Лабораторна робота №1. Shell

Мета

Метою виконання цього комп'ютерного практикуму є знайомство з середовищем користувача ОС сімейства Unix.

В результаті його виконання буде отримане загальне уявлення про інтерфейс взаємодії з системою, вивчені основні команди командної оболонки, отримані навички написання скриптів командної оболонки.

Завдання

Необхідно написати shell-скрипт, який опрацює текстовий файл log.txt у форматі *Acess log* веб-сервера Apache і виведе в консоль інформацію, яка описана в окремому документі **"Варіант завдання по КП1"**. Цей скрипт повинен використовувати стандартні інструменти shell (які включають такі утиліти, як: `cut`, `grep`, `sort`, `awk`, `date` та ін.), але не використовувати інші мови програмування, такі як C, Perl, Python та ін.

Файл складається з записів, кожен з яких займає один рядок.

Приклад запису:

```
host-24-225-218-245.patmedia.net - [01/Oct/2006: 6:33:45
-0700] "GET / example / example.atom HTTP/1.1" 304 - "-"
"NetNewsWire / 2.0b37 (Mac OS X; Lite;
http://ranchero.com/netnewswire/) "
```

Формат запису:

```
Хост клієнта - [Штамп часу з часовою зоною] Рядок HTTP-
запиту (тип, URL, версія) Код HTTP-відповіді Кількість
переданих байт або '-', якщо відповідь не має тіла Рядок
реферера ('-' означає прямий запит без реферера) Назва
клієнта (браузер)
```

Розшифровка:

01/Oct/2006: 6:33:45 -0700 з хоста **host-24-225-218-245.patmedia.net** по протоколу **HTTP/1.1** був здійснений запит типу **GET** для отримання ресурсу, що перебуває по посиланню **/example/example.atom**. Код відповіді на запит від сервера: **304**. Така відповідь не передбачає наявності тіла відповіді (кількість

* pdf generated on 2016-01-03 from markdown sources with SBCL 1.2.16 on Linux 3.16.0-38-generic with 3bmd, cl-pdf & cl-typesetting

переданих байт - **0**). Запит виконувався безпосередньо, а не по посиланню з іншого сайту (поле реферер - порожнє). Кліент використовував для звернення програму **NetNewsWire/2.0b37**, ОС клієнта: **Mac OS X**

Пояснення

Shell-скрипт - це програма, написана для інтерпретації командною оболонкою ОС (shell). Ця програма існує в текстовому вигляді і не потребує окремого етапу компіляції перед виконанням. За угодою, першим рядком скрипта є вказівка конкретного інтерпретатора, який повинен виконувати його. Взагалі кажучи, в Unix-ОС скрипти не обов'язково повинні виконуватися саме командною оболонкою, а можуть бути написані на будь-якій мові, яка підтримує інтерпретацію (наприклад, Perl або Python).

Shell-скрипт Hello World:

```
#!/bin/sh
echo "Hello World"
```

Якщо цей текст зберегти в файл hello.sh в поточній директорії, то виконати його можна двома способами:

- `$ sh hello.sh` - в цьому випадку ми запускаємо команду `sh` (власне, shell) і передаємо їй як аргумент ім'я файлу скрипта
- `$ chmod +x hello.sh; ./hello.sh` - в цьому випадку виконуються 2 команди: спочатку файлу скрипта дається право на виконання, потім запускається сам файл і командна оболонка, в якій виконується ця команда, аналізує початок файлу. Якщо це скомпільована програма, то вона містить у перших байтах т.зв. магічний номер (magic number), який ідентифікує формат виконуваного файлу, в який вона скомпілірованна - в цьому випадку shell передає управління завантажувачу програми, який підтримує відповідний формат. Якщо це скрипт і він містить т.зв. shebang-рядок (рядок, що починається з символів `#!`), то весь його вміст передається програмі, шлях до якої зазначений в цьому рядку (в даному випадку: `/bin/sh`). Інакше програма вважається скриптом самої командної оболонки і виконується нею самою.

Примітка 1: `$` - запрошення консолі ОС. Такий запис означає, що команду потрібно виконати в консолі (без самого знака $).

Примітка 2: програмна оболонка по-замовчуванню в середовищі Linux - `bash` (Bourne Again Shell). У даному прикладі в якості інтерпретатора зазначений `sh` (Bourne Shell), який є більш простим варіантом командної оболонки,

* pdf generated on 2016-01-03 from markdown sources with SBCL 1.2.16 on Linux 3.16.0-38-generic with 3bmd, cl-pdf & cl-typesetting

попередником bash. У випадках, коли в скрипті не використовуються специфічні розширення bash, правилом хорошого тону є вказувати в якості інтерпретатора саме sh (з міркувань більшої переносимості коду: не на всіх системах може бути встановлений bash).

Shell-скрипти для Boune Shell і її варіантів можуть використовувати ті ж самі команди, які можна вводити і з консолі операційної системи. Команда man - дозволяє отримати довідку по будь-якій команді. $ man sh дозволить вам вивчити синтаксис самого Shell. Важливими операторами Shell є перенаправлення виводу (>) і введення (<), а також конвеєр pipe (|), який дозволяє перенаправляти вивід однієї програми на введення іншої.

На рівні з вбудованими командами Shell може бути запущена будь-яка програма, якщо вона знайдена в пошуковому шляху PATH, або ж до неї зазначений повний шлях. PATH - це одна з змінних оточення Shell, яка доступна всім процесам. Отримати її значення можна так: $PATH, а встановити (в sh / bash): export PATH=.... Дізнатися значення всіх змінних оточення можна командою env. Серед змінних оточення є кілька спеціальних змінних, які встановлюються індивідуально для кожного запущеного процесу. Наприклад, це змінна $*, в якій містяться всі аргументи, з якими запущена дана програма, чи $1, $2, ..., які містять перший, другий і т.д. аргумент.

PATH не включає поточну директорію, тому запуск програми з поточної директорії, як правило, виконується за допомогою синтаксису ./ (Тобто $ ./program). Точка в Shell - це псевдонім для поточної директорії. Іншими псевдонімами є .. - директорія на рівень вище, і ~ - домашня директорія. Команди і програми можуть приймати рядкові аргументи, які кожна з них може інтерпретувати по-своєму. Як правило, ці аргументи бувають 3-х типів:

- Просто значення (числа, рядки), наприклад в $ echo "Hello World" "Hello World" - це просто рядок
- Шляхи, наприклад в $ cat hello.txt hello.txt - це шлях до файлу у поточній директорії. Повний шлях міг би виглядати так: /home/user/hello.txt
- Аргументи-ключі: починаються з - або --, наприклад в $ wc -l file.txt команда wc рахує кількість тільки рядків (ключ -l) у файлі file.txt. Ключ --help дозволяє отримати коротку довідку по переважній більшості команд

* pdf generated on 2016-01-03 from markdown sources with SBCL 1.2.16 on Linux 3.16.0-38-generic with 3bmd, cl-pdf & cl-typesetting

Література

Основи роботи з Linux

- http://matt.might.net/articles/basic-unix/
- http://www.funtoo.org/Linux_Fundamentals,_Part_1

Основи роботи з Bash

- http://ods.com.ua/koi/unix/bash-conspect.html
- http://www.tldp.org/LDP/abs/html/
- http://www.davidpashley.com/articles/writing-robust-shell-scripts.html
- http://mywiki.wooledge.org/BashFAQ
- Bash by Example
- Google Shell Style Guide

Робота з текстовими даними

- http://www.ibm.com/developerworks/aix/library/au-unixtext/index.html
- http://radar.oreilly.com/2011/04/data-hand-tools.html
- http://www.pement.org/awk/awk1line.txt
- http://sed.sourceforge.net/sed1line.txt
- http://www.drbunsen.org/explorations-in-unix.html

Регулярні вирази

- http://www.regular-expressions.info/
- http://www.weitz.de/regex-coach/

Просунуті інструменти командного рядка

- http://www.commandlinefu.com/commands/browse/sort-by-votes
- http://www.quora.com/Linux/What-are-some-time-saving-tips-that-every-Linux-user-should-know

* pdf generated on 2016-01-03 from markdown sources with SBCL 1.2.16 on Linux 3.16.0-38-generic with 3bmd, cl-pdf & cl-typesetting

- http://www.reddit.com/r/linux/comments/mi80x/give_me_that_one_command_you_wish_you_knew_years/
- http://kkovacs.eu/cool-but-obscure-unix-tools
- http://offbytwo.com/2011/06/26/things-you-didnt-know-about-xargs.html
- http://www.tldp.org/HOWTO/Bash-Prompt-HOWTO/

* pdf generated on 2016-01-03 from markdown sources with SBCL 1.2.16 on Linux 3.16.0-38-generic with 3bmd, cl-pdf & cl-typesetting