



Introducción a UNIX: Trabajo Práctico 2

Daniel Millán
Evelin Giaroli & Nora Moyano



Facultad de Ciencias Aplicadas a la Industria, UNCuyo
San Rafael 5600, Argentina
Junio de 2017

Para poder realizar esta práctica debe:

1. Crear en su directorio personal `/home/alumnoXY` los directorios `data`, `local`, `bin`, `lib`.
2. Copiar y descomprimir en su `/home/alumnosXY/data` los datos contenidos en el archivo `/scratch/IntroUnix2017_tp2_data.tar.xz`.

Ejercicio 1.

1. Describa tres maneras diferentes de ajustar los permisos de un archivo o directorio a `r-r-r-`. Cree un archivo y vea si esto funciona.
2. Forme un equipo con un compañero. Copie `/bin/sh` a su directorio. Teclee `chmod +s sh`. Compruebe los permisos de `sh` listando el directorio. Ahora pida a su compañero que cambie a su directorio de inicio `home` y que ejecute el programa `./sh`. Pídale que ejecute el comando `id`. ¿Qué ha pasado? Su compañero puede escribir `exit` para regresar a su `shell`.
3. ¿Qué pasaría si el administrador del sistema crea un archivo `sh` de esta manera? Elimine `sh` desde su directorio principal (o al menos cambie los permisos `chmod -s sh`).
4. Modifique los permisos de su `home` para hacerlo completamente privado. Chequee que nadie puede acceder (compañero). Ahora restablezca los permisos originales de su `home`.
5. Teclee `umask 000` y entonces cree un archivo llamado `mundo.txt` conteniendo las palabras "hola mundo". Mire los permisos del archivo. ¿Qué ha sucedido? Ahora teclee `umask 022` y cree un archivo llamado `mundo2.txt`. ¿Cuándo esta característica puede ser útil?
6. Cree un archivo llamado `hola.txt` en su directorio personal utilizando el comando `cat -u>hola.txt`. Pídale a su compañero que se cambie a su directorio de inicio y ejecute `tail -f hola.txt`. Ahora escriba varias líneas en `hola.txt`. ¿Qué aparece en la pantalla de su compañero?

Ejercicio 2.

1. Utilizar `find` para mostrar los nombres de todos los archivos en el árbol de directorios `/home`. ¿Puede hacer esto sin mostrar errores provenientes de aquellos archivos que no se pueden leer?

2. Utilizar `find` para mostrar los nombres de todos los archivos del sistema que son más grandes que 1 MB.
3. Utilizar `find` y `file` para mostrar todos los archivos en el árbol de directorios `/home`, así como una pista sobre qué tipo de archivos son. Haga esto de dos maneras diferentes.
4. Utilizar `grep` para separar la línea en `/etc/passwd` que contiene sus datos de acceso.
5. Utilizar `find`, `grep` y `sort` para mostrar una lista ordenada de todos los archivos en el árbol de directorios `/home` que contienen la palabra `hola` en algún lugar dentro de ellos.
6. Utilizar `locate` para encontrar todos los nombres de archivo que contienen la palabra `emacs`. ¿Puede combinar esto con `grep` para evitar mostrar todos los nombres de los archivos que contienen la palabra `lib`?
7. Cree un archivo conteniendo algunas líneas que usted piensa coincidirán con la expresión regular: `(^[0-9]{1,5}[a-zA-Z]+)$|none` y algunas líneas que usted piensa que no se corresponderán. Use `egrep` para ver si su intuición es correcta.

Ejercicio 3.

1. Realizar un registro (*archive*) del contenido de su directorio personal (incluidos los subdirectorios) utilizando `tar` y `cpio`. Comprima el archivo `tar` con `compress` (o `bzip2`), y el archivo de `cpio` con `gzip`. Ahora extraiga sus contenidos en dos subdirectorios independientes, `dirtar` y `dircpio`.
2. “Archive” el contenido de su directorio personal utilizando `tar`. Comprima el archivo `tar` con `gzip`. Ahora descomprima y “desarchive” el archivo `.tar.gz` usando `cat`, `tar` y `gzip`.
3. Genere un archivo `tgz` de su directorio personal con todos los subdirectorios y archivos. Descomprímalo. Elimine todos los archivos y directorios generados.

Ejercicio 4.

1. En los sistemas Linux, el archivo `/dev/urandom` es una cadena de caracteres generada de forma aleatoria. ¿Puede utilizar este archivo con `od` para imprimir un número decimal aleatorio?
2. Teclee `mount` (sin parámetros) y trate de interpretar la salida. No olvide GETA.

Ejercicio 5. Utilizar `find` para generar una lista de todos los directorios en el sistema, redirigiendo la salida de modo que la lista de directorios se almacene en un archivo llamado `directorios.txt` y la lista de mensajes de error en un archivo llamado `errores.txt`.

Ejercicio 6.

1. Pruebe la orden `sleep 10`. ¿Qué hace esta orden? Ejecutar la misma orden pero directamente en el segundo plano (o fondo) usando `&`.
2. Ejecutar `sleep 30` en el primer plano, suspender con `Ctrl-z` y luego enviarlo al segundo plano con `bg`. Teclee `jobs`. Teclee `ps`. Lleve el trabajo de nuevo al primer plano con `fg`.

3. Ejecutar `sleep 30` en el segundo plano usando `&` y, a continuación, utilizar `kill` para terminar el proceso por su número de trabajo (`job`). Repetir, excepto que esta vez elimine el proceso especificando su PID.
4. Inicie una serie de procesos `sleep 300` en segundo plano. Ponga fin a todos al mismo tiempo usando la orden `pkill`.

Ejercicio 7.

1. Utilice `ps`, `w` y `top` para mostrar todos los procesos que se están ejecutando.
2. Utilice `ps -aeH` para mostrar la jerarquía de procesos. Busque el proceso `init`. Vea si usted puede identificar importantes “demonios del sistema”*. ¿Puede identificar su `shell` y sus subprocesos?
3. Combine `ps -fae` con `grep` para mostrar todos los procesos que está ejecutando, con la excepción de los comandos `ps -fae` y `grep`.
4. Inicie un proceso `sleep 600` que se ejecute en el segundo plano. Cierre la sesión en el servidor, y vuelva a iniciarla. Liste todos los procesos que se están ejecutando. ¿Qué sucedió con el proceso `sleep`? Repita los pasos anteriores excepto que esta vez mediante la orden `nohup sleep 600`. ¿Qué diferencias observa?
5. Se pueden enviar varios trabajos desde la misma línea de órdenes utilizando los operadores `;`, `&&` y `||`. Trate de combinar las órdenes `cat inexistente` y `echo hola` utilizando cada uno de estos operadores. Invierta el orden de las órdenes y vuelva a intentarlo, ahora otra vez. ¿Cuáles son las reglas sobre cuándo se ejecutan las órdenes?

Ejercicio 8.

1. ¿Qué hace la orden `xargs`? ¿Puede combinarla con `find` y `grep` para encontrar otra manera de buscar todos los archivos en el árbol de directorios `/home` que contienen la palabra `hola`?
2. ¿Qué hace la orden `cut`? ¿Puede utilizarla junto con `w` para producir una lista de nombres de usuario y tiempos de CPU correspondientes a cada proceso activo? ¿Puede utilizar `sort` y `head` o `tail` para encontrar el usuario cuyo proceso está utilizando el mayor porcentaje de la CPU (todo en la misma línea de órdenes)?

Ejercicio 9.

1. Liste todos sus procesos, usando `sed` para sustituir su nombre de usuario por “yo”.
2. Utilice `who`, `awk`, `sort`, y `uniq` para imprimir una lista ordenada de los nombres de sesión de los usuarios activos.

*Un *daemon* es un tipo especial de proceso informático no interactivo, es decir, que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario. Este tipo de programas continua en el sistema, es decir, que puede ser ejecutado en forma persistente o reiniciado si se intenta matar el proceso dependiendo de configuración del demonio y políticas del sistema.

3. Utilizar `awk` en `/etc/passwd` para producir una lista de usuarios y las shells donde han iniciado una sesión de usuarios (donde están “logueados”).
4. Escribir un `script awk` que imprime todas las líneas de un archivo a excepción de las dos primeras.

Ejercicio 10.

1. Modifique el `script data.awk` tal que imprima sólo los equipos que tienen un FGP mayor que la media de todo la liga.
2. Cree un `script nbaEFG.awk` que calcule el *Effective Field Goals Percentage (eFG %)*[†] de la temporada de la NBA 2015. Debe imprimir sólo un dígito decimal.

Ejercicio 11. Todos los puntos de este ejercicio se refieren al análisis de la información contenida en el archivo `world-power-plants-list.csv`, el cual se debe realizar mediante órdenes y programas de Unix. Para más detalles puede consultar la información dada en el documento de Anexos de esta unidad temática.

Nota: Este ejercicio tiene nivel de **EXAMEN**.

1. Inspeccione e interprete el encabezado de la lista. ¿Cuántas plantas se listan?
2. Redireccione las últimas 1000 líneas a un nuevo archivo `wpp-last1000.csv`.
3. Genere un archivo que contenga dos columnas, en la primer columna debe especificar el tipo de planta mientras que en la segunda debe dar la información de la capacidad de diseño expresada en MWt. Emplee espacios para separar estas columnas. Utilice GETA con *Using AWK on CSV Files*[‡] como argumento. No utilizar `sed`.
4. ¿Cuál es la razón de que se pida no utilizar `sed` en el apartado anterior? Realice el apartado anterior utilizando `sed`.
5. Extraiga todas las plantas de generación de energía que se encuentran en Argentina según este documento. ¿Alguna ausencia notable? ¿Por qué?
6. ¿Cuántos países diferentes hay en la lista?
7. Genere un nuevo archivo donde sólo se muestre la información de las plantas de energía nuclear.
8. Exprese de forma porcentual la cantidad de energía generada por cada tipo de planta.
9. ¿Cuál es el porcentaje respecto al total de energía producida que se genera en: a) EEUU, b) China, b) Brasil, d) Argentina y e) San Rafael?
10. ¿Cuáles son los países que generan (–¿consumen?–) el 90 % de energía?

[†]Measures field goal percentage adjusting for the fact that a 3-point field goal is worth one more point than a 2-point field goal = $(FGM + 0.5 * 3PM) / FGA$.

[‡]https://www.joeldare.com/wiki/using_awk_on_csv_files