



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de Programación

Grupo: 135

No de Práctica(s): 2

Integrante(s): Torres Alcántara Alan Eliezer

*No. de Equipo de
cómputo empleado:* Suiza 50

No. de Lista o Brigada: 9032

Semestre: 2020 - 1

Fecha de entrega: 26/08/2019

Observaciones: Excelente

CALIFICACIÓN: 10

GNU /LINUX

Objetivo

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.

1. Sistema operativo

Linux es un sistema operativo que forma parte del movimiento de *software libre*.

- *kernel*
- *compiladores*
- interfáz
- editores
- redes
- etc.



Software libre

Todo programa que cumple con ciertas libertades:

- La libertad de utilizar el programa para cualquier fin, y sin restricciones.
- La libertad de estudiar cómo funciona el programa y modificarlo según necesidades o preferencias particulares.
- La libertad de compartir el programa.
- La libertad de hacer cambios al programa y distribuir la versión modificada.

Estas libertades se pueden proteger mediante licencias como la GPL. Esto obliga a las versiones modificadas a mantener las mismas libertades.

El software libre implica tener un código abierto.

- Colaboración
- Sin problemas de propiedad intelectual
- Sin contratos de compra de software.
- Retroalimentación.
- Desarrollo rápido y de calidad

Kernel

Es la parte del *software* de una computadora que no se ve, y que de hecho no debe ser vista. Es usado por los programas no por los usuarios.

- Asigna recursos (memoria).
- Gestiona y vincula procesos.
- Comunica periféricos (drivers).
- Da acceso a los archivos.
- Gestiona el uso de red.



2. Linux

Terminal

```
ALAN@FAMILIATORRESA MINGW64 ~  
$
```

Comandos

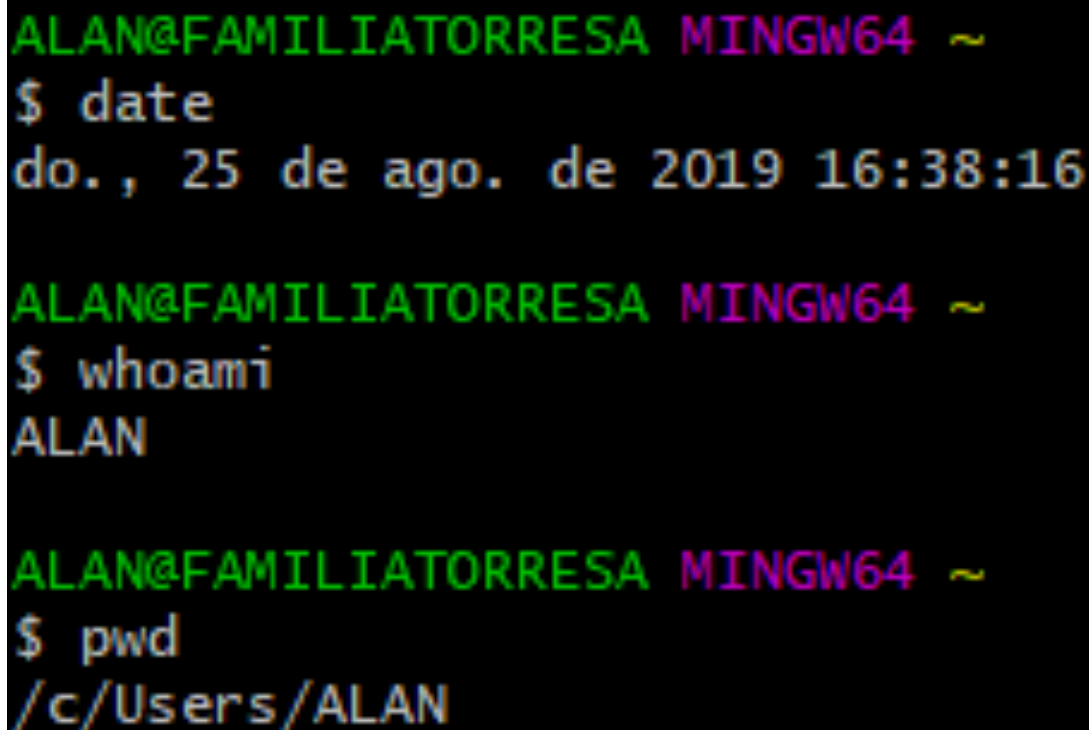
Son las instrucciones que se pueden usar en la terminal, y generalmente tienen la siguiente estructura:

\$ **comando** -**opciones** **argumentos**

Noten dos cosas importantes:

- Los espacios entre cada elemento
- El signo "\$" al inicio significa que la terminal está esperando **instrucciones**, si no lo ves es porque la terminal está trabajando en algo o esperando **entrada estándar**

Comandos sencillos



```
ALAN@FAMILIATORRESA MINGW64 ~  
$ date  
do., 25 de ago. de 2019 16:38:16  
  
ALAN@FAMILIATORRESA MINGW64 ~  
$ whoami  
ALAN  
  
ALAN@FAMILIATORRESA MINGW64 ~  
$ pwd  
/c/Users/ALAN
```

date sirve para ver la fecha, **whoami** para ver el nombre del usuario y **pwd** se usa para saber en qué carpeta está uno

Navegación de carpetas

```
ALAN@FAMILIATORRESA MINGW64 ~
$ cd /PROYECTO/

ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ cd /
archivo1          dev/          LICENSE.txt      tmp/
archivo6          etc/          mingw64/         unins000.dat
archivo7          git-bash.exe  proc/           unins000.exe
bin/              git-cmd.exe  PROYECTO/       unins000.msg
cmd/              home/        ReleaseNotes.html usr/

ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ cd /home/

ALAN@FAMILIATORRESA MINGW64 /home
$ cd ..

ALAN@FAMILIATORRESA MINGW64 /
$ cd
```

Con las diagonales (/) indicamos que pasamos de una carpeta a otra, con los dos puntos (..) se pasa a la carpeta anterior y al utilizar solo **cd** regresamos a la carpeta personal

Operaciones con archivos

```
ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ touch archivo1.txt archivo2.txt

ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ rm archivo1.txt

ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ cp archivo2.txt archivo1.txt

ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ touch archivo3.txt

ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ mv archivo2.txt archivo3.txt

ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ mkdir carpeta
```

touch sirve para crear archivos, **rm** para borrarlos, **cp** para copiarlos, **mv** para renombrarlos y **mkdir** para crear una carpeta

Lista de archivos

```
ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ ls
archivo1      archivo3.txt  archivo7     no_temp      README
archivo1.txt  archivo6      carpeta/     pruebas.temp

ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ ls -al
total 16
drwxr-xr-x 1 ALAN 197611  0 ago. 25 16:59 ./
drwxr-xr-x 1 ALAN 197611  0 ago. 25 16:50 ../
drwxr-xr-x 1 ALAN 197611  0 ago. 24 21:10 .git/
-rw-r--r-- 1 ALAN 197611  7 ago. 24 20:07 .gitignore
-rw-r--r-- 1 ALAN 197611  0 ago. 24 21:01 archivo1
-rw-r--r-- 1 ALAN 197611  0 ago. 25 16:58 archivo1.txt
-rw-r--r-- 1 ALAN 197611  0 ago. 25 16:57 archivo3.txt
-rw-r--r-- 1 ALAN 197611 40 ago. 24 20:52 archivo6
-rw-r--r-- 1 ALAN 197611  0 ago. 24 21:08 archivo7
drwxr-xr-x 1 ALAN 197611  0 ago. 25 16:59 carpeta/
-rw-r--r-- 1 ALAN 197611 49 ago. 24 20:37 no_temp
-rw-r--r-- 1 ALAN 197611  0 ago. 24 20:07 pruebas.temp
-rw-r--r-- 1 ALAN 197611 69 ago. 24 20:01 README
```

ls nos muestra la lista de archivos, si queremos una lista con mayor detalle debemos escribir **ls -al**

Ayuda sobre comandos

```
$ man ls
```

Muestra el **manual** del comando, con ayuda, ejemplos y más información útil.

```
$ ls --help
```

Muestra una descripción menos detallada que el manual, pero también de gran **ayuda** para usar el comando

ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)

\$ ls --help

Usage: ls [OPTION]... [FILE]...

List information about the FILES (the current directory by default).

Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.

-a, --all	do not ignore entries starting with .
-A, --almost-all	do not list implied . and ..
--author	with -l, print the author of each file
-b, --escape	print C-style escapes for nongraphic characters
--block-size=SIZE	with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below
-B, --ignore-backups	do not list implied entries ending with ~
-c	with -lt: sort by, and show, ctime (time of last modification of file status information); with -l: show ctime and sort by name; otherwise: sort by ctime, newest first
-C	list entries by columns
--color[=WHEN]	colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below
-d, --directory	list directories themselves, not their contents
-D, --dired	generate output designed for Emacs' dired mode
-f	do not sort, enable -aU, disable -ls --color
-F, --classify	append indicator (one of */=>@) to entries
--file-type	likewise, except do not append '='
--format=WORD	across -x, commas -m, horizontal -x, long -l, single-column -l, verbose -l, vertical -C
--full-time	like -l --time-style=full-iso
-g	like -l, but do not list owner
--group-directories-first	group directories before files; can be augmented with a --sort option, but any use of --sort=none (-U) disables grouping
-G, --no-group	in a long listing, don't print group names
-h, --human-readable	with -l and -s, print sizes like 1K 234M 2G etc.
--si	likewise, but use powers of 1000 not 1024
-H, --dereference-command-line	follow symbolic links listed on the command line
--dereference-command-line-symlink-to-dir	follow each command line symbolic link that points to a directory
--hide=PATTERN	do not list implied entries matching shell PATTERN (overridden by -a or -A)
--hyperlink[=WHEN]	hyperlink file names; WHEN can be 'always' (default if omitted), 'auto', or 'never'
--indicator-style=WORD	append indicator with style WORD to entry names: none (default), slash (-p), file-type (--file-type), classify (-F)
-i, --inode	print the index number of each file
-I, --ignore=PATTERN	do not list implied entries matching shell PATTERN
-k, --kibibytes	default to 1024-byte blocks for disk usage; used only with -s and per directory totals

Examinar archivos (diferentes comandos)

\$ echo Libro.txt	Escribe "Libro.txt"
\$ cat Libro.txt	Imprime el contenido del archivo
\$ more Libro.txt	Imprime el contenido por partes
\$ less Libro.txt	Imprime por partes y permite volver
\$ grep felicidad Libro.txt	Imprime sólo las líneas que contienen "felicidad"

No olvides las maneras de obtener **ayuda** sobre los comandos

Project Gutenberg's El paraíso de las mujeres, by Vicente Blasco Ibañez

This eBook is for the use of anyone anywhere at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this eBook or online at www.gutenberg.net

Title: El paraíso de las mujeres
Novela

Author: Vicente Blasco Ibañez

Release Date: January 24, 2004 [EBook #10822]

Language: Spanish

Character set encoding: ISO Latin-1

*** START OF THIS PROJECT GUTENBERG EBOOK EL PARAISO DE LAS MUJERES ***

Produced by Stan Goodman, Virginia Paque and PG Distributed Proofreaders

Ejemplo de **cat** Libro.txt

--Algún murciélago--volvió á decirse.

Sus ojos creyeron ver en la lóbreguez algo más obscuro aún que pasaba, flotando en el aire, por encima de su rostro. De este pájaro de la noche surgieron repentinamente dos puntos de luz, dos pequeños focos de intensa blancura, iguales á unos ojos hechos con diamantes. Un par de rayos sutiles pero intensísimos se pasearon á lo largo de su cuerpo, iluminándole desde la frente hasta la punta de los pies. El ingeniero, asombrado por el supuesto murciélago, levantó un brazo, abofeteando al vacío. Instantáneamente, el misterioso volador apagó los rayos de sus ojos, alejándose con un chillido de velocidad forzada que le hizo perderse á lo lejos en unos cuantos segundos.

Esta visita quitó el sueño á Edwin, obligándole á sentarse sobre la pequeña pradera que le servía de cama. Sus ojos pudieron ver entonces por encima de los matorrales varios puntos de luz que se movían con una evolución rítmica, cambiando la intensidad y el color de sus resplandores.

--Indudablemente son luciérnagas--murmuró--; luciérnagas de este país, distintas á todas las que conozco.

Las había de una blancura ligeramente azul, como la de los más ricos diamantes; otras eran de verde esmeralda, de topacio, de ópalo, de zafiro. Parecía que sobre el terciopelo negro de la noche todas las piedras preciosas conocidas por los hombres se deslizasen como en una contradanza. Volaban formando parejas, y sus rayos, al cruzarse, se esparcían en distintas direcciones.

Gillespie encontraba cada vez más interesante este desfile aéreo; pero de pronto, como si obedeciesen á una orden, todos los fulgores se extinguieron á un tiempo. En vano aguardó pacientemente. Parecía que los insectos luminosos se hubiesen enterado de su presencia al tocar con algunos de sus rayos la cabeza que surgía curiosa sobre los matorrales.

Pasó mucho tiempo sin que la obscuridad volviera á cortarse con la menor raya de luz, y Edwin sintió el desencanto de un público cuando se convence de que es inútil esperar la continuación de un espectáculo. Volvió á tenderse, buscando otra vez el sueño; pero, al descansar la cabeza en la hierba, oyó junto á sus orejas unos trotecillos medrosos y unos gritos de susto. Hasta sintió en su cogote el roce de varios animalejos que parecían haberse librado casualmente por unos milímetros de morir aplastados.

Entradas y salidas

Entradas

- **Estándar:** Es el **texto** que reciben algunos **comandos** para procesarlos y después dar una salida.
- **Argumentos o parámetros:** Pueden ser archivos, rutas, variables, entre otros; y el **comando** los requiere **antes** de correr.

Salidas

- **Siempre es estándar**, y es lo que el programa devuelve después de procesar lo que se le pidió

Redireccionamiento

Hacia archivo (> y >>)

Mandan la salida del **comando** al archivo indicado en lugar de a la **salida estándar**

```
ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ touch ahora.txt

ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ date > ahora.txt

ALAN@FAMILIATORRESA MINGW64 /PROYECTO (master)
$ date >> ahora.txt
```

Con > se reemplaza el contenido del archivo **ahora.txt** por la salida de **date**

Con >> se añade la salida de **date** al archivo **ahora.txt**

 ahora: Bloc de notas

Archivo Edición Formato Ver Ayuda

do., 25 de ago. de 2019 17:37:51do., 25 de ago. de 2019 17:38:08