



Introducción a UNIX: Trabajo Práctico 5

Daniel Millán, Nicolás Muzi & Eduardo Rodriguez

Facultad de Ciencias Aplicadas a la Industria, UNCuyo

San Rafael 5600, Argentina

Mayo – Junio de 2024



Ejercicio 1. Cree un **script** “*aproxima*” el cual genera un número aleatorio entre 1 y 100 (*Hint*: vea la función **shuf**), e indica pistas al usuario hasta que adivina el número. El **script** debe dar pistas del estilo “Lo siento tu intento es demasiado alto”, o “Intenta nuevamente con un numero menor” (no utilice acentos).

Ejercicio 2.

1. Escriba un **shell script** que cambie la extensión de todos los archivos **.txt** por la extensión **.text**. Podría serle de ayuda la orden **basename**.
2. Modifique el **script** anterior (cree uno nuevo) para que en caso de ser indicado cambie de una extensión dada a otra cualquiera también definida por el usuario.

Ejercicio 3.

1. Cree un **shell script** “*pidof*” el cual dado un nombre como parámetro devuelve el PID(s) del proceso con ese nombre.
2. Agregue una función **usage()** al **shell script** “*pidof*” creado anteriormente la cual imprime información sobre el modo de uso. La función debe ser llamada en caso de pasar un número erróneo de parámetros.

Ejercicio 4.

1. ¿Qué sucede si la función **findfile** dada en las notas es agregada en su **.bashrc**?
2. Cree un **script** “*swap*” tal que dados dos archivos comunes de entrada intercambie los contenidos entre ellos. ¿Es conveniente utilizar una función **swapping()**?
3. ¿Dónde se supone que debe colocar las siguientes funciones y qué hacen?

```
ls() { /bin/ls -sbF "$@"; }  
ll() { ls -al "$@"; }
```

Ejercicio 5.

1. Cree el siguiente **script** dado en clase

```
#!/bin/sh
for f in *.txt
do
    echo ordenando archivo $f
    cat $f | sort > $f.sorted
    echo archivo ordenado ha sido redireccionado a $f.sorted
done
```

Modifique el **script** (puede hacerlo en **bash**) tal que guarde el archivo con extensión **.txts** en lugar de **.txt.sorted**.

2. Ahora cree el **shell script** “*waitinput*” cuyo contenido se indica a continuación (dado en las notas):

```
#!/bin/sh
while [ ! -s input.txt ] do
    echo waiting ...
    sleep 5
done
echo input.txt is ready
```

Pruebe si funciona, en caso de que no funcione ¿Qué error tiene, puede corregirlo?

Ejercicio 6.

1. Realice un **shell script** que permita crear un conjunto N de directorios, numerados que comiencen con el mismo “nombre”. Debe chequear que dicho número N no exceda 1000, y que el largo de la cadena de caracteres del “nombre” no sea mayor a 8.
2. Crear una función **usage()** que brinde información sobre como se debe emplear este **shell script** en caso de error.
3. Cree un script que “renombre” los directorios creados en el apartado anterior.

Ejercicio 7.

1. ¿Puede explicar cada línea del siguiente **shell script**? ¿Qué hace?

```
#!/bin/sh
while [ $# -gt 0 ]
do
    echo $1
    shift
done
```

2. ¿Qué diferencia existe con el siguiente **script**?

```
#!/bin/sh
until [ $# -le 0 ]
do
    echo $1
    shift
done
```

Ejercicio 8. Dado un conjunto no definido pero finito de archivos comunes del tipo `mod_rotu_X.vtk` se desea contar con un programa que les permita cambiar el nombre a `modelo_rotura_YYYY.itk`. ¿Cómo haría esto?. Realice el **shell script** que permita resolver este problema. (Nota: X representa un número entre 0 y 9999, e YYYY un número del tipo 0000 a 9999).

Ejercicio 9. Cree un script que le permita modificar el archivo de configuración del programa AneuSurfIsolation para ejecutar dicho programa sobre una lista de casos seleccionados. Guarde cada copia modificada del archivo *.cfg en un directorio correspondiente a cada caso CXXXX*.

Entrega

Se entregarán los ejercicios 3, 5, 6.1, 6.2, 7 y 9.