



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

# Documentación Proyecto Sistemas Operativos

October 6, 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 el nombre del disco.

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 un path absoluto desde el directorio root.

Utiliza la función *recorrer\_path* para saber si existe un directorio o archivo en la ruta especificada en el path entregado.

Retorna 1 si existe y 0 en caso contrario

## 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 path absoluto desde root a un archivo o directorio específico.

Al ser un path absoluto 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 próxima iteración.

Finalmente retorna un puntero al directorio o archivo final.

### 3.3 `objective_kind(char*path)`

Recibe el path absoluto desde root a un archivo o directorio.

Revisa el path entregado para saber si el objetivo es un directorio o un archivo, utilizando la convención de que todo archivo esta marcado por una extensión luego de un ".", por lo que se compara cada caracter de fina a inicio, si se encuentra un "." antes de el primer "/", sabemos que es un archivo, sino un directorio.

Finalmente retorna un 1 si el objetivo es directorio y un 0 si es archivo.