

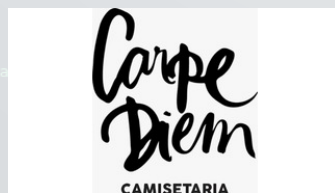


Festival Latino-americano
de Instalação de
Software Livre - 2018

Web Data Mining com BASH Script

Vinícius Silva Madureira Pereira

Apoio Cultural



Arquivos do mini-curso

➤ github.com/ViniciusMadureira/flisol2018

- Shell
- Bourne Again Shell
- Shell Script

Permissões

- chown
- chmod
- chgrp
- groupmems

Copiando da Web

- WGet
- CURL

Variáveis e atribuições

- `PI=3.1415`
- `marca="intel"`
- `endereco=("Avenida C1" 250 "Barretos")`
- `declare -A livro=([titulo]="Box Star Wars"
[paginas]=640 [editora]="Bertrand")`

Entrada e saída

- echo
- printf

Operadores

- aritméticos
- relacionais
- lógicos

Decisão - if

- `if...; then...; fi`
- `if...; then...; else...; fi`
- `if...; then...; elif...; else...; fi`

Repetição - for

- `for valor in 1 2 3; do...; done`
- `for valor in {1..10}; do...; done`
- `for valor in {2..10..2}; do...; done`

Repetição - while/until

- `while...; do...; done`
- `until...; do...; done`

Função

- `function nome_funcao {`
 ...
}

- `nome_funcao() {`
 ...
}

Parâmetros

- \$0, \$1, \$2, \$n
- #
- @

Redirecionadores

➤ >

➤ >>

➤ <

➤ <<

➤ |

Expressões Regulares

- Padrões: `.` `[]` `[^]`
- Âncoras: `^` `$`
- Meta-sequências: `\d` `\D` `\w` `\W`
- Quantificadores: `*` `?` `+` `{n, m}`

Expressões Regulares

- GREP
- AWK
- SEd

Processos ETL

➤ Extract

<td><dfn title="07:41:42 PM GMT">Apr-27-2018</dfn></td>

Extract



07:41:42 PM GMT Apr-27-2018

➤ Transform

Transform



2018-27-04 19:41:42

➤ Load

Load



date
2018-27-04 19:41:42

SQLite

- CREATE / DROP
- INSERT
- SELECT

Referências

- JARGAS, A. M. **Shell Script Profissional**. 1 ed. São Paulo: Novatec, 2008.
- NEGUS, C. **Linux a Bíblia. O Mais Abrangente e Definitivo Guia Sobre Linux**. 1 ed. Rio de Janeiro: Alta Books, 2014.
- SILVA, L. C.; FERRAR, D. G; QUERO, P. **Introdução à Mineração de Dados. Conceitos Básicos, Algoritmos e Aplicações**. 1 ed. São Paulo: Saraiva, 2016.
- CEZAR, J. **Programação Shell Linux**. 10 ed. São Paulo: Brasport, 2014.
- MORIMOTO, C. E. **kurumin 7. Guia Prático**. 1 ed. Paraná: Sulina, 2007.
- PASSOS, E. **Datamining. Conceitos, Técnicas, Algoritmos, Orientações e Aplicações**. 2 ed. Rio de Janeiro: Campus, 2015.



Festival Latino-americano
de Instalação de
Software Livre - 2018

```
#!/bin/bash
#
function extract() {
for page in {1..14}; do
page=$(printf "%02d" $page)
echo "***** $page *****"
table_rows=$(curl --silent "http://nntime.com/proxy-list_$page.htm" | grep --null-data --text --only-matching '<tr class=.*>' | sed -e 's/.*<tbody/<tbody/g')
table_rows=$(echo -e $table_rows | sed 's/<tr /<ntr /g')
IFS=$'\n'
for row in $table_rows; do
row=$(echo $row | sed 's/<td/<n<td/g')
port_length=$(echo $row | sed '3p; d' | grep --perl-regexp --only-match '(\d+)' | sed 's/\+//g' | awk '{print length}')
port=$(echo $row | sed '2p; d' | grep --perl-regexp --only-match '(?=\d+)(\d+)(\d+)' | grep --perl-regexp --only-match "^(.{${port_length}}$)")
ip=$(echo $row | grep --perl-regexp --only-match '(?=\d+)(\d+)(\d+)(\d+)(\d+)(\d+)' | grep --perl-regexp --only-match "^(.{${port_length}}$)")
type=$(echo $row | sed --expression '4p; d' | sed 's/\+//g')
updated=$(echo $row | sed '5p; d' | grep --perl-regexp --only-match '(\d{3})-(\d{1,2})-(\d{4})')
country=$(echo $row | sed '6p; d' | sed --expression 's/\+//g' --expression 's/\+//g')
owner=$(echo $row | sed '7p; d' | sed --expression 's/\+//g' --expression 's/\+//g')
proxy="{\"ip\":$ip,\"port\":$port,\"type\":$type,\"country\":$country,\"owner\":$owner}"
transform
done
done
}

function transform() {
proxy["ip"]=$(echo ${proxy["ip"]} | cut --characters=1-15)
proxy["port"]=$(echo ${proxy["port"]} | cut --characters=1-5)
resultset=$(sqlite3 proxies.db "SELECT ip as proxy_ip, id port as proxy_port, type as proxy_type, country as proxy_country, owner as proxy_owner WHERE proxy_ip != '' AND proxy_port != '' AND proxy_type != '' AND proxy_country != '' AND proxy_owner != ''")
if [[ -z $resultset ]]; then
proxy["type"]=$(echo ${proxy["type"]} | sed --expression 's/\+//g')
proxy["last_update"]=$(date --date ${proxy["last_update"]} +%Y-%m-%d %H:%M:%S)
if [[ ${proxy["country"]} =~ ^\.$ ]]; then
proxy["country"]=$(echo $country | grep --perl-regexp --only-matching '(?!\.)(\d+)' | cut --characters=1-30)
fi
proxy["owner"]=$(echo ${proxy["owner"]} | cut --characters=1-80)
load
fi
}

function load() {
dataset=({"table"="proxy"})
proxy["id"]=$(get_next_id ${dataset["table"]})
dataset=({"table"="port" ["column"]="number" ["signal"]="=" ["value"]=${proxy["port"]})
proxy["port"]=$(get_id)
dataset=({"table"="type" ["column"]="name" ["signal"]="=" ["value"]=${proxy["type"]})
proxy["type"]=$(get_id)
dataset=({"table"="country" ["column"]="name" ["signal"]="LIKE" ["value"]=${proxy["country"]})
proxy["country"]=$(get_id)
dataset=({"table"="owner" ["column"]="name" ["signal"]="=" ["value"]=${proxy["owner"]})
proxy["owner"]=$(get_id)
sqlite3 proxies.db "INSERT INTO proxy (id, ip, last_update, port, id type, id country, id owner) VALUES (${proxy["id"]}, ${proxy["ip"]}, ${proxy["last_update"]}, ${proxy["port"]}, ${proxy["type"]}, ${proxy["country"]}, ${proxy["owner"]})"
}

function get_id() {
id=$(sqlite3 proxies.db "SELECT id FROM ${dataset["table"]} WHERE ${dataset["column"]} ${dataset["signal"]} ${dataset["value"]} LIMIT 1")
if [[ -z $id ]]; then
id=$(get_next_id ${dataset["table"]})
sqlite3 proxies.db "INSERT INTO ${dataset["table"]} VALUES ($id, ${dataset["value"]})"
fi
echo $id
}

function get_next_id() {
echo $(sqlite3 proxies.db "SELECT MAX(id) FROM ${dataset["table"]}")
}

export -f extract
declare -A proxy_dataset
extract
exit 0
```

Obrigado!

Vinícius Silva Madureira Pereira

Apoio Cultural