Livro sobre filosofia Unix:
http://www.catb.org/~esr/writings/taoup/html/

Unix no Windows

Desde 2002, eu uso no windows, um shell open-source de unix para windows chamado cygwin. Esse shell na realidade sao os programas Unix **COMPILADOS** e nao **EMULADOS** em windows, oque garante excepcional performance.

Todas as demonstracoes serao feitas em windows usando Cygwin

Dados iniciais para as Demonstracoes

Foram extraídos diversos dados do Multcob - Base 09 de Porto Alegre - ITAUCARD Colchão que serão usados nessa demonstração:

- clientes.csv - Dados de Cliente
- contratos.csv - Dados de Contrato
- telefones.csv - Dados de Telefones
- emails.csv - Dados de Emails

Demonstracao

Obter todos os cpfs únicos e em ordem numérica

```
cat clientes.csv |
awk  -F',' 'NR>1 {print $3}' |
uniq |
sort > clientes_unicos.txt
```

Explicacao

Obter todos os cpfs únicos e em ordem numérica

```
cat clientes.csv
awk  -F',' 'NR>1 {print $3}'
uniq
sort > clientes_unicos.txt
```

Demonstracao

Concatenar os dados de clientes e telefones

```
rm CPF*.txt;rm *.sh;cat
clientes_unicos.txt | awk  -F','
'{print "echo \""$1"\" >>
CPF"$1".txt;echo \"Telefones\" >>
CPF"$1".txt;grep \"^"$1"\"
telefones.csv >> CPF"$1".txt" } ' >>
./processar.sh;chmod 755
./processar.sh;./processar.sh
```

Explicacao

Concatenar os dados de clientes e telefones

- `rm CPF*.txt;rm *.sh`
- `cat clientes_unicos.txt`
- `awk  -F',' '{print "echo \""$1"\" >> CPF"$1".txt;echo \"Telefones\" >> CPF"$1".txt;grep \"^"$1"\" telefones.csv >> CPF"$1".txt" } ' >> ./processar.sh`
- `chmod 755 ./processar.sh`
- `./processar.sh`

Demonstracao

Processar so que múltiplos ao mesmo tempo

```
rm CPF*.txt;rm *.sh;cat clientes_unicos.txt
| awk  -F',' '{print "echo \""$1"\" >>
CPF"$1".txt;echo \"Telefones\" >>
CPF"$1".txt;grep \"^"$1"\" telefones.csv >>
CPF"$1".txt" } ' >> ./processar.sh;chmod
755 ./processar.sh;  cat ./processar.sh |
xargs  -n 1 -P 32 -I % sh -c '%'
```

Explicacao

Processar so que múltiplos ao mesmo tempo

- `rm CPF*.txt;rm *.sh`
- `cat clientes_unicos.txt`
- `awk  -F',' '{print "echo \""$1"\" >> CPF"$1".txt;echo \"Telefones\" >> CPF"$1".txt;grep \"^"$1"\" telefones.csv >> CPF"$1".txt" } ' >> ./processar.sh`
- `chmod 755 ./processar.sh`
- `cat ./processar.sh | xargs  -n 1 -P 32 -I % sh -c '%'`

Demonstracao

Utilizando um programa python para verificar dados no multcob

```
cat clientes_unicos.txt | python -u
esta_em_acordo.py >
status_acordo_clientes.txt
```

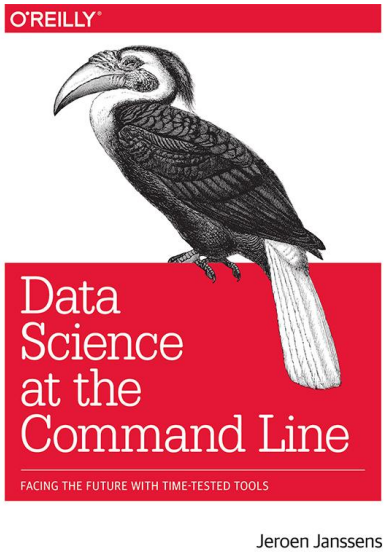
Explicacao

Utilizando um programa python para verificar dados no multcob

```
def esta_em_acordo(cpf):
    conn_str = 'host=\'{}\' dbname=\'{}\' user=\'{}\' password=\'{}\''.format(
    |    POSTGRES_HOST, POSTGRES_DATABASE, POSTGRES_USER, POSTGRES_PASSWORD)
    conn = psycopg2.connect(conn_str)
    cursor = conn.cursor()
    sql = """
    |    |select cod_acordo from acordos where cod_cliente in
    |    |(select cod_cliente
    |    | from clientes where cpf_cnpj='{}') and
    |    | status in (10) """.format(cpf)
    cursor.execute(sql)
    ret = cursor.fetchall()
    return "Tem Acordo Ativo" if len(ret) > 0 else return "Nao Tem Acordo Ativo"

while True:
    cpf = sys.stdin.readline().replace("\n","").replace("\r","")
    if cpf is None or len(cpf) == 0:
    |    break
    print(cpf + "," + esta_em_acordo(cpf))
```

Data Science in the Command Line



https://www.datascienceatthecommandline.com/

(Livro 100% Online e Free)

Mestre Foo e as Dez Mil Linhas

Mestre Foo disse uma vez a um programador visitante: “Há mais natureza Unix em uma linha de script de shell do que em dez mil linhas de C”.
O programador, que estava muito orgulhoso de seu domínio do C, disse: “Como pode ser? C é a linguagem em que o próprio kernel do Unix é implementado! ”
Mestre Foo respondeu: “É verdade. No entanto, existe mais natureza Unix em uma linha de script de shell do que em dez mil linhas de C ”.
O programador ficou angustiado. “Mas, por meio da linguagem C, vivenciamos a iluminação do Patriarca Ritchie! Tornamo-nos um só com o sistema operacional e a máquina, obtendo um desempenho incomparável! ”
Mestre Foo respondeu: “Tudo o que você diz é verdade. Mas ainda há mais natureza Unix em uma linha de script de shell do que em dez mil linhas de C ”.
O programador zombou do Mestre Foo e se levantou para partir. Mas Mestre Foo acenou com a cabeça para seu aluno Nubi, que escreveu uma linha de script de shell em um quadro branco próximo, e disse: “Programador mestre, considere este pipeline. Implementado em C puro, não abrangeria dez mil linhas? ”
O programador murmurou por entre a barba, contemplando o que Nubi havia escrito. Finalmente ele concordou que era assim.
“E quantas horas você precisaria para implementar e depurar esse programa C?” perguntou Nubi.
“Muitos”, admitiu o programador visitante. “Mas só um tolo gastaria tempo para fazer isso quando tantas outras tarefas dignas o aguardam”.
“E quem entende melhor a natureza do Unix?” Mestre Foo perguntou. “É aquele que escreve as dez mil linhas, ou aquele que, percebendo o vazio da tarefa, ganha mérito por não codificar?”
Ao ouvir isso, o programador ficou esclarecido.

Github

https://github.com/gbencke/LivingInTheShell