



Practica 02 “ GNU/Linux”

Facultad de Ingeniería

Laboratorios de docencia

Laboratorio de Computación Salas A y B

Profesor: Ing. Karina García Morales

Asignatura: Fundamentos de Programación

Grupo: 1121

No de Práctica(s): 02

Integrante(s): Daniel Yahuitl Sánchez

*No. de Equipo de
cómputo empleado:* 22

Semestre: 2019-1

Fecha de entrega: 26/08/2018

Observaciones:

CALIFICACIÓN: _____

Objetivos: Conocer la importancia del sistema operativo de una computadora, así como sus funciones. Explorar un sistema operativo GNU/Linux con el fin de conocer y utilizar los comandos básicos en GNU/Linux.

Desarrollo de la Práctica:

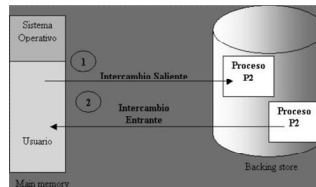
Introducción

El Sistema Operativo es el conjunto de programas y datos que administra los recursos tanto de hardware (dispositivos) como de software (programas y datos) de un sistema de cómputo y/o comunicación. Además funciona como interfaz entre la computadora y el usuario o aplicaciones.

En la actualidad existen diversos sistemas operativos; por ejemplo, para equipos de cómputo están Windows, Linux, Mac OS entre otros. Para el caso de dispositivos móviles se encuentran Android, IOS, Windows Phone entre otros. Cada uno de ellos tiene diferentes versiones y distribuciones que se ajustan a los diversos equipos de cómputo y comunicación en los que trabajan.

Los componentes de un sistema operativo, de forma general, son:

❓ Gestor de memoria,



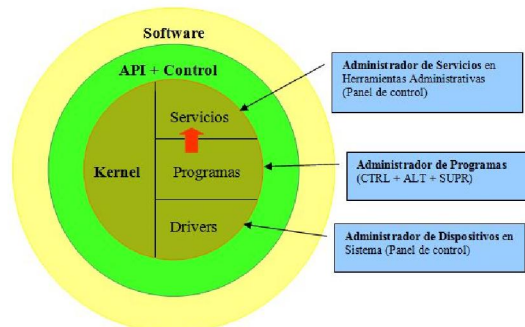
Nombre de im...	Nombre ...	CPU	Memoria ...	Descripción
BTTray.exe	autoClic	00	3.256 KB	Bluetooth...
ccApp.exe	autoClic	00	776 KB	Symantec...
csrss.exe	autoClic	00	1.392 KB	
dmv.exe	autoClic	00	45.364 KB	Administ...
eaudio.exe	autoClic	00	4.512 KB	Notificac...
edSLoader.exe	autoClic	00	3.040 KB	eDataSec...
explorer.exe	autoClic	01	30.652 KB	Explorador...
GrooveMnib...	autoClic	00	2.556 KB	GrooveM...
IAAnotif.exe	autoClic	00	2.180 KB	Event Mo...
ieuser.exe	autoClic	00	3.132 KB	Internet ...
ieupdate.exe	autoClic	00	12.840 KB	Internet ...
ieupdate.exe	autoClic	00	17.212 KB	Internet ...
mmc.exe	autoClic	00	9.056 KB	Microsoft ...
msinfo32.exe	autoClic	00	3.836 KB	Informac...
PMVService.exe	autoClic	00	2.476 KB	CyberLink...

❓ Administrador y planificador de procesos,

❓ Sistema de archivos y



❓ Administración de E/S



Comúnmente, estos componentes se encuentran en el kernel o núcleo del sistema operativo.

En cuanto a la Interfaz con el usuario, las hay de tipo texto y de tipo gráfico. En la actualidad, es común trabajar con la interfaz gráfica ya que facilita mucho seleccionar la aplicación a utilizar; inclusive esta selección se hace “tocando la pantalla” (técnica touch).

Sin embargo, cuando se desarrollan proyectos donde se elaborarán documentos y programas es necesario el uso de dispositivos de entrada y salida (hardware) y aplicaciones en modo texto (software).

Sistema Operativo Linux

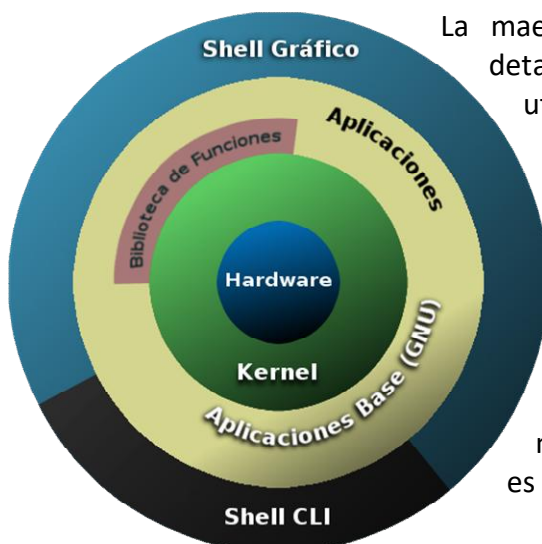
Linux es un sistema operativo tipo Unix de libre distribución para computadoras personales, servidores y estaciones de trabajo. El sistema está conformado por el núcleo (kernel) y un gran número de programas y bibliotecas. Muchos programas y bibliotecas han sido posibles gracias al proyecto GNU, por lo mismo, se conoce a este sistema operativo como GNU/Linux.

Licencia GNU

La Licencia Pública General de GNU o GNU General Public License (GNU GPL) es una licencia creada por la Free Software Foundation en 1989 y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Kernel de GNU/Linux

El kernel o núcleo de linux se puede definir como el corazón del sistema operativo. Es el encargado de que el software y el hardware del equipo se puedan comunicar. Sus componentes son los que se mencionaron en la introducción de esta práctica.



La maestra nos explicó el sistema operativo Kernel muy detalladamente y también lo de los comandos que utilizaríamos en la Mac que es muy similar su sistema operativo al de Linux.

De la figura, se puede observar que entre el kernel y las aplicaciones existe una capa que permite al usuario comunicarse con el sistema operativo y en general con la computadora, a través de programas que ya vienen instalados con la distribución de Linux (Debian, Ubuntu, Fedora, etc.) y trabajan ya sea en modo gráfico o en modo texto. Uno de estos programas es el Shell.

El Shell de GNU/Linux permite introducir órdenes (comandos) y ejecutar programas en el sistema operativo. Todas las órdenes de UNIX/Linux son programas que están almacenados en el sistema de archivos y a los que llamamos comandos, por lo tanto, todo en GNU/Linux se puede controlar mediante comandos.

Comandos básicos

La sintaxis que siguen los comandos es la siguiente:

`comando [-opciones] [argumentos]`

Esto es, el nombre del comando, seguido de algunas banderas (opciones) para modificar la ejecución del mismo y, al final, se puede incluir un argumento (ruta, ubicación, archivo, etcétera) dependiendo del comando. Tanto las opciones como los argumentos son opcionales.

Aquí se van a enlistar los comandos que utilizamos en la clase y su función.

ls y ls .

El comando `ls` permite listar los elementos que existen en alguna ubicación del sistema de archivos de Linux. Por defecto lista los elementos que existen en la ubicación actual; Linux nombra la ubicación actual con un punto (.) por lo que `ls` y `ls .` realizan exactamente lo mismo.

```
Invitado - -bash - 80x24
Last login: Wed Aug 22 19:28:37 on ttys000
Chipre26:~ Guest$ ls
Desktop      Movies      algebra     ernesto.txt
Documents    Music       algoritmo.txt expo
Downloads    Pictures    calculo     fundamentos
Library      Public     diego       quimica
Chipre26:~ Guest$
```

```
Invitado - -bash - 80x24
Last login: Wed Aug 22 19:30:39 on ttys000
Chipre26:~ Guest$ ls .
Desktop      Movies      algebra     ernesto.txt
Documents    Music       algoritmo.txt expo
Downloads    Pictures    calculo     fundamentos
Library      Public     diego       quimica
Chipre26:~ Guest$
```

ls -l

El comando `ls` realiza acciones distintas dependiendo de las banderas que utilice, por ejemplo, si se utiliza la opción `l` se genera un listado largo de la ubicación actual.

```
Invitado - -bash - 80x24
Last login: Wed Aug 22 19:25:22 on ttys000
Chipre26:~ Guest$ ls -l
total 8
drwx-----+ 3 Guest _guest 102 Aug 22 15:06 Desktop
drwx-----+ 3 Guest _guest 102 Aug 22 15:06 Documents
drwx-----+ 3 Guest _guest 102 Aug 22 15:06 Downloads
drwx-----@ 52 Guest _guest 1768 Aug 22 15:06 Library
drwx-----+ 3 Guest _guest 102 Aug 22 15:06 Movies
drwx-----+ 3 Guest _guest 102 Aug 22 15:06 Music
drwx-----+ 3 Guest _guest 102 Aug 22 15:06 Pictures
drwxr-xr-x+ 4 Guest _guest 136 Aug 22 15:06 Public
drwxr-xr-x 10 Guest _guest 340 Aug 22 15:50 algebra
-rw-r--r-- 1 Guest _guest 1 Aug 22 15:31 algoritmo.txt
drwxr-xr-x 9 Guest _guest 306 Aug 22 15:59 calculo
-rw-r--r-- 1 Guest _guest 0 Aug 22 15:16 diego
-rw-r--r-- 1 Guest _guest 0 Aug 22 15:19 ernesto.txt
drwxr-xr-x 9 Guest _guest 306 Aug 22 16:08 expo
drwxr-xr-x 9 Guest _guest 306 Aug 22 16:05 fundamentos
drwxr-xr-x 10 Guest _guest 340 Aug 22 15:41 quimica
Chipre26:~ Guest$
```

ls /

Es posible listar los elementos que existen en cualquier ubicación del sistema de archivos, para ello hay que ejecutar el comando especificando como argumento la ubicación donde se desean listar los elementos. Si queremos ver los archivos que se encuentran en la raíz, usamos.

ls /home

Para ver los usuarios del equipo local, revisamos el directorio home que parte de la raíz (/).

ls -l /home

Tanto las opciones como los argumentos se pueden combinar para generar una ejecución más específica.

```
Chipre26:~ Guest$ ls -l /home
Chipre26:~ Guest$
```

man ls

GNU/Linux proporciona el comando man, el cual permite visualizar la descripción de cualquier comando así como la manera en la que se puede utilizar.

ls /usr

Esto es, el argumento se inicia con / indicando que es el directorio raíz, seguido de usr que es el nombre del directorio. Cuando especificamos la ubicación de un archivo partiendo de la raíz, se dice que estamos indicando la “ruta absoluta” del archivo

```
Invitado -- -bash -- 80x24
Last login: Wed Aug 22 19:31:30 on ttys000
Chipre26:~ Guest$ ls /
Applications      dev
Developer         etc
LC_AB             home
Library           installer.failurerequests
Network           net
System            opt
Users             private
Volumes           sbin
anaconda3         tmp
bin               usr
cores             var
Chipre26:~ Guest$
```

```
Invitado -- -bash -- 80x24
Chipre26:~ Guest$ ls /home
Chipre26:~ Guest$
```

```
Invitado -- -bash -- 80x24
Chipre26:~ Guest$ man ls
[1]+  Stopped                  man ls
Chipre26:~ Guest$ ls /usr
X11      bin      lib      local      share
X11R6    include  libexec  sbin       standalone
Chipre26:~ Guest$
```

```
LS(1)                                BSD General Commands Manual                                LS(1)

NAME
  ls -- list directory contents

SYNOPSIS
  ls [-ABCFHLOPRSTUW@abcdeghiklmnopqrstuwx1] [file ...]

DESCRIPTION
  For each operand that names a file of a type other than directory, ls
  displays its name as well as any requested, associated information. For
  each operand that names a file of type directory, ls displays the names
  of files contained within that directory, as well as any requested, asso-
  ciated information.

  If no operands are given, the contents of the current directory are dis-
  played. If more than one operand is given, non-directory operands are
  displayed first; directory and non-directory operands are sorted sepa-
  rately and in lexicographical order.

  The following options are available:

  -@      Display extended attribute keys and sizes in long (-l) output.

  -1      (The numeric digit `one'.) Force output to be one entry per
  line. This is the default when output is not to a terminal.

  -A      List all entries except for . and ... Always set for the super-
  user.
```

ls .. o ls ../

Si bien el punto (.) es para indicar la ubicación actual, el doble punto (..) se utiliza para referirse al directorio “padre”. De esta forma si deseamos listar los archivos que dependen de mi directorio padre se escribe el este comando.

```
Chipre26:~ Guest$ ls ..
Guest      Shared      administrador  diplomado
Chipre26:~ Guest$
```

```
Chipre26:~ Guest$ ls ../
Guest      Shared      administrador  diplomado
Chipre26:~ Guest$
```

ls ../../usr

Con los primeros dos puntos se hace referencia al directorio home, con los siguientes dos puntos se refiere al directorio raíz, y finalmente se escribe el nombre del directorio usr.

```
Chipre26:~ Guest$ ls ../../usr
X11      bin      lib      local      share
X11R6    include  libexec  sbin      standalone
Chipre26:~ Guest$
```

touch nombre_archivo[.ext]

El comando touch permite crear un archivo de texto. En GNU/Linux no es necesario agregar una extensión al archivo creado, sin embargo, es recomendable hacerlo para poder identificar el tipo de archivo creado.

```
X11R6      include      libexec      sbin      standalone
Chipre26:~ Guest$ Touch Daniel.txt
Chipre26:~ Guest$
```

mkdir nombre_carpeta

El comando mkdir permite crear una carpeta.

```
Chipre26:~ Guest$ mkdir daniel yahuitl
Chipre26:~ Guest$
Chipre26:~ Guest$ ls
Daniel.txt  Library      Public      daniel      fundamentos
Desktop     Movies       algebra     diego       quimica
Documents   Music        algoritmo.txt ernesto.txt tareas
Downloads   Pictures     calculo     expo        yahuitl
Chipre26:~ Guest$
```

Para crear una carpeta en nuestra cuenta, que tenga como nombre “tareas” se escribe el siguiente comando: **mkdir tareas**

```
Chipre26:~ Guest$ mkdir tareas
Chipre26:~ Guest$ ls
Daniel.txt  Library      Public      daniel      fundamentos
Desktop     Movies       algebra     diego       quimica
Documents   Music        algoritmo.txt ernesto.txt tareas
Downloads   Pictures     calculo     expo        yahuitl
Chipre26:~ Guest$
```


cd nombre_carpeta

El comando `cd` permite ubicarse en una carpeta.

```
tareas — -bash — 81x25
Chipre26:tareas Guest$ cd tareas
-bash: cd: tareas: No such file or directory
Chipre26:tareas Guest$ pwd
/Users/Guest/tareas
Chipre26:tareas Guest$
```

Por lo que si queremos situarnos en la carpeta “tareas” creada anteriormente, se escribe el comando: **cd tareas**

cd ..

Ahora, si deseamos situarnos en la carpeta de inicio de nuestra cuenta, que es la carpeta padre, escribimos este comando.

```
[Chipre26:tareas Guest$ cd ..
[Chipre26:~ Guest$ pwd
/Users/Guest
Chipre26:~ Guest$
```

pwd

El comando `pwd` permite conocer la ubicación actual(ruta).

```
[Chipre26:tareas Guest$ pwd
/Users/Guest/tareas
[Chipre26:tareas Guest$ cd ..
[Chipre26:~ Guest$ pwd
/Users/Guest
```

find . -name cadena_buscar

El comando `find` permite buscar un elemento dentro del sistema de archivos. Al comando `find` hay que indicarle en qué parte del sistema de archivos va a iniciar la búsqueda. En el ejemplo

```
[Chipre26:~ Guest$ cd ..
[Chipre26:~ Guest$ pwd
/Users/Guest
[Chipre26:~ Guest$ find . -daniel
find: -daniel: unknown primary or operator
Chipre26:~ Guest$ clear
```

anterior la búsqueda se inicia en la posición actual (uso de `.`). Además, utilizando la bandera `-name` permite determinar la cadena a buscar (comúnmente es el nombre de un archivo). Si queremos encontrar la ubicación del archivo `tareas`, se escribe el siguiente comando: **find . -name tareas**

clear

El comando `clear` permite limpiar la consola.

```
Invitado — -bash — 96x24
Chipre26:~ Guest$
```

cp archivo_origen archivo_destino

El comando `cp` permite copiar un archivo. Si queremos una copia del archivo `datos.txt` con nombre `datosViejos.txt` en el mismo directorio, entonces se escribe el comando **cp datos.txt**

datosViejos.txt Ahora, si requerimos una copia de un archivo que está en la carpeta padre en la ubicación actual y con el mismo nombre, entonces podemos emplear las rutas relativas de la siguiente forma: **cp ../archivo_a_copiar** . Es muy importante indicar como archivo destino al punto (`.`) para que el archivo de copia se ubique en el directorio actual.

```
Invitado — -bash — 96x24
Chipre26:~ Guest$ user/guest/tareas
-bash: user/guest/tareas: No such file or directory
Chipre26:~ Guest$
Chipre26:~ Guest$ pwd
/Users/Guest
Chipre26:~ Guest$ ls
Daniel.txt  Library      Public      daniel      fundamentos
Desktop     Movies      algebra     diego       quimica
Documents  Music       algoritmo.txt ernesto.txt tareas
Downloads  Pictures    calculo     expo        yahuitl
Chipre26:~ Guest$
Chipre26:~ Guest$ cp daniel.txt/tareas
usage: cp [-R [-H | -L | -P]] [-f] [-n] [-apvXc] source_file target_file
usage: cp [-R [-H | -L | -P]] [-f] [-n] [-apvXc] source_file ... target_directory
Chipre26:~ Guest$ cp daniel.txt tareas
Chipre26:~ Guest$
```

mv ubicación_origen/archivo ubicación_destino

El comando mv mueve un archivo de un lugar a otro, en el sistema de archivos. El comando mueve el archivo desde su ubicación origen hacia la ubicación

```
Chipre26:~ Guest$ mv sanchez.txt /usr/guest/tareas/
mv: rename sanchez.txt to /usr/guest/tareas/: No such file or directory
Chipre26:~ Guest$ ls
Daniel.txt      Library      Public      daniel      fundamentos  yahuitl
Desktop        Movies      algebra     diego       quimica      sanchez.txt
Documents      Music       algoritmo.txt ernesto.txt sanchez.txt
Downloads      Pictures    calculo     expo        tareas
Chipre26:~ Guest$ cd tareas
Chipre26:tareas Guest$ ls
daniel.txt
Chipre26:tareas Guest$ █
```

deseada(destino). Si queremos que un archivo que está en la carpeta padre, reubicarlo en el directorio actual y con el mismo nombre, entonces podemos emplear las rutas relativas de la siguiente forma: **mv ../archivo_a_reubicar** . Este comando también puede ser usado para cambiar el nombre de un archivo, simplemente se indica el nombre actual del archivo y el nuevo nombre: **mv nombre_actual_archivo nombre_nuevo_archivo**

rm nombre_archivo rm nombre_carpeta

El comando rm permite eliminar un archivo o un directorio. Cuando la carpeta que se desea borrar contiene información, se debe utilizar la bandera -f para forzar la eliminación. Si la carpeta contiene otras carpetas, se debe utilizar la opción -r, para realizar la eliminación recursiva.

```
Chipre26:~ Guest$ rm daniel.txt
Chipre26:~ Guest$ ls
Desktop        Movies      algebra     diego       quimica      sanchez.txt
Documents      Music       algoritmo.txt ernesto.txt sanchez.txt
Downloads      Pictures    calculo     expo        tareas
Library        Public      daniel      fundamentos  yahuitl
Chipre26:~ Guest$ rm -r
rm: rm: No such file or directory
rm: -r: No such file or directory
Chipre26:~ Guest$
Chipre26:~ Guest$
Chipre26:~ Guest$ rm -r tareas
Chipre26:~ Guest$
Chipre26:~ Guest$ ls
Desktop        Movies      algebra     diego       quimica      sanchez.txt
Documents      Music       algoritmo.txt ernesto.txt sanchez.txt
Downloads      Pictures    calculo     expo        yahuitl
Library        Public      daniel      fundamentos
Chipre26:~ Guest$ █
```


Ejercicio de Tarea:

1. Crea un directorio que se llame "FUNDAMENTOS" y entra a el.

```
C:\WINDOWS\system32\cmd.exe
```

```
Microsoft Windows [Versión 10.0.16299.431]  
Copyright (c) 2017 Microsoft Corporation. Todos los derechos reservados.  
  
Users\Daniel_Sanchez>MKDIR FUNDAMENTOS  
Se ha creado el subdirectorio o el archivo FUNDAMENTOS.
```

2. Crea un directorio que se llame "FUNDAMENTOS2" y entra a el.

```
Users\Daniel_Sanchez>MKDIR FUNDAMENTOS2  
  
Users\Daniel_Sanchez>CD FUNDAMENTOS2
```

3. Crea un archivo llamado "NOMBRE" con tu primer nombre (puedes utilizar algún editor gráfico de Linux en terminal).

```
C:\Users\Daniel_Sanchez\FUNDAMENTOS2>COPY CON NOMBRE.TXT  
DANIEL  
^Z
```

4. Crea un archivo llamado "NUMEROS" con tu número de cuenta.

```
C:\Users\Daniel_Sanchez\FUNDAMENTOS2>COPY CON NUMEROS.TXT  
316081270  
^Z
```

5. Crea un archivo nuevo llamado "DATOS" y copia los datos de los archivos "NOMBRE" y "NUMEROS".

```
C:\Users\Daniel_Sanchez\FUNDAMENTOS2>COPY CON DATOS.TXT  
COPY NOMBRE.TXT  
COPY NUMEROS.TXT
```

6. Muestra en pantalla el contenido de ambos archivos, uno a la vez y con que comando lo muestras.

```
C:\Users\Daniel_Sanchez\FUNDAMENTOS2>FC NOMBRE.TXT NUMEROS.TXT  
Comparando archivos NOMBRE.TXT y NUMEROS.TXT  
***** NOMBRE.TXT  
DANIEL  
***** NUMEROS.TXT  
316081270  
*****
```

7. Salir del directorio FUNDAMENTOS2.

8. Crea un directorio que se llame COPIA1.

```
C:\Users\Daniel_Sanchez\FUNDAMENTOS2>MKDIR COPIA1
```

9. Copia los archivos NOMBRE y NUMEROS que creaste, al directorio COPIA1.

```
C:\Users\Daniel_Sanchez\FUNDAMENTOS2\COPIA1>COPY CON NOMBRE.TXT
^Z
    1 archivo(s) copiado(s).

C:\Users\Daniel_Sanchez\FUNDAMENTOS2\COPIA1>COPY CON NUMEROS.TXT
^Z
    1 archivo(s) copiado(s).
```

10. Muestra el contenido de tu carpeta.

```
Directorio de C:\Users\Daniel_Sanchez\FUNDAMENTOS2\COPIA1

28/08/2018  02:46 a. m.      <DIR>          .
28/08/2018  02:46 a. m.      <DIR>          ..
28/08/2018  02:45 a. m.              0 NOMBRE.TXT
28/08/2018  02:46 a. m.              0 NUMEROS.TXT
                2 archivos              0 bytes
                2 dirs 414,387,326,976 bytes libres
```

11. Mueve el archivo DATOS a la carpeta COPIA1

12. Muestra en la pantalla el número de líneas, palabras y caracteres en el archivo números.

```
C:\Users\Daniel_Sanchez\FUNDAMENTOS2>COPY CON NUMEROS.TXT
316081270
^Z
```

13. Muestra los permisos de los archivos contenidos en el directorio FUNDAMENTOS y en DATOS, explica cada uno de ellos.

```
C:\Users\Daniel_Sanchez\FUNDAMENTOS2\COPIA1>DIR DATOS.TXT
El volumen de la unidad C es Windows
El número de serie del volumen es: 20F8-D145

C:\Users\Daniel_Sanchez\FUNDAMENTOS2\COPIA1>DIR FUNDAMENTOS
El volumen de la unidad C es Windows
El número de serie del volumen es: 20F8-D145
```

14. Indica el directorio en el que te encuentras y con que comando lo muestras.

```
Directorio de C:\Users\Daniel_Sanchez\FUNDAMENTOS2
```

15. Teclea el comando cal y escribe lo que muestra.

```
C:\Users\Daniel_Sanchez\FUNDAMENTOS2\COPIA1>CAL
"CAL" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\Daniel_Sanchez\FUNDAMENTOS2\COPIA1>cal
"cal" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
```

16. Teclea el comando date y escribe la salida.

```
C:\Users\Daniel_Sanchez\FUNDAMENTOS2\COPIA1>DATE
La fecha actual es: 28/08/2018
Escriba la nueva fecha: (dd-mm-aa)
```

17. Utiliza los comandos man/info para investigar la forma de usar el resto de los comandos requeridos para la práctica.

```
C:\Users\Daniel_Sanchez\FUNDAMENTOS2\COPIA1>MAN/INFO
"MAN" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\Daniel_Sanchez\FUNDAMENTOS2\COPIA1>man/info
"man" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
```

Conclusiones

Realizar esta práctica fue muy interesante, además aprendí a hacer muchas cosas que yo creía que eran más complejas de hacer, si cuesta un poco de trabajo agarrarle o tomarle sentido a los comandos, pero la realización de la práctica en clase fue muy buena. También fue muy importante que conociéramos el sistema operativo y lo manejáramos.

Por otro lado la realización de los ejercicios de tarea si fue mucho más compleja de realizar en Windows, ya que la mayoría de comandos son diferentes y es muy estricto y específico a la hora de recibir un comando, pero creo que aun así se pudieron rescatar algunos ejercicios, sin embargo de verdad fue muy difícil elaborarlos. Lo bueno de esto es que conocí otro sistema operativo y su manera de funcionamiento.

Bibliografía

♣ Óscar Vicente Huguet Soriano, Sonia Doménech Gómez. Introducción a Linux. [Figura 1]. Consulta: Junio de 2015. Disponible en: http://mural.uv.es/oshuso/81_introduccion_a_linux.html

♣ Pablo Delgado. Integración de sistemas. Linux y su sistema gestor de ficheros (descripciones).[Figura 2]. Consulta agosto de 2016. Disponible en: <http://todobytes.es/2014/09/integracion-de-sistemas-linux-y-su-sistema-gestor-deficheros-descripciones/>

