

Actividad 4: Introducción a la programación de los intérpretes de comandos

Melissa Matrecitos Avila

21 de Febrero de 2018

1 Introducción

El siguiente texto es el reporte de la actividad 4 del curso de Física Computacional 1, en la cual se trabajó con un interprete de comandos llamado Shell, el es una interfaz de usuario para acceder a los servicios de un sistema operativo, en este caso LINUX.

En esta actividad trabajamos las distintas formas de interactuar y hacer scripts (programas) con el Shell (/bin/sh) y el Again Shell (/bin/bash). Para llevarla a cabo se utilizaron los comandos cat, chmod, echo, grep, less, ls y wc, los cuales se explicaran en los resultados de la práctica.

Por último se realizó una síntesis sobre las notas de Steve Parker "Shell Script Tutorial", de los cuales se incluyen ejemplos de las secciones que allí vienen.

2 Síntesis

En esta sección se presentará la síntesis sobre las notas de Steve Parker "Shell Script Tutorial" con algunos ejemplos.

2.1 Introducción y Filosofía

Este tutorial está escrito para ayudar a las personas a comprender algunos de los conceptos básicos de la programación de scripts de shell. Hay una serie de factores que pueden incluirse en scripts de shell buenos, limpios y rápidos.

- Los criterios más importantes deben ser un diseño claro y legible.
- La segunda es evitar comandos innecesarios.

Un diseño claro hace la diferencia entre un guión de shell que aparece como "magia negra" y uno que se mantiene y comprende fácilmente.

2.2 Un Primer Script

Para el primer script se escribirá uno que diga "Hello World", esto se logra, primero que nada especificando por quien debe ser ejecutado el script y con el comando echo se escribe en la pantalla los argumentos que lo acompañan. Es importante saber que echo pondrá automáticamente un espacio entre los argumentos, para evitarlo, el argumento se debe escribir entre comillas, otra cosa importante es saber que el símbolo # es para agregar comentarios, excepto cuando es acompañado de ! ya que #! significa que lo que sigue debe ser interpretado por el shell de Bourne.

```
#!/bin/bash
#Comentario
echo Hello World

#!/bin/bash
#Comentario
echo"Hola    Mundo"
```

2.3 Variables

Casi todos los lenguajes de programación existentes tienen el concepto de variables, un nombre simbólico para un trozo de memoria al que podemos asignar valores, leer y manipular sus contenidos. Tenga en cuenta que no debe haber espacios alrededor del signo "=", el cual tarta el comando como una asignación de variable.

Al shell no le importan los tipos de variables; pueden almacenar cadenas, enteros, números reales, cualquier cosa que te guste. Podemos establecer de manera interactiva nombres de variables usando el readcomando.

No es necesario declarar las variables en el shell Bourne, como sí lo hacen en lenguajes como C. Pero si intenta leer una variable no declarada, el resultado es la cadena vacía. No obtienes advertencias o errores.

```
#!/bin/bash
echo "Hola, ¿Cómo te llamas?"
read Nombre
echo "¿Cuál es tu color favorito, $Nombre?"
read Color
echo "El color favorito de $Nombre es $Color"

#!/bin/bash
echo "Hola, para saber el año en el que naciste escribe el año actual:"
read Actual
echo "¿Cuál será tu edad este año?"
read Edad
echo "El año en que naciste es"
expr $Actual-$Edad
```

2.4 Comodines

Los comodines no son muy útiles en los scripts de shell. Esta sección es realmente solo para mover y editar archivos en la terminal. Son más utilizados cuando hay ciclos.

```
#!/bin/bash
#Copiar archivos de a hacia b
cp / tmp / a / * / tmp / b /

#!/bin/bash
#Escribir lo que contiene la carpeta a
echo /tmp/a/*
```

2.5 Caracteres de Escape

La mayoría de los caracteres (*, ', etc) no se interpretan (es decir, que se toman literalmente) a menos de que sean colocados entre comillas dobles ("). Pero para los caracteres ", ', *y siguiendo interpretado como código a pesar de las comillas, para esto se utiliza el carácter de escape* el cual hace que el intérprete no los interprete, sino que los pase al comando que se esta utilizando.

```
#!/bin/bash
echo "Tienes que pagar \$ 10. "
```

```
#!/bin/bash
echo "Mi nombre es \"Melissa\" "
```

2.6 Ciclos

La mayoría de los lenguajes tienen el concepto de bucles: si queremos repetir una tarea veinte veces, no queremos tener que escribir el código veinte veces, con un ligero cambio cada vez. Como resultado, tenemos `for` y `while` bucles en el shell Bourne.

Los bucles "for" iteran a través de un conjunto de valores hasta que se agote la lista:

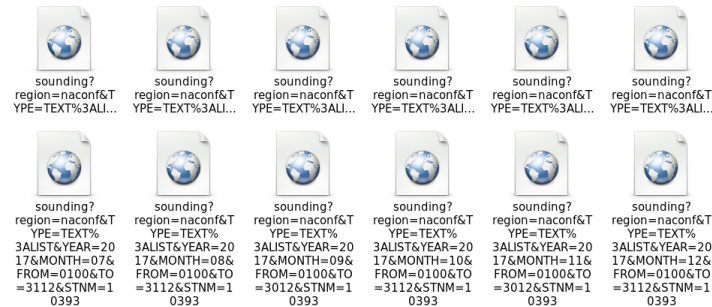
```
#!/bin/bash
for i in 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
do
    echo "Summer"
done
echo "What time is it ? Summertime!"
```

Los bucles "while" iteran hasta que se cumpla una condición de salida, éste tipo de bucles se dice que son más "divertidos" debido a que el usuario decide si va a seguir dentro o va a salir del ciclo:

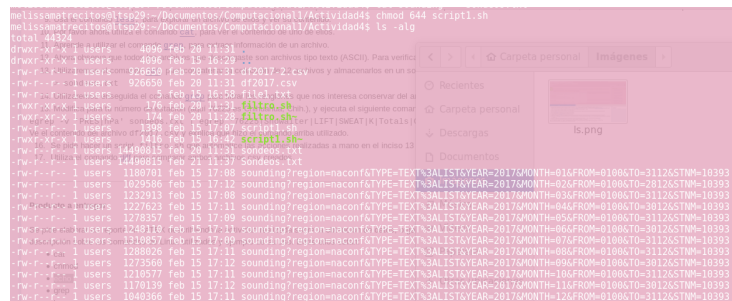
```
#!/bin/bash
salida=Halo
while [ "$salida" != "Tschüss" ]
do
    echo "Escribe algo (Tschüss para salir)"
    read salida
    echo "Escribiste: $salida"
done
```

3 Resultados de la práctica

A continuación se describirá la practica realizada: Después de crear la carpeta "Actividad 4" en la cual se guardarían todos los archivos utilizados, se descargó el archivo proporcionado por el profesor llamado "script1.sh", en el cual se hicieron los cambios pertinentes para descargar los datos de la estación 10393, correspondiente a Linderburg en Alemania. Después con el comando **"chmod"** se creó el ejecutable de dicho script que se confirmo utilizando el comando **"ls -alg"**, que después se ejecutó en la terminal. Los 12 archivos descargados fueron:



Archivos descargados por el script1.sh



Comandos "chmod" y "ls -alg" en la terminal

Posteriormente con el comando **"less"** se exploró el archivo correspondiente al mes de febrero, este mostró el archivo en la terminal, ubicando el puntero al inicio del archivo:

```

melissamatrecitos@lisp29:~/Documentos/Computacional/Actividad4$ less soundings7region\wmoconf\&TYPE=TEXT\3ALIST\&YEAR=2017\&MONTH=02\&FROM=0100\&TO=2012\&STMP=10393
melissamatrecitos@lisp29:~/Documentos/Computacional/Actividad4$

<HTML>
<TITLE>University of Wyoming - Radiosonde Data</TITLE>
<LINK REL="StyleSheet" HREF="/resources/select.css" TYPE="text/css">
<BODY BGCOLOR="white">
<H2>10393 Lindenberg Observations at 00Z 01 Feb 2017</H2> 4 (2018-1)
<PRE>
... Actividad a realizar ...
PRES HGHT TEMP DWPT RELH MIXR DRCT SKNT THTA THTE THTV
hPa Para imilar esta l ciudad, c a en tu sistema g/kg mpa (dega "Acknotd 4". K K K
-----
1007.0 datos 112.5 1:0.0 vros n 1.5 sales de 96 o 203.42 60 4 271.7 281.1 272.3
1000.0 Vamc166 elecc 1.3 la es 1.8 n de im 96 del 3.37 - la U. 55 Wyoming 6 v271.9 m 281.1 m 272.4
925.0 Edita 829 2.0 2.7 a hacer s mbic 3.40 produce 25 alor en 10 276.8 m 286.3 m 277.5
920.0 Guard 898 2.0 2.9 a carpeta de Archivos 4. Eje 36 el com 277.5 280.0 277.6
912.0 Cam 898 2.0 2.9 a carpeta de Archivos 4. Eje 36 el com 277.5 280.0 277.6
883.0 64 1155 - 2.5 - 2.5 permisos de sea ejecutable (-rwx-r-x-r-x). El com 280.4 280.0 281.1
870.0 1273 pt - 1.9 - 2.2 permisos de sea ejecutable (-rwx-r-x-r-x). El com 280.4 280.0 281.1
862.0 Eje 1346 y obs 1.5 como 2.0 descarga de 3.85 a dia (7 d para un 14 d) 283.4 294.5 284.1
850.0 Una 1458 oncluzo 11 proc 2.7 ejecuta 96 mas 0.93 - al 80 para ver q 14 n 283.9 294.6 284.1
832.0 Usa 1627 mando 3.0 s, p 3.5 plorar u 96 chiv 3.55 los y v 80 u conte 16. 284.7 295.0 285.4
814.0 Por 1799 hora 3.0 s el c 4.4 do cat 96 v 3.40 atenido 80 uno de 14 s 285.6 295.5 286.2
787.0 2065 114 5.3 como 5.8 rep. p 96 extra 3.18 mación 60 un arch 12 286.9 296.2 287.4
772.0 2216 - 0.6 - 6.5 - 97 3.06 85 19 287.6 296.6 288.1
754.0 2402 - 7.0 - 7.5 - 97 2.91 80 14 288.5 297.1 289.8
13. Utilizaremos el comando cat para concatenar la colección de 12 archivos y almacenarlos en un solo archivo que llamare

```

Comando less

Siguiendo con la actividad se utilizó el comando **"cat"** par explorar un archivo, a diferencia del less, este te mostraba rápidamente el archivo y dejaba el puntero al final del archivo:

```

melissamatrecitos@lisp29:~/Documentos/Computacional/Actividad4$ cat soundings7region\wmoconf\&TYPE=TEXT\3ALIST\&YEAR=2017\&MONTH=02\&FROM=0100\&TO=2012\&STMP=10393
melissamatrecitos@lisp29:~/Documentos/Computacional/Actividad4$

<FORM>
<INPUT CLASS="button" TYPE="button" VALUE="Close this window"
onClick="window.close();">
<INPUT CLASS="button" TYPE="button" VALUE="Select another map."
onClick="window.blur();">
</FORM>
<HR SIZE="1">
<I>Interested in graduate studies in atmospheric science?
Check out our program at the de interés del sitio de la U. de Wyoming y vamos a anotar el número de la estación seleccio
<a href="http://www.uwyo.edu/atsc/howtoapply/">de el valor en la variable STATION, de el caso que estudias.
target= topUniversity of Wyoming de Actividad 4. Ejecuta el comando: ls -alg y observa los permisos del archivo
</a></I>
<HR SIZE="1"><FONT SIZE="1">
Questions about the weather data provided by this site can be
addressed to A HREF="mailto:ldoolman@uwyo.edu" (será un poco).
Larry Doolman (ldoolman@uwyo.edu)</A></FONT>
<HR SIZE="1">
<SCRIPT TYPE="text/javascript">at, para ver el contenido de uno de ellos.
<!-- 11. Aprende a utilizar el comando grep para extraer información de un archivo.
window.focus() para que todos los archivos que descargaste son archivos tipo texto (ASCII). Para verificar esto, utiliza el comando
// -->
</SCRIPT>
</BODY>
</HTML>
melissamatrecitos@lisp29:~/Documentos/Computacional/Actividad4$

```

Comando cat

Otro comando utilizado fue el **"echo"**, el cual sirvió para escribir texto en un archivo, en este caso el archivo "file1":

```

melissamatrecitos@lisp29:~/Documentos/Computacional/Actividad4$ wc soundings7region\wmoconf\&TYPE=TEXT\3ALIST\&YEAR=2017\&MONTH=02\&FROM=0100\&TO=2012\&STMP=10393
14400 134200 1029000 soundings7region\wmoconf\&TYPE=TEXT\3ALIST\&YEAR=2017\&MONTH=02\&FROM=0100\&TO=2012\&STMP=10393
melissamatrecitos@lisp29:~/Documentos/Computacional/Actividad4$
melissamatrecitos@lisp29:~/Documentos/Computacional/Actividad4$ cat soundings7region\wmoconf\&TYPE=TEXT\3ALIST\&YEAR=2017\&MONTH=02\&FROM=0100\&TO=2012\&STMP=10393
melissamatrecitos@lisp29:~/Documentos/Computacional/Actividad4$

```

Comando echo

También se utilizó el comando **"wc"**, con este contamos las palabras de un archivo:

Comando wc

5 Bibliografía

- Shell (computing). (2018). En.wikipedia.org. Recuperado el 20 Febrero 2018, de [https://en.wikipedia.org/wiki/Shell_\(computing\)](https://en.wikipedia.org/wiki/Shell_(computing))
- Parker, S. (2017). Shell Scripting Tutorial. Shellscrip.sh. Recuperado el 21 Febrero 2018, de <https://www.shellscrip.sh/index.html>
- Atmospheric Soundings. (2018). Weather.uwyo.edu. Recuperado el 21 Febrero 2018, de <http://weather.uwyo.edu/upperair/sounding.html>

6 Apendice

1. ¿Qué fue lo que más te llamó la atención en esta actividad?
La manera tan sencilla en que se pueden automatizar procesos que manejan gran cantidad de datos ya que solo conocía comandos muy básicos.
2. ¿Qué consideras que aprendiste?
Como automatizar procesos por medio de scripts, así como los comandos que estos utilizan.
3. ¿Cuáles fueron las cosas que más se te dificultaron?
Al inicio batallé mucho para saber que era lo que estaba haciendo, ya que no sabía para que eran los comandos. Creo que esto se debe a que un nuevo "lenguaje".
4. ¿Cómo se podría mejorar en esta actividad?
Me hubiera gustado una introducción teórica a lo que íbamos hacer, para no estar tan perdida y así poder agilizar el desarrollo de la actividad.
5. ¿En general, cómo te sentiste al realizar en esta actividad?
Al inicio estaba muy confundida, pero al final me gustó mucho todo lo que hicimos.