



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Marco Antonio Martínez Quintana

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): 2

Integrante(s): Ramírez Pérez Daniela Itzel

*No. de Equipo de
cómputo empleado:* No aplica

No. de Lista o Brigada: 36

Semestre: 1er

Fecha de entrega: Viernes 16 de octubre

Observaciones:

CALIFICACIÓN: _____

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.

Introducción

Un Sistema Operativo es un conjunto de programas y datos que administra los recursos tanto de dispositivos como de programas y datos de un sistema de cómputo y/o comunicación.

Algunos sistemas operativos de computadoras son:

- Windows
- Linux
- Mac OS entre otros.

Para el caso de dispositivos móviles se encuentran

- Android
- IOS
- Windows Phone

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

- Gestor de memoria,
- Administrador y planificador de procesos,
- Sistema de archivos y
- Administración de E/S.

Hay que también agregar la Interfaz con el usuario siendo:

- Texto
- Gráfico

Con la interfaz gráfica siendo más utilizada, con el usuario solo teniendo que “tocar la pantalla” (técnica touch).

Comandos

comando ls	
<p>Permite listar los elementos que existen en alguna ubicación del sistema de archivos de Linux.</p> <p>ls y ls .</p>	<pre>[fp03alu36@samba ~]\$ ls Escritorio</pre> <pre>[fp03alu36@samba ~]\$ ls . Escritorio</pre>
<p>Se utiliza la opción -l para generar un listado largo de la ubicación actual:</p> <p>ls -l</p>	<pre>[fp03alu36@samba ~]\$ ls -l Escritorio</pre>
<p>Listar los elementos que existen en cualquier ubicación del sistema de archivos:</p> <p>ls /</p>	<pre>[fp03alu36@samba ~]\$ ls / bin database etc lib boot dev home lib64 lost+found mnt proc run sitio media opt root sbin software srv tmp usr webserver sys users var</pre>
<p>Ver los usuarios del equipo local, revisamos el directorio home que parte de la raíz (/):</p> <p>ls /home</p>	<pre>[fp03alu36@samba ~]\$ ls /home administrador mena MNR squid</pre>
<p>Para generar una ejecución más Específica, tanto las opciones como los argumentos se pueden combinar para:</p> <p>ls -l /home</p>	<pre>[fp03alu36@samba ~]\$ ls -l /home administrador mena MNR squid</pre>

comando **man**

Permite visualizar la descripción de cualquier comando, así como la manera en la que se puede utilizar.

man ls

```
ls(1)                                User Commands                                ls(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -ftwsX 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
      scale sizes by SIZE before printing them. E.g., '--block-size=M' prints sizes in units of 1,048,576 bytes. 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 -li: show ctime and sort by name otherwise: sort by ctime, newest first

  -C
      list entries by columns

  --color[=WHEN]
      colorize the output. WHEN defaults to 'always' or can be 'never' or 'auto'. More info below

  -d, --directory
      list directory entries instead of contents, and do not dereference symbolic links

  -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

  -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.

      augment 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, print sizes in human readable format (e.g., 1K 234M 2G)

  --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)

  --indicator-style=WORD
      append indicator with style WORD to entry names: none (default), slash (-p), file-type (--file-type), classify (-F)

  -l, --inode
      print the index number of each file

  -I, --ignore=PATTERN
      do not list implied entries matching shell PATTERN

  -k, --kibibytes
      use 1024-byte blocks

  -l
      use a long listing format

  -L, --dereference
      when showing file information for a symbolic link, show information for the file the link references rather than for the link itself

  -m
      fill width with a comma separated list of entries

  -n, --numeric-uid-gid
      like -l, but list numeric user and group IDs

  -N, --literal
```

```

-N, --literal
    print raw entry names (don't treat e.g. control characters specially)
-o
    like -l, but do not list group information
-P, --indicator-style=slash
    append / indicator to directories
-g, --hide-control-chars
    print ? instead of non graphic characters
--show-control-chars
    show non graphic characters as-is (default unless program is 'ls' and output is a terminal)
-Q, --quote-name
    enclose entry names in double quotes
--quoting-style=WORD
    use quoting style WORD for entry names: literal, locale, shell, shell-always, c, escape
-r, --reverse
    reverse order while sorting
-R, --recursive
    list subdirectories recursively
-s, --size
    print the allocated size of each file, in blocks
-S
    sort by file size
--sort=WORD
    sort by WORD instead of name: none -U, extension -X, size -S, time -t, version -v
--time=WORD
    with -l, show time as WORD instead of modification time: atime -u, access -u, use -u, ctime -c, or
    status -c; use specified time as sort key if --sort=time
--time-style=STYLE
    with -l, show times using style STYLE: full-iso, long-iso, iso, locale, +FORMAT. FORMAT is inter-
    preted like "date"; if FORMAT is %FORMAT%a%P%T, %FORMAT% applies to non-recent files and
    %FORMAT%2 to recent files; if STYLE is prefixed with 'posix-', STYLE takes effect only outside the
    POSIX locale
-t
    sort by modification time, newest first
-T, --tabsize=COLS
    assume tab stops at each COLS instead of 8
-u
    with -lt: sort by, and show, access time with -l: show access time and sort by name otherwise: sort
    by access time
-U
    do not sort; list entries in directory order
-v
    natural sort of (version) numbers within text
-W, --width=COLS
    assume screen width instead of current value
-x
    list entries by lines instead of by columns
-X
    list entries by lines instead of by columns
-X
    sort alphabetically by entry extension
-l
    list one file per line

Linux options:
--icontext
    Display security context. Enable -l. Lines will probably be too wide for most displays.
-Z, --context
    Display security context so it fits on most displays. Displays only mode, user, group, security
    context and file name.
--scontext
    Display only security context and file name.
--help
    Display this help and exit
--version
    output version information and exit

SIZE is an integer and optional unit (example: 10M is 10*1024*1024). Units are K, M, G, T, P, E, Z, Y
(powers of 1024) or Ki, Mi, ... (powers of 1000).

Using color to distinguish file types is disabled both by default and with --color=none. With
--color=auto, ls emits color codes only when standard output is connected to a terminal. The LS_COLORS
environment variable can change the settings. Use the dircolors command to set it.

Exit status:
0      if OK,
1      if minor problems (e.g., cannot access subdirectory),
2      if serious trouble (e.g., cannot access command-line argument).

AUTHOR
    Written by Richard M. Stallman and David Mackenzie.

REPORTING BUGS
    Report ls bugs to bug-coreutils@gnu.org
    GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
    General help using GNU software: <http://www.gnu.org/gethelp/>
    Report ls translation bugs to <http://translationproject.org/team/>

COPYRIGHT
    Copyright © 2012 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later
    <http://gnu.org/licenses/gpl.html>.
    This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent
    permitted by law.

SEE ALSO
    The full documentation for ls is maintained as a Texinfo manual. If the info and ls programs are properly
    installed at your site, the command

        info coreutils 'ls invocation'

    should give you access to the complete manual.

```

Comando para “navegar” por el sistema de archivos Linux	
Para ver la lista de los archivos del directorio usr, podemos escribir el comando: ls /usr	
<pre>[fp03alu36@samba ~]\$ ls /usr bin etc games include lib lib64 libexec local sbin share src tmp</pre>	
Comandos ls .. o ls ../	
Si se desea listar los archivos que dependen del directorio padre, el doble punto (..) se utiliza para referirse al directorio “padre”. De esta forma se escribe el siguiente comando: ls .. o ls ../	
<pre>[fp03alu36@samba ~]\$ ls .. fp03alu01 fp03alu07 fp03alu13 fp03alu19 fp03alu25 fp03alu31 fp03alu37 fp03alu43 fp03alu49 fp03alu55 fp03alu02 fp03alu08 fp03alu14 fp03alu20 fp03alu26 fp03alu32 fp03alu38 fp03alu44 fp03alu50 fp03alu56 fp03alu03 fp03alu09 fp03alu15 fp03alu21 fp03alu27 fp03alu33 fp03alu39 fp03alu45 fp03alu51 fp03alu57 fp03alu04 fp03alu10 fp03alu16 fp03alu22 fp03alu28 fp03alu34 fp03alu40 fp03alu46 fp03alu52 fp03alu05 fp03alu11 fp03alu17 fp03alu23 fp03alu29 fp03alu35 fp03alu41 fp03alu47 fp03alu53 fp03alu06 fp03alu12 fp03alu18 fp03alu24 fp03alu30 fp03alu36 fp03alu42 fp03alu48 fp03alu54 [fp03alu36@samba ~]\$ ls ../ fp03alu01 fp03alu07 fp03alu13 fp03alu19 fp03alu25 fp03alu31 fp03alu37 fp03alu43 fp03alu49 fp03alu55 fp03alu02 fp03alu08 fp03alu14 fp03alu20 fp03alu26 fp03alu32 fp03alu38 fp03alu44 fp03alu50 fp03alu56 fp03alu03 fp03alu09 fp03alu15 fp03alu21 fp03alu27 fp03alu33 fp03alu39 fp03alu45 fp03alu51 fp03alu57 fp03alu04 fp03alu10 fp03alu16 fp03alu22 fp03alu28 fp03alu34 fp03alu40 fp03alu46 fp03alu52 fp03alu05 fp03alu11 fp03alu17 fp03alu23 fp03alu29 fp03alu35 fp03alu41 fp03alu47 fp03alu53 fp03alu06 fp03alu12 fp03alu18 fp03alu24 fp03alu30 fp03alu36 fp03alu42 fp03alu48 fp03alu54</pre>	
Comando ls ../../usr	
Si nuestra cuenta depende de home, la ruta relativa para listar los archivos de del directorio usr es: 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.	<pre>[fp03alu36@samba ~]\$ ls ../../usr</pre>

(comando touch)	
Crea un archivo de texto, su sintaxis es la siguiente: touch nombre_archivo[.ext]	<pre>[fp03alu36@samba ~]\$ touch nombre_archivo[.ext]</pre>

(comando mkdir)	
El comando mkdir permite crear una carpeta, su sintaxis es la siguiente: mkdir nombre_carpeta	<pre>[fp03alu36@samba ~]\$ mkdir nombre_carpeta</pre>
Para crear una carpeta en nuestra cuenta, que tenga como nombre "tareas" se escribe el siguiente comando: mkdir tareas	<pre>[fp03alu36@samba ~]\$ mkdir tareas</pre>

(comando cd)	
El comando cd permite ubicarse en una carpeta, su sintaxis es la siguiente: cd nombre_carpeta	<pre>[fp03alu36@samba ~]\$ cd nombre_carpeta</pre>
Por lo que si queremos situarnos en la carpeta "tareas" creada anteriormente, se escribe el comando: cd tareas	<pre>[fp03alu36@samba nombre_carpeta]\$ cd tareas</pre>
Ahora, si deseamos situarnos en la carpeta de inicio de nuestra cuenta, que es la carpeta padre, escribimos el comando: cd ..	<pre>[fp03alu36@samba tareas]\$ cd ..</pre>

(comando pwd)	
El comando pwd permite conocer la ubicación actual(ruta), su sintaxis es la siguiente: pwd	<pre>[fp03alu36@samba nombre_carpeta]\$ pwd</pre>

(comando find)	
<p>El comando find permite buscar un elemento dentro del sistema de archivos, su sintaxis:</p> <p>find . -name cadena_buscar</p> <p>Al comando find hay que indicarle en qué parte del sistema de archivos va a iniciar la búsqueda.</p>	<pre data-bbox="570 233 1520 268">[fp03alu36@samba nombre_carpeta]\$ find . -name cadena_buscar</pre>

(comando clear)	
<p>El comando clear permite limpiar la consola, su sintaxis es la siguiente:</p> <p>Clear</p>	<pre data-bbox="828 785 1252 821">[fp03alu36@samba ~]\$ clear</pre>

(comando cp)	
<p>El comando cp permite copiar un archivo, su sintaxis es la siguiente:</p> <p>cp archivo_origen archivo_destino</p>	<pre>[fp03alu36@samba ~]\$ cp archivo_origen archivo_destino cp: se omite el directorio «archivo_origen»</pre>
<p>Para obtener una copia del archivo datos.txt con nombre datosViejos.txt en el mismo directorio, entonces se escribe el comando</p> <p>cp datos.txt datosViejos.txt</p>	<pre>[+p03alu36@samba ~]\$ cp datos.txt datosViejos.txt cp: se omite el directorio «datos.txt»</pre>
<p>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:</p> <p>cp ../archivo_a_copiar .</p> <p>Es muy importante indicar como archivo destino al punto (.) para que el archivo de copia se ubique en el directorio actual.</p>	<pre>[fp03alu36@samba nombre_carpeta]\$ cp ../archivo_a_copiar .</pre>

(comando mv)	
<p>El comando mv mueve un archivo de un lugar a otro, en el sistema de archivos; su sintaxis es la siguiente:</p> <pre>mv ubicación_origen/archivo ubicación_destino</pre>	<pre>[fp03alu36@samba nombre_carpeta]\$ mv ubicación_origen/archivo ubicación_destino</pre>
<p>El comando mueve el archivo desde su ubicación origen hacia la ubicación 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:</p> <pre>mv ../archivo_a_reubicar .</pre>	<pre>[fp03alu36@samba nombre_carpeta]\$ mv ../archivo_a_reubicar .</pre>
<p>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:</p> <pre>mv nombre_actual_archivo nombre_nuevo_archivo</pre>	<pre>[fp03alu36@samba ~]\$ mv nombre_actual_archivo nombre_nuevo_archivo</pre>

(comando rm)	
<p>El comando rm permite eliminar un archivo o un directorio, su sintaxis es la siguiente:</p> <pre>rm nombre_archivo rm nombre_carpeta</pre> <p>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.</p>	<pre>[fp03alu36@samba ~]\$ rm nombre_archivo [fp03alu36@samba ~]\$ rm nombre_carpeta</pre>

Conclusión

Gracias a esta práctica empezamos a tener el conocimiento de cómo funcionan y que son los sistemas operativos que usamos todos los días, aunque apenas estemos empezando con los comandos más básicos serán de gran ayuda cuando queramos empezar nuestros propios proyectos.