

INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO

LABORATORIO: Sistemas Operativos

### Práctica 3

Comandos básicos de Linux y WinNT

LosDos

*Integrantes:*

Amanda Velasco Gallardo - 154415  
Carlos Octavio Ordaz Bernal - 158525

*Fecha(s) de elaboración de la práctica:*

8 de febrero de 2019

## Introducción

De manera general, existen dos formas en las que un usuario interactúa con un sistema operativo al momento de querer comandar instrucciones: utilizando únicamente el teclado, o mediante una combinación de ratón y teclado. El intérprete de comandos, que es el encargado de traducir las órdenes de los usuarios a un conjunto de instrucciones que el núcleo del sistema pueda entender, puede ser a través de una interfaz gráfica, o con un programa conocido como línea de comandos. El primero combina el uso de ratón y teclado, mientras que el segundo sólo hace uso de las líneas que el usuario escribe utilizando solamente el teclado.

La línea de comandos es característica de los sistemas operativos de la familia **UNIX**, mientras que los sistemas operativos de la familia **Windows NT** utilizan de una interfaz gráfica como intérprete de comandos. A continuación detallaremos cada uno de éstos así como su intérprete de comandos.

Por un lado, los sistemas operativos de la familia **UNIX** son sistemas multiusuarios y multiprogramación desarrollados en 1969 por un grupo de empleados de los laboratorios Bell de AT&T. La variante **Linux** cuenta con lo que se conoce como consola (*Linux console*), la cual provee una forma al kernel, y a otros procesos, para desplegar texto como salida al usuario y recibir texto como entrada a manera de instrucciones.

Dentro de **Linux**, varios dispositivos pueden ser utilizados como consola del sistema, por ejemplo: una terminal virtual, los puertos seriales, los puertos USB, las salidas de VGA, entre otros. Además, muchos de los sistemas operativos de hoy en día han reemplazado el arranque del sistema computacional utilizando la entrada y salida del kernel por un ambiente gráfico como el logo o una barra de progreso.

Por otro lado, los sistemas operativos que forman parte del grupo **Windows New Technologies** o **Windows NT** son desarrollados por Microsoft, y su primera versión fue publicada en 1993. Las versiones que forman parte de esta familia son: **Windows 10**, **Windows 8**, **Windows 7**, **Windows Vista**, **Windows XP**, **Windows Server 2003**, **Windows 2000** y **Windows NT**. Todas éstas son versiones de sistemas operativos multiprogramación diseñados para trabajar con uno o múltiples procesadores. Ejecutan los procesos de manera asíncrona y su arquitectura es modular. Además, son multiusuarios con la capacidad de funcionar en modo usuario y modo núcleo.

Los sistemas operativos descritos en el párrafo anterior, cuentan con un intérprete de comandos gráfico conocido como *Explorador de archivos*, el cual

es una interfaz gráfica que le permite al usuario dar instrucciones al sistema por medio de una combinación de ratón y teclado. Además cuenta con una aplicación que es el símbolo de sistema, `command prompt`, o de manera corta `cmd`, el cual es un intérprete de comandos similar a la consola de Linux pero que no posee la función de cargar la configuración al cargar el sistema operativo.

## Desarrollo

1. ¿Cuál es el tamaño de la memoria central tanto en Windows como en Linux? Expresa la memoria tanto en bytes como en megabytes.

- En Windows: 17,179,869,184 bytes, 16,384 megabytes.
- En Linux: 2,110,521,344 bytes, 2,012 megabytes.

2. ¿Para qué sirve el comando `man`? Aplique el comando `man man`. ¿Qué nos indica el desplegado? Despliegue las 10 primeras líneas de `man man`. ¿Para qué sirve el comando `ps`? ¿Al ejecutar el comando `ps` qué indican los valores de las columnas TTY y CMD? Despliegue las 10 primeras líneas de `man ps`. Existe un caso especial con `man ascii`. ¿En este caso qué despliega la ayuda?

El comando `man` despliega la página del manual en línea del comando especificado. Al aplicar `man man`, el resultado es el manual sobre el comando `man`. El desplegado de `man man` se muestra en la figura 1.

```
MAN(1)                                Manual pager utils                                MAN(1)

NAME
    man - an interface to the on-line reference manuals

SYNOPSIS
    man [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
    locale] [-m system[,...]] [-M path] [-S list] [-e extension] [-t|-I]
    [--regex|--wildcard] [--names-only] [-a] [-u] [--no-subpages] [-P
    pager] [-r prompt] [-7] [-E encoding] [--no-hyphenation] [--no-justifi-
    cation] [-p string] [-t] [-T[device]] [-H[browser]] [-X[dpi]] [-Z]
    [[section] page ...] ...
```

Fig. 1: Resultado del comando `man man`.

El comando `ps` reporta un *snapshot* de los procesos actuales. TTY indica la terminal en la que está corriendo el proceso mientras que CMD indica el nombre del comando del proceso. El desplegado de `man ps` se muestra en la figura 2

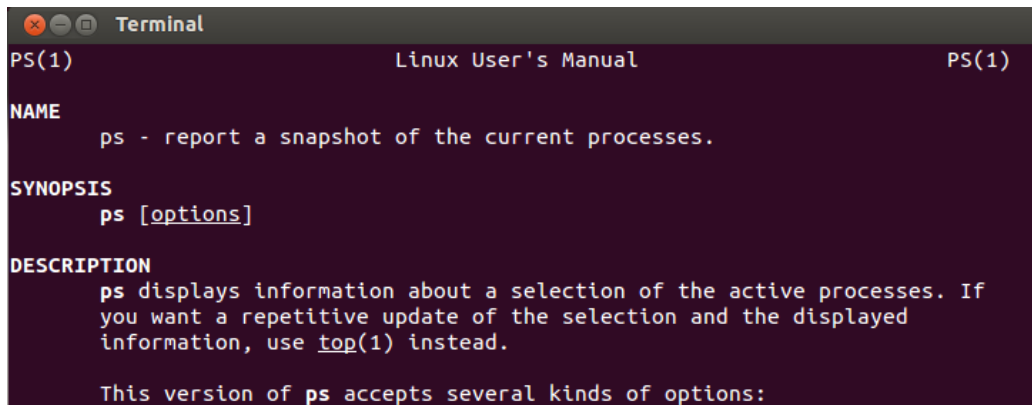


Fig. 2: Resultado del comando `man ps`.

Por último, el comando `man ascii` arroja como resultado la tabla del código ASCII así como detalles del mismo.

**3. El comando `ps` en su formato `ps -l` lista los procesos activos bajo la sesión *sisops*. ¿Qué diferencia hay con el formato `ps -e`? ¿Cuál es el PID del proceso *ps*? ¿Cuál es el proceso padre de *ps* y por qué? ¿Qué función desarrolla el proceso padre de *ps*?**

El comando `ps -e` despliega la lista de todos los procesos de la máquina virtual, a diferencia de `ps -l`, el cual despliega la lista de los procesos del usuario actual y en la terminal actual. El PID del proceso *ps* es 1000. Su padre es el proceso *tcsh* y esto es visible ya que el comando `ps` muestra el proceso actual (PID) y su padre (PPID). A su vez, *tcsh* es el shell de UNIX basado en y compatible con el lenguaje C.

**4. Aplique el comando `uname -a`. ¿Qué significa cada uno de los valores desplegados?**

Los valores que el comando devuelve se encuentran en el orden siguiente:

- Nombre del kernel
- Hostname

- Versión del kernel
- Información sobre la última compilación
- Arquitectura del sistema
- Nombre del sistema operativo

**5. Con ayuda del comando `id`, de su cuenta, y el contenido de la variable de ambiente `shell`, diga:**

- *¿Cuál es el identificador de la cuenta?* sisops
- *¿Cuál es el número de la cuenta?* 1000
- *¿Cuál es el directorio base (home directory) de la cuenta?* home/sisops
- *¿Cuál es el intérprete de comandos (shell) asociado a la cuenta?* tcsh
- *¿Cuál es el grupo al que pertenece la cuenta?* 1000

**6. ¿Para qué sirven los comandos `logout`, `exit` y `Ctrl+d`?**

De manera general, los 3 comandos sirven para terminar la sesión de un usuario en una terminal. El comando que se debe de utilizar depende del tipo de shell que se esté utilizando.

**7. Suponga que su directorio base se encuentra en *luis* y usted se encuentra en el directorio de trabajo *juan*. ¿Cómo comprobaría que se encuentra en el directorio *juan*? ¿Qué desplegaría dicho comando?**

Para comprobar el directorio actual se puede utilizar el comando `pwd`. Éste desplegaría la ruta absoluta hasta el directorio actual.

- *Escriba la trayectoria absoluta del archivo `prod`:* usr/users/Luis/dat/prod
- *Escriba la trayectoria relativa del archivo `prod` desde *juan*:* ../Luis/dat/prod
- *Escriba la trayectoria relativa del archivo `hosts` desde *juan*:* ../../etc/hosts

**8. Genere el siguiente ambiente mediante el comando `mkdir` de la manera más óptima.**

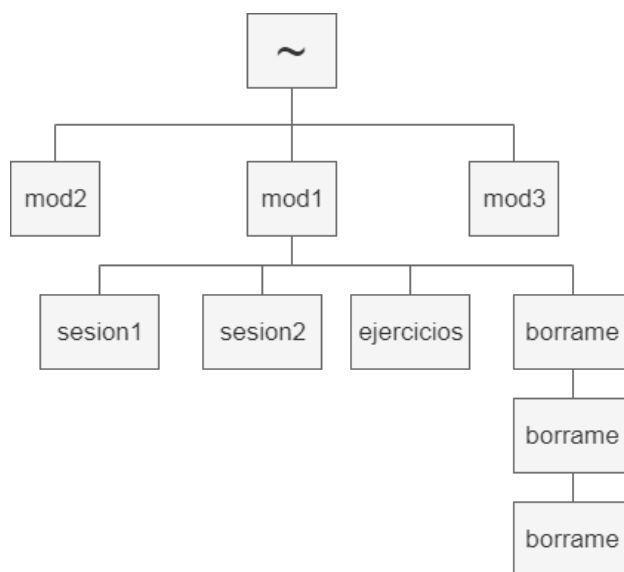


Fig. 3: Ambiente por crear.

```
ubuntu:~> mkdir -p mod1 mod2 mod3
ubuntu:~> mkdir -p mod1/borrame/borrame/borrame mod1/sesion1 mod1/sesion2 mod1/ejercicios
ubuntu:~>
```

Fig. 4: Creación del ambiente de la figura 3 en Linux.

9. Suponiendo que se encuentra en el subdirectorio *sesion1* del inciso 8, muestre dos alternativas diferentes (de un solo comando) de cambiarse al subdirectorio *mod3*.

- `cd ~ /mod3`
- `cd ../../mod3`

10. ¿Qué diferencias hay entre aplicar el comando `ls` en alguna de las siguientes maneras: `ls`, `ls -a` y `ls -l`?

`ls` solo muestra el contenido del directorio, mientras que `ls -l` muestra los contenidos con sus nombres en formato largo. `ls -a` muestra todos los contenidos del directorio sin exceptuar aquellos cuyos nombres inician con un punto.

**11. ¿Qué hace el comando `tree` en Windows? ¿Cuál comando de Linux (nativo) es similar al comando `tree` de Windows? Muestre el comando completo en Linux.**

`tree` despliega toda la estructura del directorio en un disco específico. Si no se le agregan argumentos muestra al directorio actual. En Linux, un comando similar es `ls -R`.

**12. Guarde en un archivo el árbol de directorios del inciso 8 con alguno de los formatos del comando `ls`, siempre a partir del *home directory*. Muestre el comando aplicado.**

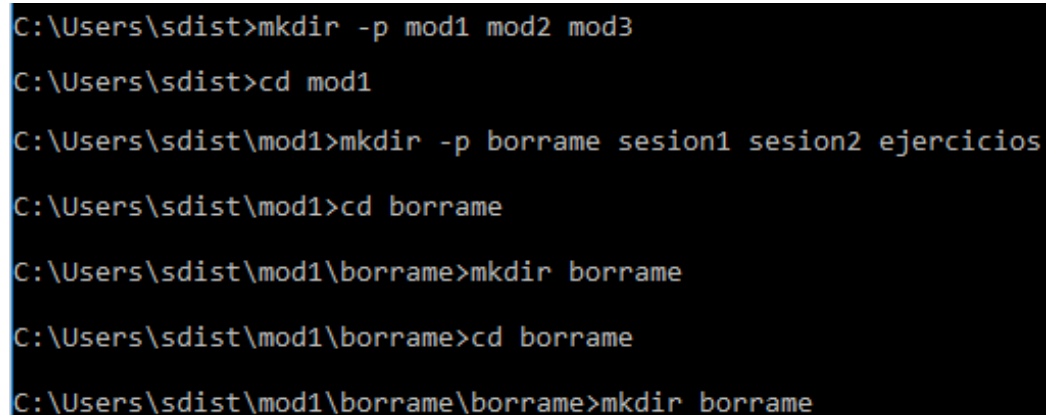
Suponiendo que el árbol resultante se desea guardar en el archivo *arbol8.txt*, el comando que se debe utilizar es `ls -R >arbol8.txt`.

**13. Muestre el o los comandos que tendría que aplicar, desde cualquier trayectoria en el árbol de directorios, para que al mandar ejecutar el comando `pwd`, éste despliegue el directorio base de su cuenta.**

Para desplegar la trayectoria del directorio base de la cuenta desde cualquier directorio, podemos utilizar los siguientes comandos:

- `cd ~/`, y en la siguiente línea `pwd`
- `cd`, y en la siguiente línea `pwd`

**14. Repita el inciso 8 en WinNT, desde una ventana de DOS.**



```
C:\Users\sdist>mkdir -p mod1 mod2 mod3
C:\Users\sdist>cd mod1
C:\Users\sdist\mod1>mkdir -p borrame sesion1 sesion2 ejercicios
C:\Users\sdist\mod1>cd borrame
C:\Users\sdist\mod1\borrame>mkdir borrame
C:\Users\sdist\mod1\borrame>cd borrame
C:\Users\sdist\mod1\borrame\borrame>mkdir borrame
```

Fig. 5: Creación del ambiente de la figura 3 en WinNT.

**15. Repita el inciso 9 en WinNT, desde una ventana de DOS.**

Los comandos que se pueden aplicar son:

- `cd ../../mod3`
- `cd C:/Users/sdist/mod3`

**16. ¿En cuál día de la semana ocurrió la fecha 18 de marzo de 1948?**

**¿Con cuál comando encontró tal día?**

Tal fecha cayó en jueves. Esto se puede ver utilizando el comando `cal 03 1948`. El despliegue correspondiente se muestra en la figura 6.

```
ubuntu:~> cal 03 1948
      March 1948
Su Mo Tu We Th Fr Sa
      1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

Fig. 6: Despliegue del comando `cal`.

**17. Explique para qué sirven los comandos `who` y `finger`.**

`who` muestra información sobre los usuarios que actualmente tienen sesión iniciada. `finger` permite revisar información de cualquier usuario de forma local o remota.

**18. Explique la diferencia de operación entre los comandos `rm` y `rmdir`.**

`rm` elimina los contenidos de un directorio mientras que `rmdir` elimina al directorio en sí (si está vacío).

**19. Explique tanto el comando `free -mt` como el contenido de su despliegue. ¿Alguno de estos valores ya lo había visto anteriormente, dentro de esta práctica o su aproximación?**

`free -mt` muestra el tamaño de la memoria central y de la memoria swap expresado en megabytes. Específicamente, muestra el total, la memoria libre,



la memoria en uso y la memoria en cache. El valor de la memoria central se había visto ya en el punto 1.

**20. Explique los siguientes comandos así como sus respectivos despliegues:**

- `wmic computersystem get manufacturer`: Nombre del fabricante de la computadora (Lenovo).
- `wmic cpu get NumberOfCores, NumberOfLogicalProcessors`: Número de cores de cada procesador así como el número de procesadores (4 cores, 8 procesadores).
- `wmic computersystem get totalphysicalmemory`: Memoria física expresada en bytes (17,083,961,344).
- `wmic memorychip get capacity`: Capacidad de cada memoria RAM en bytes (8,589,934,592).

En los últimos dos comandos, ¿coincide el resultado de la memoria? ¿Por qué? Si los valores son diferentes, ¿cuál aparece como un valor mayor? ¿De cuánto es la diferencia? Los resultados de estos dos comandos están expresados en bytes. Expréselos en gigabytes y muestre los cálculos correspondientes.

No, las memorias no coinciden ya que el despliegue de `wmic memorychip get capacity` muestra la memoria total instalada mientras que el despliegue de `wmic computersystem get totalphysicalmemory` muestra la memoria disponible descontando el espacio apartado previamente para el kernel y manejo del hardware. Es por esto que `wmic memorychip get capacity` aparece mayor y con una diferencia de 95,907,840 bytes.

Para convertir de bytes a gigabytes es necesario recordar que 1 gigabyte equivale a  $1024^3$  bytes.

- `wmic computersystem get totalphysicalmemory`:

$$17083961344B \times \frac{1GB}{1024^3B} = 15,91GB$$

- `wmic memorychip get capacity`:

$$17179869184B \times \frac{1GB}{1024^3B} = 16GB$$

## Conclusiones

La presente práctica nos permitió conocer la forma en la que trabajan los intérpretes de comando (línea de comandos e interfaz gráfica) de dos de los sistemas operativos más utilizados hoy en día. Es de gran importancia estar familiarizado con los comandos que permiten navegar y ejecutar instrucciones simples que son utilizadas de manera constante. El comando de mayor relevancia que encontramos fue `man` debido a que éste, dentro de un ambiente `Linux`, nos permite obtener información de cualquier otra instrucción de la que se desconozca su funcionalidad. Por último, sería bueno contar con este tipo de prácticas durante los primeros semestres de la carrera debido a que sistemas operativos como `Linux` son utilizados en diferentes materias y es de gran importancia tener nociones sobre su funcionamiento, principalmente de los comandos que se pueden ingresar desde la consola.

## Referencias

- Ríos, J. (2019). Notas del curso de Sistemas Operativos. Recuperado el 06 de febrero de 2019, del sitio web: Comunidad ITAM.