

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Sistemas Operativos

(Micro) Sistema de Archivos

Alumnos:

- Ortega González Luis Abraham
- Páez Lampón Bidkar Herson

Profesor: Ing. Wolf Iszaevich Gunnar Eyal

Semestre 2017-1

# **Índice**

- 1. Presentación del Documento**
- 2. Problema a Resolver**
- 3. Lógica de Programación**
- 4. Bibliotecas**
  - 4.1 Necesarias para Desarrollo**
  - 4.2 Necesarias para Ejecución**
- 5. Ejemplos de Uso e Invocación**

## **1. Presentación del Documento**

El objetivo del presente documento es el de documentar el software “(Micro) Sistema de Archivos”, desarrollado por los alumnos Ortega González Luis Abraham y Páez Lampón Bidkar Herson. En este documento se explica el problema que se busca resolver, la lógica de operación y/o programación, las librerías necesarias para su desarrollo y ejecución, y se muestran ejemplos de uso e invocación del mismo en un Sistema Operativo Ubuntu 14.04.5 LTS.

## **2. Problema a Resolver**

Desarrollar un software “(Micro) Sistema de Archivos” que se encargue de gestionar los archivos contenidos a su vez en otro archivo, el cual será tomado como raíz, permitiendo al usuario realizar operaciones con estos como listar directorio, crear o eliminar archivos así también como la lectura, la escritura y append.

## **3. Lógica de Programación**

Programa

Mientras el volumen de almacenamiento exista y el comando introducido no sea 'exit' el programa esperará instrucciones por parte del usuario de manera indefinida.

Función listar

Recibe apuntador a la raíz y una ruta relativa en forma de cadena. Si la ruta se refiere a un archivo imprime su nombre, si se refiere a un directorio y no está vacío se itera sobre los archivos contenidos e imprime sus nombres llamándose a sí misma.

Función check

Recibe nada. Comprueba la existencia del archivo que fungirá como volumen de almacenamiento, si no existe el programa no se ejecuta.

Función nuarch

Recibe apuntador a la raíz y el nombre del archivo en forma de cadena.  
Permite crear un nuevo archivo vacío, cuyo nombre será el indicado por el usuario.

#### Función deletes

Recibe apuntador a la raíz y el nombre del archivo en forma de cadena.  
Permite eliminar un archivo indicado por el usuario mediante su nombre.

#### Función readf

Recibe apuntador a la raíz y el nombre del archivo en forma de cadena.  
Permite leer un archivo indicado por el usuario mediante su nombre.

#### Función writeto

Recibe apuntador a la raíz y el nombre del archivo en forma de cadena.  
Permite escribir sobre un archivo, indicado por el usuario mediante su nombre, sustituyendo su contenido.

#### Función addto

Recibe apuntador a la raíz y el nombre del archivo en forma de cadena.  
Permite escribir sobre un archivo, indicado por el usuario mediante su nombre, añadiendo lo indicado al final del documento respetando el contenido previo.

#### Función validate

Recibe el comando escrito por el usuario en forma de cadena y la lista de comandos disponibles en forma de arreglo. Valida que el comando introducido por el usuario sea válido comprobando que este exista en la lista de comandos.

#### Función clear

Recibe nada. Limpia pantalla.

#### Función task

Recibe el comando escrito por el usuario y el nombre del archivo sobre el que se desea operar en forma de cadena y un apuntador a la raíz.

## **4. Bibliotecas**

Ruby 2.2 – Lenguaje de programación utilizado para el desarrollo.

### **4.1 Necesarias para Desarrollo**

Zip – Biblioteca necesaria para el manejo de archivos zip

### **4.2 Necesarias para Ejecución**

Es de fundamental importancia que al momento de ejecutar el software las bibliotecas y el lenguaje de programación se encuentren en la versión especificada, pues la variación de la misma podría causar un mal funcionamiento.

- Ruby 2.2
- Zip

## **5. Ejemplos de Uso e Invocación**

La primer imagen corresponde a la invocación desde línea de comandos, las 3 restantes al uso del software trabajando con un archivo llamado “nuevo.txt”.

# En Ubuntu

```
bidkar@bidkar-HP-14-Notebook-PC: ~/Documentos/ruby
bidkar@bidkar-HP-14-Notebook-PC:~/Documentos/ruby$ ruby pr.rb
```

```
bidkar@bidkar-HP-14-Notebook-PC: ~/Documentos/ruby
Porfavor ingresa ayuda para ver los comandos si es tu primera vez en este sitio místico
[deusex@machine]$
```

```
bidkar@bidkar-HP-14-Notebook-PC: ~/Documentos/ruby
Porfavor ingresa ayuda para ver los comandos si es tu primera vez en este sitio místico
[deusex@machine]$ayuda
Este es el menu de ayuda

Lista de comandos

ld: Lista los archivos del sistema
clr: limpia la consola
exit:cierra el programa
newfile [nombrearchivo]: crea un archivo con el nombre que sustituyas en [nombrearchivo]
writeto [nombrearchivo]: Escribe a un archivo con el nombre que sustituyas en [nombrearchivo], este tambien crea un archivo pero con contenido
delete [nombrearchivo]: Elimina el archivo del nombre que sustituyas en [nombrearchivo]
addto [nombrearchivo]: Muestra el contenido del archivo que sustituyas en [nombrearchivo], y permite que le agregues mas contenido
readf [nombrearchivo]: Muestra el contenido del archivo que sustituyas en [nombrearchivo]

[deusex@machine]$ld
/readme
/hola.txt
/ayk
/morena
[deusex@machine]$newfile nuevo.txt
[deusex@machine]$ld
/readme
/hola.txt
/ayk
/morena
/nuevo.txt
[deusex@machine]$readf nuevo.txt

[deusex@machine]$writeto nuevo.txt
Introduce el contenido para escribir en el archivo
Este archivo es nuevo
[deusex@machine]$readf nuevo.txt
Este archivo es nuevo
[deusex@machine]$
```

```
bidkar@bidkar-HP-14-Notebook-PC: ~/Documentos/ruby

[deusex@machine]$addto nuevo.txt
El contenido es :
Este archivo es nuevo
Introduce el contenido para anadir en el archivo
Agrego informacion extra.
[deusex@machine]$readf nuevo.txt
Este archivo es nuevo Agrego informacion extra.
[deusex@machine]$ld nuevo.txt
/readme
/hola.txt
/ayk
/morena
/nuevo.txt
[deusex@machine]$delete nuevo.txt
[deusex@machine]$ld
/readme
/hola.txt
/ayk
/morena
[deusex@machine]$exit
bidkar@bidkar-HP-14-Notebook-PC: ~/Documentos/ruby$
```