

Laboratorio de Sistemas Operativos
Semestre A-2018
Práctica Uno
Poslaboratorio

Prof. Rodolfo Sumoza
Prep. Alvaro Araujo
Prep. Luis Sanchez

1. Comando `ls`

El comando `ls` es uno de los mas utilizados en la consola nativa de los sistemas LINUX, este permite listar los elementos (archivos o directorios) contenidos en un directorio determinado, si no se especifica una ruta como argumento, el comando lista los elementos del directorio actual '.', si se quiere listar cualquier otro directorio se debe especificar como argumento su ruta(absolute o relativa). El comando `ls` acepta opciones que son de gran utilidad, las mas usadas son:

- **-a** Incluye en el listado los elementos los cuales su nombre empieza con el caracter '.'.
- **-l** Lista los elementos con más detalle, se especifica para cada uno de ellos sus permisos, el nombre del propietario, el grupo al que pertenece, el tamaño en bytes, si se trata de un directorio o un archivo y la fecha de la última modificación.
- **-R** Lista los contenidos de todos los directorios recursivamente.

para mas información acerca de este comando y sus opciones se puede consultar el manual del sistema (General Commands Manual GNU).

2. Nueva versión de ls.

En este postlaboratorio se implementará una versión resumida del comando **ls**, básicamente la nueva versión tendrá la misma funcionalidad que posee **ls**, solo que esta versión tendrá solo tres parámetros (**-d -f -h**) y se listarán todos los ficheros o directorios, incluidos los que empiecen con **.**. Normalmente esto es posible con el parámetro (**-a**). A continuación se describen los parámetros que tendrá esta nueva versión:

- **-d** La salida debe contener una lista de los directorios contenidos en la ruta especificada.
- **-f** La salida debe contener una lista de los archivos contenidos en la ruta especificada.
- **-h** Se debe desplegar la ayuda del programa (Ya se encuentra implementada).

Al igual que el comando **ls**, la nueva versión debe tener la posibilidad de que el usuario especifique la ruta del directorio del cual desea listar su contenido y en caso de que no lo haga se debe tomar la ruta actual. A continuación se muestran algunas salidas que debería tener el programa para ciertas ejecuciones:

```
usuario@pc:/$ ./ls /opt
```

```
Directorio --> google
Archivo --> aleph-1.6f.tbz
Directorio --> argouml-0.34
Directorio --> Linux
Directorio --> ..
Directorio --> eclipse
Directorio --> spotify
Directorio --> sublime_text
Directorio --> .
Archivo --> reaver-1.4.tar.gz
Directorio --> reaver-1.4
Directorio --> aleph
```

```
usuario@pc:/$ ./ls -f /opt
```

```
Archivo --> aleph-1.6f.tbz
Archivo --> reaver-1.4.tar.gz
```

```
usuario@pc:/$ ./ls -d /opt
```

```

Directorio --> google
Directorio --> argouml-0.34
Directorio --> Linux
Directorio --> ..
Directorio --> eclipse
Directorio --> spotify
Directorio --> sublime_text
Directorio --> .
Directorio --> reaver-1.4
Directorio --> aleph

```

Es de esperarse que si se agregan los parámetros **-f** y **-d** simultáneamente, se obtiene la misma salida que si no se colocara ningún parámetro. Para usar ambos parámetros se pueden colocar separados (**-f -d**) o juntos (**-fd**). La salida será la misma.

3. Implementación de la nueva versión de ls.

Junto con esta guía se proporciona un archivo de código fuente *ls.c*, en este archivo ya hay parte implementada de la nueva versión, ya está terminada la parte de cómo se manejan los parámetros y ya se encuentra lista la opción de ayuda (**-h**), para el manejo de los parámetros se usó **getopt**¹, aunque no es necesario saber de **getopt** para completar el programa, se recomienda consultar su documentación ya que es una herramienta muy útil.

Para desarrollar el segmento que lista los directorios y archivos no se puede utilizar una llamada al ya existente programa **ls**, se pueden utilizar las funciones **opendir**, **readdir**, **closedir**, si no posee información sobre estas funciones utilice la documentación, estas funciones se encuentran en la biblioteca **dirent.h**, la lectura de los directorios es muy similar a la de los archivos, primero se debe abrir el directorio, luego leer su contenido y por último cerrar el directorio. La lectura se puede lograr de la siguiente manera:

```
while (estDirectorio = readdir(ptoDirectorio))
```

Investigue acerca de qué tipo de dato devuelve la función **readdir**, es probable que de ahí se pueda obtener el tipo del elemento, si es un archivo o un directorio y también su nombre.

4. Llamadas al Sistema.

Luego de finalizado el programa, ejecútelo y observe qué llamadas al sistema utiliza éste, luego compárelo con las llamadas obtenidas en el primer ejercicio del Laboratorio de la Práctica Uno, liste las nuevas llamadas e investigue qué funciones las generaron.

¹Linux Programmer's Manual -><http://man7.org/linux/man-pages/man3/getopt.3.html>