

Datos Geoespaciales - 513376

R. Cifuentes Lobos ¹ N. Soto Delis ¹ Franco Retamal ¹

¹Departamento de Geofísica
Universidad de Concepción
Concepción, Chile

Scripts en Bash

Temario.

Introducción a los scripts.

Características.

- Archivo de texto con una secuencia de comandos.
- Por convención tienen una extensión `.sh`
- Se ejecutan anteponiendo `./` a su nombre.
- No son sensibles a indentación (es recomendable indentar para mantener orden en script).

Nociones básicas.

Comienzo de scripts (hash-bang):

- `#!/bin/bash`

Comentarios:

- Anteponer `#`

Creación de nuevos archivos:

- `output > nuevo_archivo.txt`

Anexar a archivos:

- `output >> archivo.txt`

Variables

Definición de Variables

- `var="texto"`
- `var=numero` (ej. 1, 2, 3...)
- `var=$(output)`
- `array = (v1 v2 v3 v4 ... vn)`

Llamado de Variables

- `$var`
- `echo $var`
- `echo ${array[*]}`

Interacción con el usuario.

Input y output.

- Ingreso de valores
 - read
 - echo

Sintaxis.

```
#!/bin/bash  
echo ingresar variable  
read variable  
echo $variable
```

Aritmética en bash.

Cuadro: Operadores matemáticos.

Sintaxis	Significado
$a+b$, $a-b$	Adición, sustracción
$a*b$, a/b	Multiplicación, división
$a \% b$	Módulo
$a**b$	Potencia

Operadores.

Cuadro: Operadores Numéricos.

Operador	Singnificado
-eq	Igual que
-ne	Distinto que
-gt	Mayor que
-ge	Mayor o igual que
-lt	Menor que
-le	Menor o igual que

Cuadro: Operadores de texto.

Operador	Singnificado
==	Igual que
!=	Distinto que

Ciclos y sentencias.

Tipos.

- Ciclos (*loops*).
 - for
 - while
- Sentencias (*statements*).
 - if
 - case
- Estos pueden ser combinados (*nesting*).

if.

- Sintaxis.

```
if [ $var -cond valor ]
    then
        Comandos
    elif [ $var2 -cond2 valor2 ]
    then
        Comandos
    else
        Comandos
fi
```

for.

- Sintaxis.

```
for VARIABLE in Secuencia
do
    comando1
    comando2
    ...
    comandoN
done
```

Secuencias.

Formas de definir secuencias y listas.

- 1 2 3 4 ...
- {1..2}
- {1..10..2}
- ((i=1; i<=5; c++))
- \$(ls ...)

case.

- Sintaxis.

```
case $var in
1)
Accion si var vale 1
;;
2)
Accion si var vale 2
;;
*)
Accion en otros casos
;;
esac
```

while.

- Sintaxis.

```
while [ $var -cond valor ]  
do  
    Comandos  
done
```

Buscar si un valor es primo.

```
#!/bin/bash
#Corre un loop x de 2 a 10
for x in {2..10}
do

#Define una variable $primo
# para ser igual a 1
primo=1

#Define y igual a 2
y=2

while [ $y -lt $x ]
do

#Calcula el resto de la division de x/y
resto=$((x%$y))

# Si el resto es cero para cualquier par
# de ($x,$y) cambia $primo a cero
if [ $resto -eq 0 ]
```

```
then
primo=0
fi

#Aumenta el valor de y por 1
y=$((y+1))
#Termina la condicion de "while"
done

# Si $primo es 1, ningun valor de la
# division tuvo resto cero, luego
# $x es primo
if [ $primo -eq 1 ]
then
echo $x "es primo"
fi

#Termina el loop de "for"
done
```

- 1 Cree un programa que acepte de entrada un número y determine si es par o impar.
- 2 Defina un array con su nombre completo (de la forma Nombre1 Nombre2 Apellido1 Apellido2), luego, imprima en pantalla diferentes frases del tipo "Mi nombre es: (Parte del array con su nombre)", "Mis apellidos son: (Parte del array con sus apellidos)".
- 3 Cree un programa que acepte dos números (uno más grande que el otro) y que entregue en terminal una lista con los números impares entre ellos.
- 4 Cree un programa que cuente la cantidad total de directorios que tenga en \$home.
- 5 Cree un programa que cuente la cantidad de archivos .txt que tenga un directorio de su elección.
- 6 Cree un programa que organice sus archivos en diferentes carpetas según se extensión.