

MakeFile:

پیاده سازی دایرکتوری های پروژه در این فایل انجام شده است و با دستور make و clean پروژه ساخته و پاک میشود. و دایرکتوری / در کنار make file ساخته میشود و همچنین دایرکتوری های bin/dev/etc/home/mnt/proc/tmp/usr همراه فایل متادیتاشون در این فایل قرار دارند.

Configfile:

دستورات در فایل کانفیگ با ", " از یکدیگر جدا شده اند و دستور مربوط به CRUD آن پروسس ها در هر خط وارد شده است.

Metadata:

متادیتای نه دایرکتوری در این فایل اضافه شده است که شامل path, total size, free size, number of files, and permissions میباشد.

GetFolderSize(const char\* path):

این تابع اندازه یک دایرکتوری و زیر شاخه های آن را به صورت بازگشتی محاسبه می کند. از تابع stat برای دریافت اطلاعات فایل استفاده می کند و اندازه همه فایل های معمولی را در فهرست اضافه می کند.

void make\_directory(const char\* path):

این تابع دایرکتوری ها را به صورت بازگشتی برای یک مسیر مشخص ایجاد می کند.

countFileRecurisvly(const char\* path):

این تابع تعداد فایل های یک دایرکتوری و زیرشاخه های آن را به صورت بازگشتی می شمارد. از توابع stat و readdir برای عبور از ساختار دایرکتوری و شمارش فایل ها استفاده می کند.

Update\_meta():

این تابع متادیتا (اندازه کل، اندازه آزاد و تعداد فایل ها) را برای هر دایرکتوری در آرایه متا به روز می کند. از توابع getFolderSize و countFilesRecurisvly برای محاسبه مقادیر استفاده می کند و متادیتای به روز شده را در یک فایل meta می نویسد.

create\_file(char\* path, char\* fileName, char\* write\_string):

با گرفتن مسیر دایرکتوری و اسم فایل به استرینگ در آن فایل ذخیره میشود

read\_file(char\* path, char\* fileName):

با خواندن مسیر و اسم فایل ، فایل کت شده به ما نمایش داده میشود

update\_file(char\* path, char\* fileName, char\* append string):

با خواندن مسیر و اسم فایل append را انجام میدهد. و تنها تفاوتش با create این است که در متود create رایت انجام میشود اما در اینجا append.

delete\_file(char\* path, char\* fileName):

با خواندن مسیر فایل آن را پاک میکند.

havepermission(char\* ptype, char\* path, int read , int write)

در این متود با گرفتن ptype موردنظر و همچنین مسیر امکان دسترسی داشتن را بررسی میکنیم.