Universidad Simón Bolívar
Depto. de Computación y T.I.
Sistemas de Operación (CI-3825)
Abril-Julio 2.022

**Proyecto 1 (14%)**

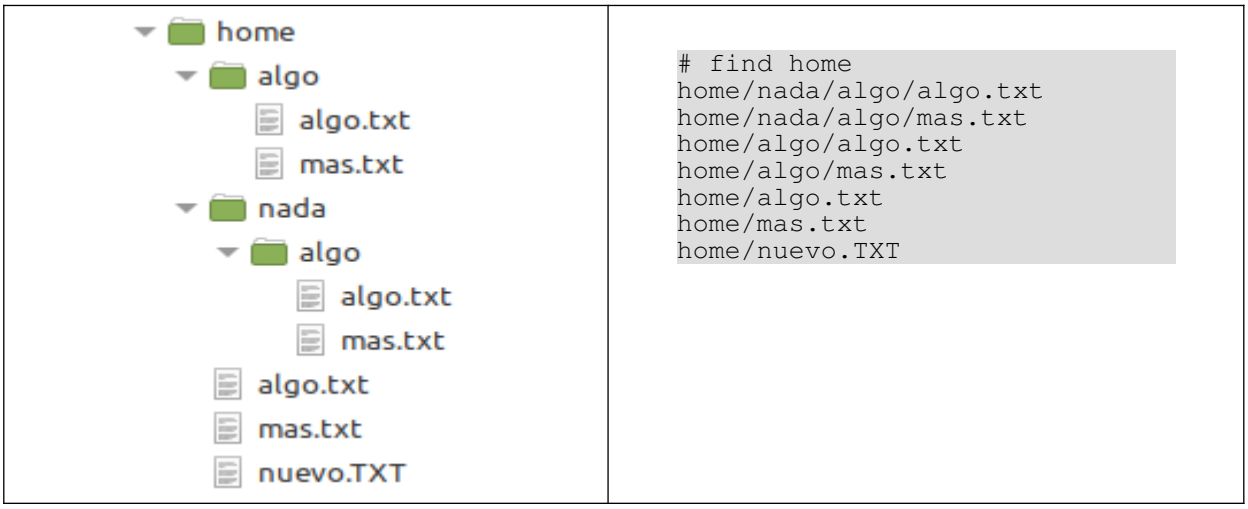
## Objetivo

Familiarizarse con las llamadas al sistema para el acceso a las estructuras del sistema de archivos.

## Introducción al Problema

Linux ofrece diversas herramientas para procesar el contenido de un árbol de directorios. Por ejemplo, *find*, *cp*, *ls*[1] entre otros. Estas herramientas se ofrecen como comandos internos o comandos externos de los interpretadores de comandos. Los internos se implementan como funciones dentro del mismo interpretador, y los externos son ejecutables que pueden ser invocados por el interpretador. Desde el punto de vista del usuario, si el comando es interno o no, no es relevante, a él sólo le interesa el resultado de la orden. Todos estos elementos en conjunto permiten ofrecer a los usuarios un ambiente para realizar tanto tareas simples como complejas.

Considere la figura siguiente que muestra una estructura de directorios del sistema Unix y a un interpretador de comando recibiendo la orden `find home` en la raíz de este árbol. Para su procesamiento, el interpretador toma la orden, identifica que se trata de un comando externo, ubica el ejecutable, lo invoca y éste imprime la salida por el terminal que comparten.




En particular, este comando muestra por el terminal los archivos en la estructura de directorio

1 Estos últimos con la opción -R

que tiene como raíz `home`, es decir, recorre en forma recursiva la estructura hasta llegar a las hojas y por cada directorio imprime su contenido. Los nombres de los archivos se escriben en forma relativa a la raíz del árbol.

**Descripción general del problema**

Para este proyecto Ud. debe implementar un interpretador de comandos simple que ofrezca un conjunto de comandos internos. En otras palabras, los comando implementados no tienen ejecutables asociados a ellos en el sistema de archivos, se implementan como funciones del interpretador.

Este interpretador deberá invocarse de la siguiente forma:

# ./myutil <directorioRaiz>

Los comandos que debe reconocer son los siguientes:

1. ***find [<cadena>]***: imprime los nombres de los archivos que tienen en su nombre la cadena <*cadena*>. Este comando es *case sensitive*, es decir, distingue entre mayúsculas y minúsculas. Use el siguiente formato para la salida.

```
myutil> find txt
home/nada/algo/algo.txt
home/algo/algo.txt
home/algo.txt
myutil>
```

   En caso de no especificarse <*cadena*>, se debe imprimir todos los archivos.

2. ***ifind [<cadena>]***: igual al anterior pero no es *case sensitive*. Use el siguiente formato para la salida.

```
myutil> ifind txt
home/nada/algo/algo.txt
home/algo/algo.txt
home/algo.txt
home/nuevo.TXT
myutil>
```

3. ***cfind <cadena1> <cadena2>***: igual a find pero adicionalmente verifica que el *contenido del archivo tenga la cadena* <*cadena2*>. Use el siguiente formato para la salida.

```
myutil> cfind txt hola
home/nada/algo/algo.txt
```

```
myutil>
```

4. ***repla <file>***: toma un archivo con pares ***<file>***[2] y aplica las sustituciones en los archivos regulares que se encuentren en el árbol. El archivo ***<file>*** no estará dentro del árbol de archivos de myutil.

   Este comando no imprime nada.

```
myutil> repla palabras.txt
myutil>
```

5. ***wc***: por directorio dentro del árbol, imprime el total de líneas y caracteres de los archivos regulares que contiene en forma recursiva, incluyendo la raíz del árbol. Use el siguiente formato para la salida.

```
myutil> wc
…
El directorio home/nada/algo: 25 líneas y 200 caracteres
El directorio home/nada: 50 líneas y 2000 caracteres
…
El directorio home: 400 líneas y 50000 caracteres
myutil>
```

6. ***codif***: Realiza un proceso de codificación básico: por cada archivo regular, invierte su contenido, es decir, hará intercambio del primer y último carácter, del segundo y penúltimo, y así hasta que todos los caracteres hayan sido intercambiados. Note que si el archivo tiene número impar de caracteres, el carácter del centro permanecerá en la misma posición.

   Este comando no imprime nada.

```
myutil> codif
myutil>
```

   *No debe crear archivos auxiliares.*

7. ***roll [<n>]***: realiza otro proceso de codificación: rota en <n> caracteres el contenido de cada archivo regular, es decir,
   - si <n> es positivo, desplaza <n> caracteres a la derecha, los últimos <n> caracteres deben aparecer al comienzo del archivo.
   - si <n> es negativo, desplaza -<n> caracteres a la izquierda, los primeros -<n>

2 Ver enunciado tarea.

deben aparecer al final del archivo.

Puede asumir que los archivos tendrán más de |<n>| caracteres.

Este comando no imprime nada.

```
myutil> roll 5
myutil> roll -5[3]
myutil>
```

*No debe crear archivos auxiliares.*

**Note**

1. [ x ] indica que x puede estar presente o no en la línea de comando.
2. El contenido de un archivo no puede estar en memoria RAM al mismo tiempo.
3. Al ejecutar los comandos repla, codif, y roll, cada bloque de datos será procesado exactamente una vez.

## Recomendaciones

Entender bien el problema antes de diseñar su propuesta.

- Diseñe toda su solución antes de comenzar a programar.
- Lea la información dada por las páginas de manual del sistema sobre las funciones que debe usar.
- Vaya documentado su código a medida que lo vaya generando.
- Trabaje en forma ordenada!
- Si tiene alguna duda, con tiempo aclárela directamente con su profesor.

## Entrega de la tarea:

**Realización: Individual**

**Digital:** Hasta las 11:59pm del Viernes de semana 7.

Cada estudiante debe colocar en un archivo que con nombre carnet.tar.gz:

- los fuentes de su tarea (archivos .c y .h). Estos deben estar bien identificados, documentados y estructurados. Este conjunto será usado para compilar y correr su tarea.
- Un archivo de nombre *codigo.pdf* con los fuentes de su tarea. Exactamente la misma versión que los .c y .h. Éste será usado para evaluar el código de su tarea.
- El Makefile que pueda compilar/enlazar su aplicación

Note que debe estar subscrito al espacio canvas para poder optar a esta opción, ***no espere al día de la entrega para notificar que tiene problemas o que no se ha registrado***.

3 Note que este comando revierte el efecto del anterior