



Introducción a UNIX: Examen Final



Profesor: Daniel Millán

FCEN – Mendoza, 5 de Mayo de 2016

Ejercicio 1 [20pts=4x5pts]

Analice mediante órdenes simples los datos existentes en el directorio “data_unix_examen”. Ahora responda:

- a) ¿Qué tamaño en bytes ocupa el directorio? ¿En MB?
- b) ¿Cuántos archivos comunes hay?
- c) ¿Cuántos archivos comunes existen tal que poseen un tamaño mayor a 102 Kilobytes?
- d) Genere un archivo “dirssize” que contenga en dos columnas (I) el tamaño en bytes de cada subdirectorio, y (II) el nombre del correspondiente subdirectorio.

Ejercicio 2 [15pts]

Utilizando como base el script “data.awk” cree un script “dirscheck.awk” tal que imprima todos aquellos directorios que almacenan mas Kilobytes que la media.

Ayuda 1: Utilizar el archivo “dirssize” del Ejercicio 1.d).

Ayuda 2: $1024b = 1K$

Ayuda 3: El script debe imprimir en pantalla información similar a la presentada a continuación:

```
linea 102  size= 5923K  dir:30=./da011
```

Ejercicio 3 [15pts=3x5pts]

Se debe enviar el directorio “data_unix_examen” a un server de forma remota, en el cual usted tiene una cuenta de usuario.

```
user:  rucacura
server: rucamaiten.gob.ar
```

- a) Genere el “archivo tar” del directorio de datos y comprímalo utilizando 3 formas distintas. ¿Cuál es su conclusión?
- b) ¿Es posible generar un archivo .tar.algo que ocupe menos de 1MB? Indique de que manera lo haría si su respuesta es afirmativa.
- c) Supongamos que es la primera vez que accede al server. ¿Qué procedimiento “de tres pasos” debe seguir para colocar los datos en su directorio ~/datos/trabajo en ese server?

Ejercicio 4 [30pts=6x5pts]

#Cree un directorio “datainfo”.

- a) Copie dentro de “datainfo” todos los archivos comunes “topout_*.*” que sean mayores de 50K.

#Ahora trabaje dentro del directorio datainfo.

- b) Realice un script en bash que recorra todos los archivos en datainfo y que genere un nuevo archivo (reemplace el archivo original) pero conteniendo la informacion contando desde la linea que posee la cabecera:

| PID | USER | PR | NI | VIRT | RES | SHR | S | %CPU | %MEM | TIME+ | COMMAND |
|-----|------|----|----|------|-----|-----|---|------|------|-------|---------|
|-----|------|----|----|------|-----|-----|---|------|------|-------|---------|

- c) Genere un archivo “toptodo.txt” que contenga toda la información almacenada en los archivos generados en el punto anterior, es decir en un solo archivo.
- d) Elimine todas las líneas que contengan la palabra PID del archivo “toptodo.txt”.
- e) Ordene el contenido del archivo “toptodo.txt” según el numero PID. Indique el mínimo y el máximo valor de PID.
- f) ¿Cuántas veces aparece la palabra `sleep` en el archivo “toptodo.txt”?

[5pts] **Bonus track.** Colecte en el archivo:

- g) “topsortPID.txt” el número de PIDs no idénticos que se encuentran contenidos en “topsort.txt”;
- h) “topsortCOMMAND.txt” los nombres de órdenes que son diferentes. –¿Cuántas veces aparece la palabra `sleep`? Reconsidere su respuesta del Ejercicio 4f)–

Ejercicio 5 [20pts=2x10pts]

Se desea renombrar los archivos “topout_*.*” contenidos en el directorio “datainfo” tal que posean la misma extension ‘.txt’. Además, no se desea perder la información que en ellos se halla contenida. Nota: Tener en cuenta que hay muchos archivos que poseen el mismo nombre pero diferente extensión.

- a) Identifique todos los nombres de archivo que son diferentes. Coléctelos de forma ordenada en un archivo “topnombreunico.txt”.
- b) Realice un script en bash “colectar.bash”, el cual crea para los archivos que poseen el mismo nombre y que se encuentran almacenados en “topnombreunico.txt” un único archivo con extension “.txt”. En este archivo se debe almacenar de forma ordenada según el numero PID toda la información contenida en los archivos que poseen un nombre idéntico. Las líneas de cabecera no deben ser incluidas (o deben ser eliminadas).

[5pts] **Bonus track.**

- c) ¿Cómo modificaría el script “colectar.bash” tal que la cabecera sea incluida una única vez al principio de cada archivo?