



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2333 - SISTEMAS OPERATIVOS Y REDES

Documentación Proyecto Sistemas Operativos

October 7, 2019

2º semestre 2019 - Profesor Cristián Ruz

Grupo Maniha

1 Funciones Generales

Funciones definidas como generales en el enunciado y se encuentran en "cr_API.c".

1.1 cr_mount(char* diskname)

Función que recibe la ruta local hacia el disco que se quiere utilizar.

Se levantan dos errores y se finaliza la ejecución del programa en caso de que la ruta entregada no sea correcta o que el archivo objetivo no sea un archivo con extension binaria (".bin") como espera.

Retorna el puntero a la posición local del archivo ".bin" (nuestro disco virtual) a una variable global definida en "cr_API.h" llamada "DISK_PATH".

1.2 cr_exist(char* path)

Función que recibe el ruta absoluta desde el directorio root.

Utiliza la función *recorrer_path* para saber si existe un directorio o archivo en la ruta especificada en la ruta entregada.

Retorna 1 si existe y 0 en caso contrario

1.3 `cr_ls(char* path)`

Función que recibe el ruta absoluta desde el directorio root.

Si el objetivo es un directorio, lista en consola todos los archivos y subdirectorios que estan dentro de el. Si es un archivo, retorna un mensaje diciendo que el objetivo es un archivo. Si la ruta esta mala, retorna un error en consola.

Esta función es de tipo void, por lo que no retorna nada.

2 Funciones de Manejo de Archivos

Funciones definidas como de manejo de archivos en el enunciado y se encuentran en "cr_API.c".

3 Funciones Utiles

Funciones definidas en "Util.c".

3.1 `encontra_directorio(char* path, int posición)`

Función que recibe el nombre del archivo o directorio a buscar y la posición del bloque de directorio en el que se realizará la búsqueda.

Dada la posición entregada, es decir, el directorio dentro del cual se quiere buscar, se compara cada entrada del directorio con el nombre del archivo, para saber si existe dicho archivo o directorio dentro del directorio en el que se quiere buscar.

Retorna un puntero al directorio o archivo encontrado y NULL en caso de que no exista dicho archivo o directorio.

3.2 `recorrer_path(char* path`

Recibe el ruta absoluta desde root a un archivo o directorio especifico.

Al ser un ruta absoluta se aprovecha la existencia del caracter "/" como separador y se itera para avanzar por los directorios utilizando los nombres de estos. Una vez encontrado cada nombre de subdirectorio o archivo, se utiliza la función *encontrar_directorio* para saber si existe dentro del directorio en el que nos encontramos. En caso de ser un directorio y no ser el objetivo, se procede a guardar su posición para poder comenzar a buscar dentro de este en la proxima iteración.

Finalmente retorna un puntero al directorio o archivo final.

3.3 **directorio_a_agregar(char* path)**

Recibe una ruta absoluta desde root a un archivo o directorio.

Utiliza la estructura de un ruta absoluta para crear un arreglo con la ruta hasta el directorio en el que se encuentra contenido el archivo o directorio objetivo.

Retorna un la ruta hasta el directorio que contiene al objetivo.

3.4 **objective_kind(char*path)**

Recibe el ruta absoluta desde root a un archivo o directorio.

Utiliza el bit de validez del directorio o archivo objetivo para saber si es un archivo o un directorio.

Finalmente retorna un 1 si el objetivo es directorio, un 0 si es archivo y un 23 en caso de que no exista la ruta dada.

3.5 **print_ls(char* path)**

Recibe el ruta absoluta desde root a un archivo o directorio especifico.

Permite avanzar por los directorios hasta encontrar el objetivo y utiliza la función *encontrar_directorio* para obtener la posición del directorio dado, una vez que tiene la posicion donde se encuentra el archivo objetivo, entrega dicha posición a *print_all*.

Esta función es de tipo void, por lo que no retorna nada.

3.6 **print_all(int posicion)**

Recibe la posición de un archivo o directorio especifico.

A partir de la posición entregada, retorna todos los subdirectorios y archivos que se encuentran contenidos en el directorio especifico.

Esta función es de tipo void, por lo que no retorna nada.

3.7 isBin(char* path)

Recibe la ruta a un archivo o directorio específico.

Se preocupa de ver si el archivo objetivo de la ruta es de extensión ".bin", compara las ultimas 4 letras de la ruta para saber esto.

Retorna un char* con un string de los ultimos 4 caracteres de la ruta.