PERL Practical Basics

GOC Security Scope for UNIX OS (All varieties) into managed servers.

Luciano Trillo Pellizzari / luciano.trillo@hp.com

PERL Practical Basics

Content

Practical Extraction & Report Language

Introduction

- Scope.
- PERL Strengths.
- Requirements & Tools.

Basic Requirements & Practical Solutions.

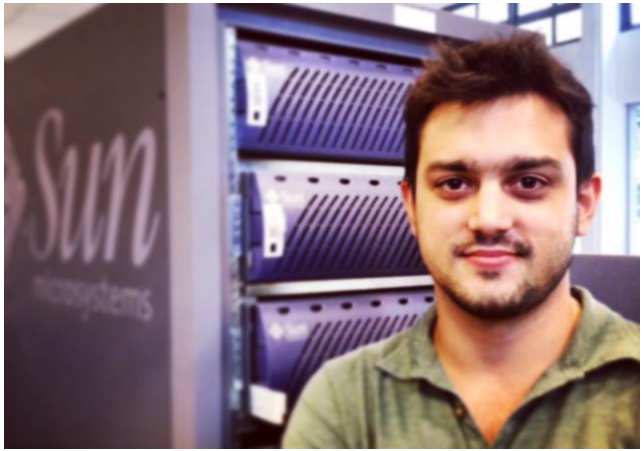
- Related & Common UNIX requirements.
- Line Commands: Search & Replace process inside OS files.
- Line Commands: Insert Process inside OS files (2 Structures).

Introduction

Scope

This document describes useful commands regarding PERL (practical Extraction & Report Language) programming into the UNIX Operative Systems (applicable to all varieties).

During this course, you will learn important details and ways to accelerate the file data change/fix procedure through the server system console, inside the supported UNIX Servers, managed by the HP Technical Specialists.



Introduction

PERL Strengths

- It's very **easy to learn**, and learning a little PERL can get you a long way.

- Is very **portable**. PERL is available for a huge variety of operating systems and computers, and properly written programs should run almost anywhere that PERL does without any change.

- It thinks about **words and sentences**, where other languages see the character at a time. It’s 'regular expressions' allow you to search for and transform text in innumerable ways with ease and speed.

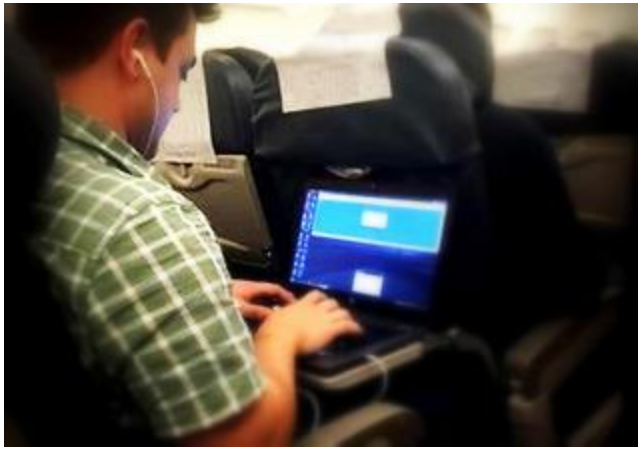
- Is what is termed a '**high-level language**'. Some languages concern you with unnecessary, 'low-level' details about the computer's operation. PERL cuts you free from all this.

Introduction

Requirements & Tools

- Authorized access to the supported servers from HP.

- Installed UNIX server system access console ‘Reflection X’.



Basic Requirements & Practical Solutions

Related & Common UNIX requirements.

- UNIX File Data Update:

  Insert/replace/delete (CRUD) particular information (parameters, values, etc.) into the servers UNIX OS Files without the necessity to get into the servers files to complete that needed change action (ie. By using the “vi” editor).

Basic Requirements & Practical Solutions

Line Commands: Search & Replace process inside OS files.

- Basic Requirements & Commands Solutions (Theory):

Structure:

find **/theDirectory/theFile** | perl -pi -e 's/**existingText**/**theNewText**/'

get into into the chosen file *search* *&* *replace*

Basic Requirements & Practical Solutions

Line Commands: Search & Replace process inside OS files.

- Basic Requirements & Commands Solutions (Practice):

Example 1: *Activate the "MINUPPER" parameter at the "/etc/default/passwd" file (Uncomment "# " entry).*

find **/etc/default/passwd** | perl -pi -e 's/**#MINUPPER/MINUPPER/**'

get into into the chosen file *search & replace*

TIP 1: It can be "/# MINUPPER/" if the text includes a space after the comment (#) entry.

TIP 2: If a specific word or letter must be deleted, into the 'replace' field do not include anything (//).

Basic Requirements & Practical Solutions

Line Commands: Search & Replace process inside OS files.

- Basic Requirements & Commands Solutions (Practice):

Example 2: *Set up the value "1" into the "MINUPPER" parameter to satisfy the current security settings.*

find **/etc/default/passwd** | perl -pi -e 's/**MINUPPER=[0-9]*/MINUPPER=1/**'

get into into the chosen file *search* *&* *replace*

TIP 3: It can be "/MINUPPER=[0-9]." with a dot and asterisk (.*), to clean-up any text and/or space after.*

TIP 4: Other search options are [a-z]; [A-Z]; [0-9-] with a dash at the end (-) to include special characters).

Basic Requirements & Practical Solutions

Line Commands: Search & Replace process inside OS files.

- Basic Requirements & Commands Solutions (Practice):

Implementation Example (Commands):

1) Back-Up creation (as a security matter, a suggestion is to always perform a back-up process).
2) Implementation change (apply the PERL commands according to the needful).
3) Data verification (Review/Check the changes made if everything is correct).

```
theDate=`date '+%m-%d-%y'`; cp -p /etc/default/passwd /etc/default/passwd.$theDate
find /etc/default/passwd | perl -pi -e 's/#MINUPPER=[0-9]*/MINUPPER=1/‘
echo "";grep MINUPPER= /etc/default/passwd;echo ""
```

Basic Requirements & Practical Solutions

Line Commands: Insert Process inside OS files (2 Structures).

- Basic Requirements & Commands Solutions (Theory):

Structure 1:

perl -pi -e 'print "**theNewText**\n" if $. == FileLineNumber' **/theDirectory/theFile**

text to be inserted *file line number* *the file chosen*

Basic Requirements & Practical Solutions

Line Commands: Insert Process inside OS files (2 Structures).

- Basic Requirements & Commands Solutions (Practice):

Structure 1 (Example): *Insert the "# HP Argentina" phrase at the top of the "/etc/profile" file.*

perl -pi -e 'print "**# HP Argentina**\n" if $. == **1**' **/etc/profile**

text to be inserted *line number* *the file chosen*

TIP 5: "\n" is a formatting character to add a new line.

Basic Requirements & Practical Solutions

Line Commands: Insert Process inside OS files (2 Structures).

- Basic Requirements & Commands Solutions (Theory):

Structure 2:

perl -i -pe 'chomp,$_.="\n**theNewText**" if **/theReferenceText/**' **/theDirectory/theFile**

text to be inserted below the reference *the reference text* *the file chosen*

Basic Requirements & Practical Solutions

Line Commands: Insert Process inside OS files (2 Structures).

- Basic Requirements & Commands Solutions (Practice):

Structure 2 (Example): *Insert the "# GOC Security Module" phrase below the "# HP Argentina" phrase into the "/etc/profile" file.*

perl -i -pe 'chomp,$_.="\n**# GOC Security Module**" if **/# HP Argentina/' /etc/profile**

text to be inserted below the reference *the reference text* *the file chosen*

Basic Requirements & Practical Solutions

Line Commands: Insert Process inside OS files (2 Structures).

- Basic Requirements & Commands Solutions (Practice):

Implementation Example (Commands):

1) Back-Up creation (as a security matter, a suggestion is to always perform a back-up process).
2) Implementation change (apply the PERL commands according to the needful).
3) Data verification (Review/Check the changes made if everything is correct).

```
theDate=`date '+%m-%d-%y'`; cp -p /etc/profile /etc/profile.$theDate
perl -pi -e 'print "# HP Enterprise Services\n" if $. == 1' /etc/profile
perl -i -pe 'chomp,$_.="\n# GOC Security Module\n" if /# HP Enterprise Services/' /etc/profile
head -5 /etc/profile
```

Q&A

Luciano Trillo Pellizzari

Thank you

Special Gratefulness:

Nicolas Gallia

Luciano Guadagnini