

# Actividades prácticas de la Shell de Linux

Informática II – UTN-FRC

Gonzalo F. Perez Paina

## Ejercicios

1. Utilizar el comando `echo` para imprimir el valor de las variables de entorno `USER`, `PATH`, `SHELL`, `PS1`, `PS2`, `PWD`, `OLDPWD`. ¿Qué información contiene cada variable de entorno?
2. Buscar información sobre las variables de entorno `PS1` y `PS2` y probar de modificarlas.
3. Codificar el script del listado 1, el cual imprime algunas variables de entorno, darle permiso de ejecución y ejecutar de las sig. formas:  
> `./envvar.sh`  
> `./envvar hola`  
> `./envvar hola 123`

Listing 1: Script de shell para imprimir variables de entorno (`envvar.sh`)

```
1 #!/bin/bash  
2  
3 echo $#  
4 echo $0  
5 echo $1
```

¿Qué sucede en cada caso?

4. Escribir un programa que inicialice dos enteros y devuelva el valor de la suma mediante el `return` de la función `main`. Utilizar `echo` para mostrar dicho valor almacenado en la variable de entorno `?` (`echo $?`).
5. ¿Qué hacen los siguiente comando de la shell?
  - a) `ls $HOME`
  - b) `ls /home/$USER`
  - c) `cd`
  - d) `cd $OLDPWD`
6. Utilizar el comando `man` para ver las páginas de manuales de las funciones de la biblioteca matemática (archivo de cabecera `math.h`) para las funciones `sqrt()` y `pow()`.
7. Ir al directorio `home` del usuario y crear un directorio (`mkdir`) de nombre `tmp` y moverse (`cd`) a este directorio. Luego, generar una estructura de directorios y archivos (usar el comando `touch`) que forme el siguiente árbol:

```
tmp  
|-- dir1  
|-- dir2  
|   |-- dir2a  
|   |   |-- file1
```

```
|  |-- dir2b
|      |-- file2
|-- dir3
```

A continuación:

- a) Ir al directorio `dir2b` y mover el archivo `file2` al directorio `dir2a` (utilizar path relativo)
  - b) Estando en el directorio `dir2b` listar (`ls`) el contenido del directorio `tmp`
  - c) Ir al directorio `dir1` y crear el archivo `file3a` bajo el directorio `dir3` utilizando path relativo
  - d) Estando en el directorio `dir1`, crear el archivo `file3b` bajo el directorio `dir3` utilizando path absoluto
  - e) Ir al directorio `dir2a` y cambiar el nombre del archivo `file1` a `file` (comando `mv`)
  - f) Ir al directorio `tmp` y borrar todos los archivos (`rm`) y directorios (`rmdir`) utilizando paths relativos
8. Escribir un programa que calcule la raíz cuadrada de un número entero, que tenga la siguiente interacción con el usuario:
- a) Solicitar al usuario un número entero (cero para salir)
  - b) Mostrar en la salida estándar (`stdout`) la raíz cuadrada del entero ingresado
  - c) Mostrar en la salida de error estándar (`stderr`) si el entero es negativo

Ejecute el programa:

- a) Redireccionando a un archivo la salida estándar
- b) Redireccionando a un archivo la salida de error estándar
- c) Redireccionando la entrada desde un archivo
- d) Redireccionando la salida de error estándar al archivo `/dev/null`

(Utilizar la función `fprintf()` de biblioteca de entrada/salida estándar (`stdio`). Ver la página de manual de dicha función.)

9. Crear un script de bash “*Hola mundo*” y guardar con nombre `hola.sh`

Listing 2: Script de shell *Hola mundo* (`hola.sh`)

```
1 #!/bin/bash
2 echo "Hola Informática-II"
```

Luego:

- a) Ejecutar con: `> ./hola.sh`. ¿Qué pasa?
- b) Observar los permisos del archivo `hola.sh` (`ls -l`)
- c) Agregar permisos de ejecución del usuario (`chmod u+x hola.sh`)
- d) Ejecutar nuevamente
- e) Desplegar el archivo en la terminal (`cat hola.sh`)
- f) Quitarle permiso de lectura del usuario (`chmod u-r hola.sh`)
- g) Desplegar nuevamente el archivo con el comando `cat`. ¿Qué pasa?
- h) Ejecutar el script
- i) Agregar permisos de lectura y de ejecución (`chmod u+wr hola.sh`)

10. Ir al directorio del usuario y crear un directorio de prueba

```
> cd [ENTER]
> mkdir prueba
```

Luego:

a) Crear un directorio

```
> mkdir dir
```

b) Crear un archivo vacío (0 bytes) dentro del directorio creado

```
> touch dir/vacio
```

c) Mostrar los permisos del archivo creado

```
> ls -l dir/vacio
```

d) Mostrar los permisos del propio directorio (archivo)

```
> ls -ld dir
```

Anotar los permisos en forma numérica (p.e., 735)

e) Quitar permiso de lectura al directorio

```
> chmod u-r dir
```

f) Mostrar los archivos del directorio

```
> ls -l dir
```

¿Qué pasa?

g) Moverse al directorio (`mv dir`) y listar los archivos (`ls`)

h) Restablecer los permisos originales del directorio creado utilizando notación numérica

i) Quitar permiso de escritura al directorio (`chmod u-w dir`) y crear otro archivo vacío

j) Borrar los archivos (`rm`) y directorios (`rmdir`) creados haciendo los cambios de permisos necesarios

11. Abrir una terminal, ejecutar un programa y matarlo con el comando `kill`, por ejemplo:

a) ejecutar `gedit` en segundo plano

```
> gedit &
```

b) listar los procesos y filtrar la salida con `grep`

```
> ps aux | grep gedit
```

c) anotar el número de proceso del programa `gedit`

d) matar el proceso

```
> kill -KILL <PID>
```