

پروژه سیستم عامل

این پروژه هدف شبیه‌سازی یک سیستم فایل ساده با ویژگی‌های مختلف است که به شما امکان مدیریت فایل‌ها، دایرکتوری‌ها و عملیات‌های مختلف بر روی دیسک مجازی را می‌دهد. در این سیستم فایل، داده‌ها به صورت بلوک‌هایی ذخیره می‌شوند و یک جدول تخصیص فایل (FAT) برای ردیابی بلوک‌های تخصیص یافته به هر فایل مورد استفاده قرار می‌گیرد.

**ساختار کلی کد:**

**تعریف دیسک مجازی و تنظیمات اولیه**:

برای شروع، باید برخی تنظیمات اولیه مانند اندازه بلوک‌ها، تعداد بلوک‌های دیسک، و اندازه جدول تخصیص فایل (FAT) تعریف شود.

**DISK\_FILE** مسیر فایل دیسک مجازی را مشخص می‌کند.

**BLOCK\_SIZE**  اندازه هر بلوک داده در دیسک را تعیین می‌کند. در اینجا، فرض شده که اندازه هر بلوک ۴۰۹۶ بایت است.

**NUM\_BLOCKS** تعداد بلوک‌های دیسک را مشخص می‌کند که برای مدیریت فضای ذخیره‌سازی استفاده می‌شود.

**FAT\_TABLE\_SIZE**  تعداد ورودی‌های جدول تخصیص فایل (FAT) است که معادل تعداد بلوک‌های دیسک است.

**کلاس FileSystem**

این کلاس مرکزی است که تمامی عملیات‌های مربوط به سیستم فایل را انجام می‌دهد.

**ویژگی‌ها و توابع کد:**

1. **فرمت کردن دیسک (format\_disk)**

* این تابع دیسک مجازی را فرمت می‌کند و تمامی بلوک‌ها را به مقدار صفر بازنشانی می‌کند. این به این معنا است که تمام داده‌های موجود پاک می‌شوند.
* جدول تخصیص فایل (FAT) نیز برای بازگشت به وضعیت اولیه پاک می‌شود.
* این تابع وقتی که دیسک به‌طور کامل از بین رفته باشد یا دیسک موجود قابل استفاده نباشد، فراخوانی می‌شود.

1. **بارگذاری دیسک (load\_disk)**

* در صورتی که دیسک موجود باشد، این تابع دیسک را بارگذاری کرده و آن را به حالت آماده برای استفاده درمی‌آورد.
* اگر دیسک وجود نداشته باشد، آن را به طور خودکار فرمت کرده و یک دیسک جدید ایجاد می‌کند.

1. **تخصیص بلوک (allocate\_block)**

* این تابع برای تخصیص یک بلوک جدید از دیسک به یک فایل استفاده می‌شود.
* در جدول تخصیص فایل (FAT)، بلوک مورد نظر به عنوان بلوک تخصیص یافته علامت‌گذاری می‌شود (یعنی مقدار -1 در FAT برای آن بلوک قرار می‌گیرد).

1. **آزادسازی بلوک (free\_block)**

* پس از اتمام استفاده از یک بلوک، این تابع برای آزادسازی آن بلوک در جدول FAT استفاده می‌شود.
* با آزادسازی بلوک، آن بلوک مجدداً در دسترس قرار می‌گیرد تا برای ذخیره‌سازی فایل‌های جدید استفاده شود.

1. **نوشتن در بلوک (write\_block)**

* این تابع داده‌ها را در یک بلوک خاص دیسک ذخیره می‌کند.
* این داده‌ها می‌توانند شامل اطلاعات مختلف از جمله محتوای فایل‌ها باشند.

1. **خواندن از بلوک (read\_block)**

* این تابع داده‌ها را از یک بلوک خاص دیسک می‌خواند.
* برای دسترسی به محتوای فایل‌ها و نمایش آن‌ها در سیستم مورد استفاده قرار می‌گیرد.

1. **ایجاد فایل (create\_file)**

* این تابع به شما اجازه می‌دهد تا یک فایل جدید بسازید و محتوای آن را در دیسک ذخیره کنید.
* فایل‌های ایجاد شده در دایرکتوری ریشه سیستم فایل قرار می‌گیرند.
* در اینجا، هر فایل به یک یا چند بلوک از دیسک تخصیص داده می‌شود.

1. **حذف فایل (delete\_file)**

* این تابع یک فایل را از سیستم فایل حذف می‌کند.
* پس از حذف فایل، بلوک‌های تخصیص داده شده به آن فایل در جدول FAT آزاد می‌شوند.

1. **تغییر نام فایل (rename\_file)**

* این تابع به شما این امکان را می‌دهد که نام یک فایل را تغییر دهید. توجه داشته باشید که تغییر نام فقط در دایرکتوری ریشه سیستم فایل انجام می‌شود.

1. **خواندن فایل (read\_file)**

* این تابع محتویات یک فایل را از دیسک خوانده و نمایش می‌دهد.

1. **اجرای فایل (run\_file)**

* این تابع به شما اجازه می‌دهد که محتوای یک فایل را به عنوان یک کد Python اجرا کنید.
* این می‌تواند برای سیستم‌هایی که نیاز به اجرای اسکریپت‌ها دارند مفید باشد.
* اگر فایل متنی باشد محتوای متن را نشان می‌دهد.

1. **نمایش لیست فایل‌ها (list\_files)**

* این تابع تمامی فایل‌های موجود در دایرکتوری ریشه را لیست می‌کند.

1. **کپی کردن فایل (copy\_file)**

* این تابع یک فایل را از یک مکان به مکان دیگر در سیستم فایل کپی می‌کند.

1. **جستجو برای فایل‌ها (find\_file)**

* این تابع به جستجو برای فایل‌هایی که شامل یک کلمه خاص در نام یا محتویات خود هستند، می‌پردازد.

1. **ایجاد دایرکتوری (mkdir)**

* این تابع یک دایرکتوری جدید در سیستم فایل ایجاد می‌کند.
* دایرکتوری‌ها به شما این امکان را می‌دهند که فایل‌ها را به صورت مرتب‌تری سازماندهی کنید.

1. **حذف دایرکتوری (rmdir)**

* این تابع برای حذف دایرکتوری‌های خالی استفاده می‌شود.

1. **فشرده‌سازی فایل (compress\_file)**

* این تابع از الگوریتم فشرده‌سازی zlib برای فشرده‌سازی محتوای یک فایل استفاده می‌کند.
* پس از فشرده‌سازی، فایل جدیدی با پسوند .zip ذخیره می‌شود.

1. **استخراج فایل (decompress\_file)**

* این تابع برای بازگردانی فایل‌های فشرده به حالت اولیه خود استفاده می‌شود.
* فایل فشرده شده را از دیسک می‌خواند و محتوای آن را استخراج می‌کند.

1. **زمان‌بندی اجرای فایل (schedule\_file)**

* این تابع به شما اجازه می‌دهد که یک فایل را با تأخیر مشخص اجرا کنید.
* زمان‌بندی می‌تواند مفید باشد در شرایطی که نیاز به اجرای فایل‌ها در زمان‌های خاص دارید.

1. **تغییر دایرکتوری (cd)**

* این تابع به شما امکان می‌دهد که دایرکتوری جاری را تغییر دهید.
* تغییر دایرکتوری در سیستم فایل برای جابجایی میان پوشه‌ها استفاده می‌شود.

1. **دریافت مسیر دایرکتوری جاری (pwd)**

* این تابع مسیر کامل دایرکتوری فعلی را نمایش می‌دهد.

1. **دریافت نام دایرکتوری جاری (cwd)**

* این تابع فقط نام دایرکتوری فعلی را نمایش می‌دهد.

1. **ثبت لاگ دستورات (log\_command)**

* این تابع تمامی دستورات ورودی را در یک فایل لاگ (log.txt) ذخیره می‌کند تا بتوانید تاریخچه دستورات اجرا شده را مشاهده کنید.

1. **نمایش راهنما (help\_menu)**

* این تابع تمامی دستورات و ویژگی‌های سیستم فایل را به همراه توضیحات کوتاه در مورد نحوه استفاده از آن‌ها نمایش می‌دهد.

1. **خاموش کردن سیستم (shutdown)**

* این تابع برای بستن دیسک مجازی و ذخیره تغییرات در آن استفاده می‌شود.

**نحوه استفاده از سیستم فایل:**

برای استفاده از این سیستم فایل، ابتدا باید دیسک را فرمت کرده و سپس از دستورات مختلف برای مدیریت فایل‌ها و دایرکتوری‌ها استفاده کنید. این سیستم به شما این امکان را می‌دهد که فایل‌ها را ایجاد، حذف، فشرده‌سازی، جستجو و اجرا کنید. علاوه بر این، قابلیت‌های زمان‌بندی، مدیریت دایرکتوری‌ها و ثبت دستورات را نیز داراست.

شبنم خاقانپور

4013623007