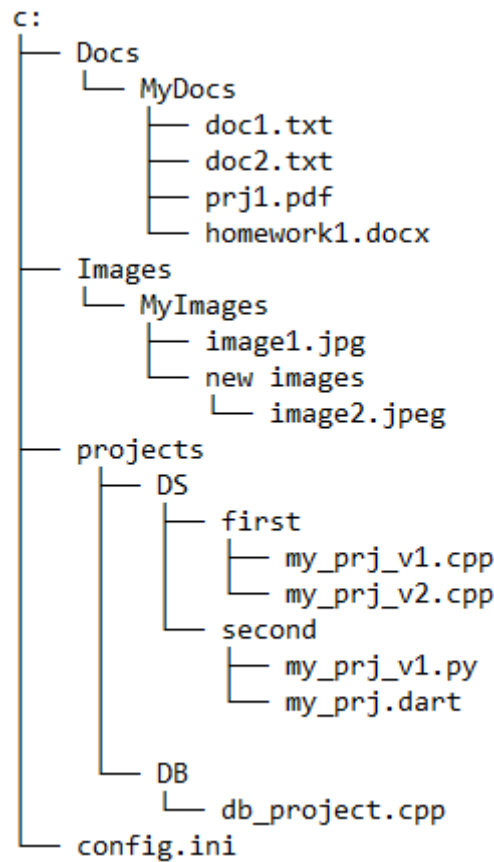


درخت دودویی برای پیاده سازی یک سیستم فایل

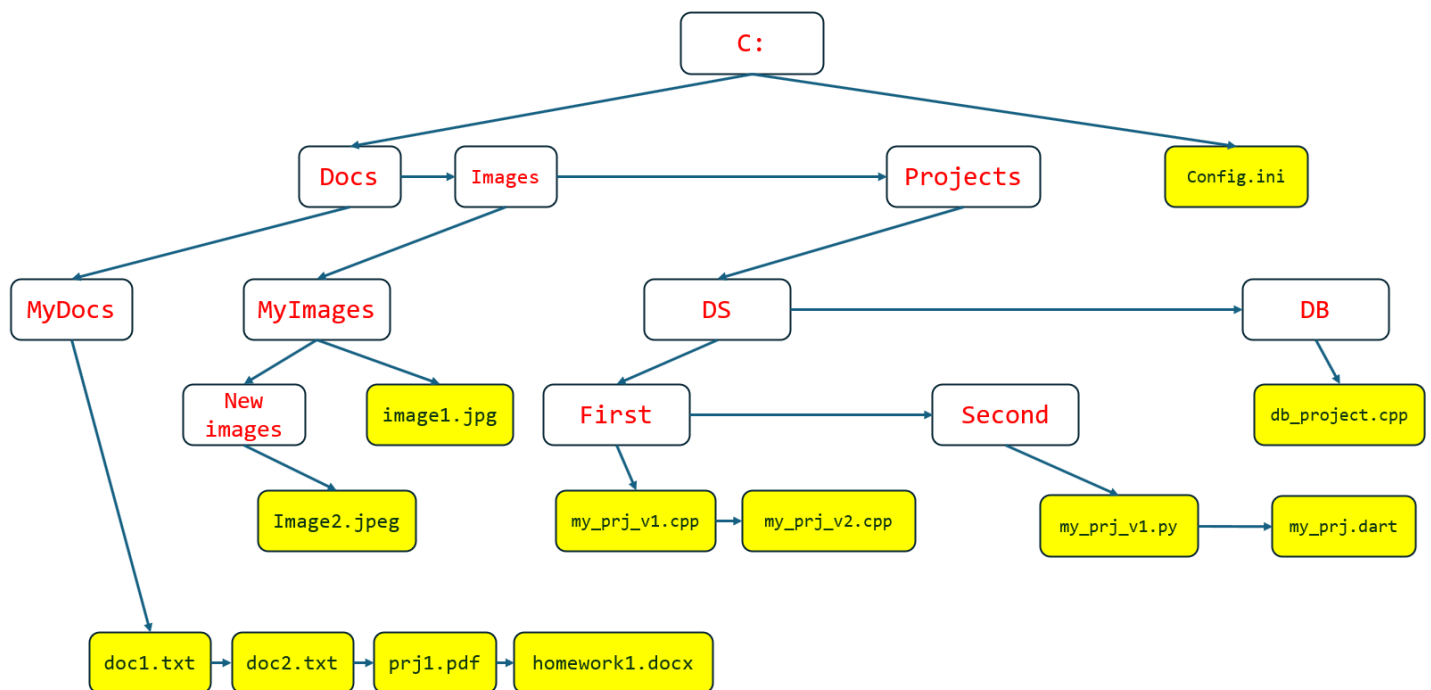
تعریف مسأله

در این پروژه سعی داریم یک سیستم فایل فرضی شبیه به سیستم فایل ویندوز را با استفاده از ساختار درخت دودویی پیاده سازی نماییم. همان گونه که می دانید، سیستم فایل در سیستم های عامل عمدتاً ساختار سلسله مراتبی دارد. به عنوان مثال در ویندوز، هر پارتیشن از تعداد دایرکتوری و فایل تشکیل شده که هر دایرکتوری نیز می تواند از چندین زیردایرکتوری و فایل دیگر تشکیل شده و این سلسله مراتب می تواند تا چندین سطح ادامه یابد. نمونه ای از این ساختار را در شکل زیر مشاهده می نمایید.



تصویر ۱- نمونه ای از ساختار سلسله مراتبی یک سیستم فایل نمونه

برای پیاده سازی سیستم فایلی شبیه توصیف فوق، هر پارتیشن از سیستم را به صورت یک درخت دودویی نمایش داده که ریشه ی درخت متناظر با *root directory* آن پارتیشن می باشد. همچنین هر گره در این درخت می تواند دو فرزند داشته باشد که فرزند چپ به لیست دایرکتوری های زیر آن دایرکتوری و فرزند راست به لیست فایل های زیر آن دایرکتوری اشاره دارد. بدیهی است که گره های متناظر با فایل ها فرزندی نخواهند داشت. ساختار درختی متناظر با شکل ۱ در تصویر ۲ نشان داده شده است. با توجه به این که سیستم فایل شامل مجموعه ای از پارتیشن ها می باشد، ساختمان داده ی نهایی شامل مجموعه ای از درخت هایی مشابه تصویر ۲ خواهد بود. (یک جنگل)



تصویر ۲ - نمایش درختی سلسله مراتب نمایش داده شده در تصویر ۱

ساختار پیشنهادی هر گره از لیست پیوندی

برای این پروژه می‌توان دو نوع گره با ساختارهای زیر پیشنهاد نمود:

الف) ساختار گره‌های متناظر با فایل‌ها شامل مؤلفه‌های زیر:

- ❖ مؤلفه‌ی name برای نگهداری نام فایل (شامل شامل نام فایل و extention آن)
- ❖ مؤلفه‌ی size برای نگهداری حجم فایل
- ❖ مؤلفه‌ی date برای نگهداری تاریخ ایجاد فایل
- ❖ مؤلفه‌ی attributes برای نگهداری ویژگی‌های فایل (فقط خواندنی، خواندنی/نوشتنی، مخفی و...) - توجه داشته باشید یک فایل می‌تواند چند مورد از صفات را همزمان داشته باشد.
- ❖ اشاره‌گر link برای اشاره فایل بعدی در دایرکتوری جاری

ب) ساختار گره‌های متناظر با دایرکتوری‌ها شامل مؤلفه‌های زیر:

- ❖ مؤلفه‌ی name برای نگهداری نام دایرکتوری
- ❖ مؤلفه‌ی date برای نگهداری تاریخ ایجاد دایرکتوری
- ❖ مؤلفه‌ی attributes برای نگهداری ویژگی‌های دایرکتوری (فقط خواندنی، خواندنی/نوشتنی، مخفی) - توجه داشته‌اید یک دایرکتوری می‌تواند چند مورد از صفات را همزمان داشته باشد.
- ❖ یک اشاره‌گر left برای اشاره به اولین زیردایرکتوری یک دایرکتوری
- ❖ اشاره‌گر right برای اشاره به اولین فایل یک دایرکتوری
- ❖ اشاره‌گر link برای اشاره دایرکتوری بعدی در دایرکتوری جاری

همچنین برای هر پارتیشن میزان حافظه تخصیص داده شده به آن پارتیشن و نیز میزان حافظه باقی‌مانده از هر پارتیشن باید نگهداری شود.

امکانات برنامه

برنامه باید به‌عنوان ورودی از یک فایل متنی، یک سیستم فایل (مشابه نمونه‌ی ارایه شده در بخش بعدی) را خوانده و امکانات زیر را در اختیار کاربر قرار دهد. برای این کار یک خط فرمان به کاربر نشان داده می‌شود که کاربر می‌تواند با دستورات زیر در سیستم فایل تغییرات انجام دهد.

در شروع کار خط فرمان در مسیر ریشه‌ی اولین پارتیشن قرار دارد.

دستورات مورد نظر عبارتند از:

۱- رفتن به یک مسیر تعیین شده با دستور CD. به عنوان مثال برای رفتن به دایرکتوری second در مثال بالا دستور زیر را اجرا می‌کنیم:

CD c:\projects\DS\second

۲- اضافه کردن یک فایل جدید در یک مسیر (مسیر جاری و یا مسیر تعیین شده در خط فرمان) با دستور Create File. به عنوان مثال برای ایجاد یک فایل جدید به نام new_prj.py در دایرکتوری second با حجم ۱۲۰ کیلوبایت، دستور زیر را اجرا می کنیم.

Create File C:\Projects\DS\Second\new_prj.py 120

چنانچه مسیر ایجاد فایل تعیین نگردد، فایل در مسیر جاری ایجاد می گردد.

CD C:\Projects\DS\Second
Create File new_prj.py 120

می توان همراه ایجاد فایل ویژگی فایل را نیز تعیین نمود. ویژگی دارای مقادیر r برای فقط خواندنی، rw برای خواندنی، نوشتنی و h برای فایل مخفی می باشد. ویژگی پیش فرض برای فایل ها rw می باشد. بدیهی است که یک فایل نمی تواند دارای ویژگی های r و rw با هم باشد. به عنوان مثال دستور زیر ویژگی فایل جدید ایجاد شده با حجم ۱۰۲۴ کیلوبایت را فقط خواندنی و مخفی در نظر می گیرید.

Create File C:\Docs\MyDocs\new_doc.txt 1024 r h

همچنین در هنگام ایجاد یک فایل زمان ایجاد فایل برابر با زمان جاری سیستم تعیین می گردد. بدیهی است هنگام ایجاد فایل باید بررسی کنیم که در مسیر انتخابی فایلی با نام و پسوند فایل جدید وجود نداشته باشد. چنانچه فایلی با همان نام و پسوند در مسیر انتخابی وجود داشته باشد، باید از کاربر برای جایگزینی یا رد درخواست پرسیده شود. همچنین هنگام ایجاد فایل باید میزان فضای باقی مانده در پارتیشن جاری و امکان ایجاد فایل با حجم تعیین شده بررسی گردد.

۳- اضافه کردن یک دایرکتوری جدید در یک مسیر با دستور Create Dir. به عنوان مثال برای ایجاد یک دایرکتوری جدید به نام Nature در دایرکتوری MyImages دستور زیر را اجرا می کنیم.

Create Dir C:\MyImages\Nature

تمام موارد ذکر شده در مورد دستور ایجاد فایل (به جز بررسی ظرفیت باقی مانده در پارتیشن جاری) در مورد دستور ایجاد دایرکتوری نیز صدق می کند.

۴- حذف کردن یک فایل یا دایرکتوری در یک مسیر (مسیر جاری و یا مسیر تعیین شده در خط فرمان) با دستور Delete. به عنوان مثال برای حذف فایل new_doc.txt اضافه شده به دایرکتوری MyDocs، دستور زیر را اجرا می کنیم.

Delete C:\Docs\MyDocs\new_doc.txt

۵- تغییر نام یک فایل و یا دایرکتوری با دستور Rename. دستور زیر نام دایرکتوری Nature را به Nature_Images تغییر می دهد.

Rename C:\MyImages\Nature Nature_Images

بدیهی است هنگام تغییر نام فایل (دایرکتوری) باید بررسی کنیم که در مسیر انتخابی فایل (دایرکتوری) با نام و پسوند فایل جدید وجود نداشته باشد. چنانچه فایلی (دایرکتوری) با همان نام و پسوند در مسیر انتخابی وجود داشته باشد، باید از کاربر برای جایگزینی یا رد درخواست پرسیده شود.

۶- تغییر ویژگی های یک دایرکتوری و یا تغییر ویژگی ها و حجم یک فایل موجود در یک مسیر (مسیر جاری و یا مسیر تعیین شده در خط فرمان) با استفاده از دستور Change. به عنوان مثال برای با دستورات زیر دایرکتوری DB فقط خواندنی شده و نیز حجم فایل config.ini به مقدار ۲۵۰ کیلوبایت تغییر یافته و فایل مخفی نیز می شود.

Change C:\Projects\DB r
Change C:\config.ini 250 h

بدیهی است که در صورت تغییر حجم فایل، بررسی ظرفیت باقی مانده از پارتیشن جاری باید انجام شود.

۷- کپی کردن یک فایل یا یک دایرکتوری موجود در یک مسیر (مسیر جاری و یا مسیر تعیین شده در خط فرمان) به مسیر جدید با استفاده از دستور Copy. دستور زیر فایل pri_1.pdf را از مسیر C:\Docs\MyDocs به مسیر C:\Projects\Ds\First کپی می نماید.

Copy C:\Docs\MyDocs\pri_1.pdf C:\Projects\Ds\First

کپی کردن دایرکتوری نیز به منزله ی کپی کردن دایرکتوری همراه با تمام زیردایرکتوری ها و فایل های موجود در آن می باشد.

بدیهی است هنگام کپی کردن فایل (دایرکتوری) باید بررسی کنیم که در مسیر انتخابی فایلی (دایرکتوری) با نام و پسوند فایل جدید وجود نداشته باشد. چنانچه فایلی با همان نام و پسوند در مسیر انتخابی وجود داشته باشد، باید از کاربر برای جایگزینی یا رد درخواست پرسیده شود. همچنین هنگام کپی باید میزان فضای باقی مانده در پارتیشن جاری و امکان کپی فایل (دایرکتوری) بررسی گردد.

۸- جابه جا کردن یک فایل یا یک دایرکتوری موجود در یک مسیر (مسیر جاری و یا مسیر تعیین شده در خط فرمان) به مسیر جدید با استفاده از دستور Move. دستور زیر فایل image_1.jpg را از مسیر C:\Images\MyImages به مسیر C:\Images\MyImages\New_Images جابه جا می نماید.

Move C:\Images\MyImages\image_1.jpg C:\Images\MyImages\New_Images

جابه جا کردن دایرکتوری نیز به منزله ی جابه جا کردن دایرکتوری همراه با تمام زیردایرکتوری ها و فایل های موجود در آن می باشد.

بدیهی است هنگام جابه‌جا کردن فایل(دایرکتوری) باید بررسی کنیم که در مسیر انتخابی فایلی (دایرکتوری) با نام و پسوند فایل جدید وجود نداشته باشد. چنانچه فایلی با همان نام و پسوند در مسیر انتخابی وجود داشته باشد، باید از کاربر برای جای‌گزینی یا رد درخواست پرسیده شود. همچنین هنگام کپی باید میزان فضای باقی‌مانده در پارتیشن جاری و امکان کپی فایل(دایرکتوری) بررسی گردد.

۹- لیست کردن تمام دایرکتوری‌ها و فایل‌های موجود در یک مسیر (مسیر جاری و یا مسیر تعیین شده در خط فرمان) با دستور Dir. دستور زیر تمام دایرکتوری‌ها و فایل‌های موجود در مسیر C:\Docs را در خروجی نمایش می‌دهد.

Dir C:\Docs

در نتیجه‌ی این فرمان ابتدا دایرکتوری‌ها و سپس فایل‌های موجود در مسیر نمایش داده خواهند شد.

۱۰- یافتن فایل‌ها و یا دایرکتوری‌های موجود در یک مسیر (مسیر جاری و یا مسیر تعیین شده در خط فرمان) که نامشان شامل یک رشته‌ی خاص باشد با دستور Find. دستور زیر تمام فایل‌ها و دایرکتوری‌های موجود در مسیر C:\ که در نام یا پسوندشان کلمه‌ی pdf باشد را لیست می‌کند.

Find C:\ pdf

۱۱- چاپ کردن تمام ساختار سلسله‌مراتبی یک پارتیشن با فرمتی مناسب در خروجی با دستور TreeView. دستور زیر ساختار درختی پارتیشن C را در خروجی نمایش می‌دهد.

۱۲- ذخیره تمام سیستم فایل در یک فایل با فرمتی مشابه فرمت فایل ورودی

فایل ورودی (in.txt)

اطلاعات سیستم فایل در ساختاری مشابه فایل زیر دریافت می‌گردند. به‌عنوان مثال فایل زیر اطلاعات فایل‌ها و دایرکتوری‌های ساختار نشان داده شده در تصویر ۲ می‌باشند. همان‌گونه که مشاهده می‌نمایید، مقابل نام هر دایرکتوری علاوه بر نام دایرکتوری، ویژگی‌های دایرکتوری و تاریخ ایجاد آن آمده است. همچنین برای هر فایل، علاوه بر اطلاعات ذکر شده برای دایرکتوری‌ها، حجم فایل برحسب کیلوبایت نیز آمده است. همچنین برای هر پارتیشن، میزان فضای تخصیص داده شده به آن پارتیشن بر حسب کیلوبایت آمده است.

```
c: 52,428,800
├── Docs rw 2024-12-20
│   ├── MyDocs rw 2024-12-23
│   │   ├── doc1.txt rw 2023-02-10 175
│   │   ├── doc2.txt rw 2023-02-10 799
│   │   ├── prj1.pdf rw 2024-11-17 5377
│   │   └── homework1.docx rw 2024-10-10 2507
├── Images rw 2020-10-10
│   ├── MyImages rw 2020-10-10
│   │   ├── image1.jpg rw h 2020-10-10 6822
│   │   └── new images rw 2023-01-01
│   │       └── image2.jpeg rw 2022-08-01 3020
├── projects rw 2024-07-01
│   ├── DS rw 2024-09-01
│   │   ├── first rw 2024-09-01
│   │   │   ├── my_prj_v1.cpp r 2024-09-11 544
│   │   │   └── my_prj_v2.cpp r h 2024-09-12 823
│   │   └── second rw 2024-10-07
│   │       ├── my_prj_v1.py rw 2024-10-10 282
│   │       └── my_prj.dart r h 2024-10-12 369
│   └── DB rw 2024-11-01
│       └── db_project.cpp rw 2024-11-01 778
└── config.ini r h 2010-01-19 25
```

تذکر

- ۱- برنامه شما باید به‌صورت شی‌گرا نوشته شود. به برنامه‌هایی که به‌صورت غیر شی‌گرا نوشته می‌شوند، نمره‌ای تعلق نخواهد گرفت.
- ۲- به برنامه‌های شبیه به هم نمره‌ی منفی تعلق خواهد گرفت.