



Introducción a UNIX: Trabajo Práctico 2

Daniel Millán & Nicolás Muzi

Facultad de Ciencias Aplicadas a la Industria, UNCuyo

San Rafael 5600, Argentina

Mayo – Junio de 2019



Para poder realizar esta práctica debe:

1. Crear en su directorio personal los directorios `data`, `local`, `bin`, `lib`.
2. Copiar y descomprimir en la carpeta `data` los datos contenidos en el archivo `introunix.txz`.
3. Mueva las carpetas contenidas en el directorio `2019_Electiva_FCAI/Clases/introunix/` a `data/`.
4. Elimine los directorios vacíos contenidos en `data`.

Ejercicio 1.

1. Cree el grupo “moccai”.
2. Cree un usuario “pepe”^{*} cuya clave personal sea “introunix”. Usted puede asignarle los datos del usuario que considere pertinentes (nombre, dirección, etc.). Logueado como “pepe” copie el directorio `data` de su usuario en `/home/pepe/data`. Asígnele a `data` permisos 700.
3. Asigne como grupo primario de su usuario personal y del usuario “pepe” el grupo “moccai”. Verifique que esas acciones hayan sido efectuadas. *Ayuda:* puede usar `ls -l /home` o la orden `groups`.
4. 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.
5. Modifique los permisos de su `home` (`/home/alumno`) para hacerlo completamente privado. Chequee que nadie puede acceder (u: pepe). Ahora restablezca los permisos originales de su `home`.

^{*}Se recomienda que utilice `adduser` en vez de `useradd`, ya que `adduser` crea usuarios de acuerdo a la configuración del archivo `/etc/adduser.conf`. De esa forma salva el problema de configuración de *autocompletion* y de todo el perfil del usuario.

Ejercicio 2.

1. Realice el Ejercicio 1.
2. 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? Si la respuesta es positiva, muestre cómo.
3. Utilizar `find` para mostrar los nombres de todos los archivos contenidos en `"/usr"` que son más grandes que 1 MB (`man find`).
4. Utilizar `find`, `grep` y `sort` para mostrar una lista ordenada de todos los archivos en el árbol de directorios `/home/alumno` que contienen la palabra `hola` en algún lugar dentro de ellos.
5. 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`?

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 `tar` y `gzip`.
3. Genere un archivo `txz` y un `tgz` de su directorio personal con todos los subdirectorios y archivos. ¿Existe alguna diferencia entre ambos?

Ejercicio 4. 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 5.

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 6.

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 sub-procesos?
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 terminal y vuelva a abrirla. 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 7.

1. Liste todos sus procesos, usando `sed` para sustituir su nombre de usuario por “yo”.
2. Busque la palabra `pepe` en todos los archivos `.txt` y replácela por `Pepe Argentó`, utilizando `sed`.
3. Utilice `who`, `awk`, `sort`, y `uniq` para imprimir una lista ordenada de los nombres de sesión de los usuarios activos.

Entrega

Se pide entregar los Ejercicios 5, 6 y 7.

[†]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.