



Introducción a UNIX: Trabajo Práctico 4

Daniel Millán & Nicolás Muzi



Facultad de Ciencias Aplicadas a la Industria, UNCuyo
San Rafael 5600, Argentina
Marzo – Abril de 2022

Ejercicio 1.

1. Use *vi* para crear el **script** “*simple*” de **shell** dado en las notas. Ejecutarlo, y ver cómo el contenido de la secuencia de comandos se relaciona con la salida.
2. Cree el **script** “*variables.sh*” con las órdenes dadas a continuación, ejecútelo como un archivo común sin permisos de ejecución dando argumentos de entrada **uno**, **dos**, **tres**, **cuatro**.

```
#!/bin/sh
echo "$#:" $#
echo '$#:' $#
echo '$-:' $-
echo '$?:' $?
echo '$$:' $$
echo '$!:' $!
echo '$3:' $3
echo '$0:' $0
echo '$*:' $*
echo '$@:' $@
```

3. ¿Qué sucede si quitamos la extensión del archivo *variables.sh*?
4. Ahora vamos a crear un **script** *parameter.sh* para ver como es posible sustituir parámetros en Unix. Cree un archivo que contenga:

```
#!/bin/sh
param0=$0
test -n "$1" && param1=$1
test -n "$2" && param2=$2
test -n "$3" && param3=$3
echo 0: $param0
echo "1: ${param1-1}: \c" ; echo $param1
echo "2: ${param2-2}: \c" ; echo $param2
echo "3: ${param3+3}: \c" ; echo $param3
```

En el `script` primero realizamos un `test` (`man test`) para ver si existe la variable, en caso de existir le asignamos su valor a un parámetro. Luego se reportan los valores, y observamos cuando y como se realiza la sustitución.

En una primera ejecución pruebe

```
$ ./parameter.sh
```

En una segunda corrida ejecute

```
$ ./parameter.sh uno dos tres cuatro
```

5. ¿Qué sucede si utilizamos `bash` en lugar de `sh` en el `script` `parameter.sh`? ¿Cómo debemos modificar `parameter.sh` para que la salida sea exactamente la misma en `bash`?

Ejercicio 2. Cree el siguiente `script` “*batchwork*”

```
#!/bin/bash
echo "- - nohup: run commands in the background - -"
#avoid terminating the programs when logging off from a remote SSH session
echo "nohup $1 > foo.out 2> foo.err < /dev/null &"
nohup $1 > foo.out 2> foo.err < /dev/null &
```

1. ¿Puede explicar para que sirve este `script`?
2. ¿Cómo funciona? ¿Qué sucede con la salida estándar y el error estándar?
3. Verifique su utilidad conectándose al servidor de la FCAI (179.0.136.155 o 192.168.200.2, según corresponda), ejecute `sleep 600` mediante este `script`, salga y vuelva a conectarse.

Ejercicio 3. En relación al `script` “*batchwork*” del ejercicio anterior.

1. Inspeccione su árbol de directorios.
2. ¿Qué contiene su `.bashrc`? Identifique las variables `PATH` y `LD_LIBRARY_PATH`.
3. ¿Dónde debe colocar `script` “*batchwork*” dentro de su `home` local tal que pueda ser ejecutado desde cualquier sitio?

Ejercicio 4. Se deben generar `alias` tal que puedan ser ejecutados desde cualquier sitio dentro de su sesión.

1. ¿En que archivo se deben ubicar éstos si utiliza `bash`?
2. ¿Cuál es la orden que debe ser ejecutada una vez que se han agregado los `alias` deseados, tal que éstos sean exportados a otras `shells`?

Ejercicio 5. A continuación se brinda una serie de `alias`, que poseen pequeños errores, corríjalos, compruebe que funcionan y discuta su relevancia.

1. `alias ll=`ls -l``
2. `alias la = ls -la`
3. `alias fcai=ssh alumnoXY@179.0.136.155`

Ejercicio 6. Pruebe loguearse al servidor de la FCAI sin introducir su `password` y mediante la ejecución del `alias` creado en el ejercicio anterior, es decir:
`$ fcai`

Entrega

Se entregarán los ejercicios 1 y 2.