

به نام خدا

عنوان پروژه :

**نرم افزار SYNC**

تیرماه ۹۸

فهرست مطالب

۲	..... کارکرد های برنامه
۲	..... تشکیل data base و جداول
۳	..... الگوریتم و توابع sync
۳	..... Copy & delete
۵	..... گرافیک
۶	..... نحوه کار با نرم افزار

## کارکرد های برنامه

این برنامه وظیفه سینک دو فایل را دارد به این صورت که اطلاعات دو فولدر را دریافت کرده و آن هارو سینک میکند و در تاریخچه خود ذخیره میکند تا در بار بعدی که دو فولدر را سینک را میکند تنها فایل هایل جدید و فایل هایی که دچار تغییر شده اند را سینک کند

به عنوان مثال یک درایو و یک سرور را با هم سینک میکنیم و در پایان هرروز اطلاعات جدید را سینک میکنیم که با این نرم افزار به مراتب عملکرد بهتری از نظر پردازش اطلاعات و زمان ان خواهد داشت

برای اجرای برنامه **mysync** را اجرا کنید

## تشکیل data base و جداول

ابتدا توضیحی در خصوص جداول و ستون های هر کدام از جداول میدهیم.جدولی به نام **previous dir** وجود دارد که در واقع برای برنامه حکم **history** دارد ، به این صورت که هر **directory** که وارد برنامه میشود در این جدول ذخیره میشود و آخرین بار که این **directory** ها آپدیت میشوند در ستون **last check** ثبت میشود.این جدول از آن رو ساخته شده است که دیگر هر بار یک **table** جدید ساخته نشود و زمان اجرای برنامه از این طریق کاهش پیدا کند و به نوعی برنامه بهینه باشد و هر بار برای **directory** های قبلا ساخته شده ،زمان صرف ساختن دوباره **table** نشود.به طور مثال اگر **directory** ای وارد شود که حجم آن زیاد باشد و برای بار اول مثلا یک دقیقه طول بکشد،برای بار دوم فقط در حد چند ثانیه طول میکشد.

همچنین جداول دیگری برای **directory** های دیگر که وارد برنامه میشوند درست میشود.این جداول دارای ستون هایی هستند که هر کدام را به صورت مجزا و به تفصیل توضیح خواهیم داد.همچنین نام این جداول متناسب با اسم **directory** می باشد ولی از آنجا که برای نام **table** نمیتوان از \ یا / استفاده کرد تابعی در برنامه به نام **decode** و **encode** وجود دارد که **directory** دریافتی **code** میکند و رویه کد کردن نیز با دستور **replace** میباشد.مثلا برای \ از حرف \$ استفاده شده است و برای : از \_ استفاده شده است.همچنین همانطور که ذکر شد در ابتدا قبل از ساختن جدول یک **query** در **previous dir** زده میشود که آیا **id** ای متناظر با **directory** وارد شده وجود دارد یا نه .اگر وجود نداشت جدول ساخته میشود و در **previous dir** نیز زمان و مشخصات آن ثبت میشود و اگر وجود داشت باید جداول مربوطه آپدیت شوند.اگر فایلی تغییر کند و **modified time** آن جلوتر باشد ،آپدیت میشود و اگر **created time** اش جلوتر باشد آپدیت میشود.

ستون **name**: در این ستون نام هر فایل و یا فولدر وجود دارد .

ستون **path**: در این ستون آدرس هر فایل یا فولدر وجود دارد و نام **path** هم به این صورت است که ابتدای آدرس یعنی مثلا **D:\** حذف شده است .

ستون **rank** : برای هر فایل و یا فولدر یک متغیر به نام **rank** وجود دارد که از صفر شروع میشود .مثلا فایل های صفحه اول **directory** داده شده فایل های مرتبه صفر هستند.وقتی وارد یک فولدر میشویم فایل های داخل آن صفحه برابر مرتبه دو هستند و به همین شکل مرتبه بیشتر و بیشتر میشود.نکته مهم آن که فولدر از مرتبه یک شروع میشود.

ستون **created**: مربوط به **created time** برای هر فایل یا فولدر می باشد.

ستون **modified**: مربوط به **modified time** برای هر فایل یا فولدر می باشد..

ستون **size \_ file**: **size** مربوط به هر فایل و فولدری را نشان میدهد.

ستون **flag**: برای آن است که بررسی شود کدام فایل یا فولدر باید کپی شود.

ستون **copied**: ستونی است که مشخص میکند که بررسی سود یا خیر.

ستون **is\_Folder**: برای بررسی کردن اینکه فایل است یا فولدر.

توجه: نام فولدر **path** آن میباشد و **path** آن **parent** مربوط به آن میباشد. برای گرفتن **parent** از دستور **pardir** استفاده میکنیم.

## الگوریتم و توابع sync

الگوریتمی که در این بخش اتخاذ شده است به این صورت است که پس از وارد کردن **directory** ابتدا فایل های مرتبه صفر و فولدر های مرتبه ۱ در دو **directory** داده شده، مقایسه میشوند. در اینجا فایل های مشترک و فولدر های مشترک در دو **directory** تشخیص داده میشوند و فایل ها و فولدر های غیر از آنها، **flag** مربوط به آنها به گونه ای میشود که نشان دهد باید کپی شوند. مثلاً **flag** مربوط به آنها '5' میشود.

همین مراحل برای **rank** های بعدی استفاده میشوند. و همیشه در بررسی و آپدیت کردن فایل ها و فولدر های متفاوت **rank** فایل یک عدد کمتر از **rank** فولدر است. برای آنکه همین روند را ادامه دهیم ابتدا **max(rank)** را بدست آورده و به تعداد آن **update** کردن و **select** کردن انجام میشود.

اگر یک فولدر برای کپی کردن انتخاب شد دیگر تمامی فایل های داخل آن و فولدر های داخل آن همگی **copied** شان برابر '8' خواهد شد به این معنا که دیگر نمیخواهد آنها را بررسی کنی. در مرحله بعدی که سراغ فایل های مرتبه ۱ میرویم که این فایل ها یا فولدر ها، موارد مشترک بین هر دو هستند (چرا که موارد غیر مشترک فقط کپی میشوند) دیگر این فایل ها چون **copied** شان برابر '8' شده است دیگر مورد بررسی برای کپی کردن قرار نمیگیرند. این عملیات توسط **parent** فایل ها و فولدر ها انجام میشود و تمامی فولدر های تودرتو را پوشش می دهد.

**هوشمندی الگوریتم** آن جا مشخص میشود که دو بخش را ترکیب میکند در واقع برای اولین بار که دایرکتوری جدید وارد میشود از الگوریتم های بالا استفاده می کند و برای ورودی های بعدی از دستوراتی در بخش تشکیل جداول عمل میکند این کار توسط **modified time** و **created time** انجام میشود

چنانچه فولدر یا فایل جدیدی ایجاد شود با استفاده از **created time** آن را به جدول اضافه میکند و چنانچه فایل آپدیت شود و محتویات آن تغییر کند حتی در صورتی که سائز آن هم تغییر نکد با کمک **modified time** آن را اضافه میکند این عملیات برای ایجاد و تغییر در فایل ها و فولدر های پیچیده و تودرتو نیز کار میکند.

## Copy & delete

وقتی یک فایل **modified** شود یعنی تغییر در آن صورت گیرد، برای اینکه فرآیند **sync** صورت گیرد ابتدا باید فایل موجود در طرف دیگر پاک شود و فایل جدید در آنجا کپی شود. به طور کلی فایلی که تایم **modified** آن جلوتر است برای باقی ماندن مقدم تر است. همچنین کپی کردن فایل و فولدر جدای از هم می باشد و این امکان وجود دارد که حتی **فایل های تو در تو نیز پس از تغییر پاک و کپی میشوند.**



## گرافیک:

در صفحه اول برنامه زمان اجرا تعدادی کلید میبینیم که آن ها به تناسب با روند اجرای کار روشن میشوند به فرص در ابتدا کلید **analyze** و **sync** خاموش هستند زیرا هنوز کاربر دایرکتوری ای را وارد نکرده که بتوان روی آن ها کار انجام داد.

به فرض که کاربر نیز یک دایرکتوری را فقط بدهد، باز هم این دو دکمه خاموشند و آن ها زمانی روشن میشوند که هر دو دایرکتوری داده شده باشند.

بعد از وارد کردن دایرکتوری ها و فعال شدن دکمه های نامبرده ، پس از زدن دکمه ی **analyze** ، تمام فایل ها و فولدر های درون آن دو دایرکتوری را نمایش میدهد( با رنگ های متفاوت) که این فایل ها به صورت **treeView** میباشند و با زدن روی هر فولدر ، محتویات آن فولدر تماما نمایش داده میشوند.

همچنین پس از زدن این دکمه نیز یک **thread** اجرا شده و آن شروع به کار میکند که تابعی که آن ترد کال میکند، تابعی که دیتا بیس این دو دایرکتوری را میسازد و **table** های مورد نیاز را تولید میکند.

در قسمت پایین و سمت چپ پنجره مورد نمایش، یک **progressBar** موجود میباشد که تا زمانی که ساخت این **table** ها طول میکشد ، در حال لود شدن و اجرا میباشد بدون اینکه هیچگونه خللی در کار گرافیک را ایجاد کند.

همچنین پس از اجرا کامل تابع سازنده ی دیتا بیس، این **prigressBar** مقدار ۱۰۰ درصد را نمایش داده و از صفحه نمایش پاک شده و به جای آن یک آلارم با این مضمون که " ساخت دیتابیس با موفقیت انجام شد" نمایش داده میشود.

حال در صورتی که کاربر دکمه **sync** را کلیک کند، باز هم یک ترد شروع به کار میکند که تابعی که آن ترد کال میکند همان تابعی است که دو دایرکتوری را بار هم سینک کرده و محتویات آن ها را با هم یکی میکند( که آن را از ماژولی که **import** کرده و خودمان آنرا نوشته ایم ، کال میکنیم).

همزمان با سینک کردن، یک **progressBar** در قسمت پایین ، سمت چپ پنجره وجود دارد که آن، تا زمان کامل شدن سینک، به صورت آنلاین تغییر میکند و هنگامی که سینک کامل شد، آن **progressBar** از صفحه محو شده و آلارمی با این مضمون که "سینک با موفقیت انجام شد" نمایش داده میشود.

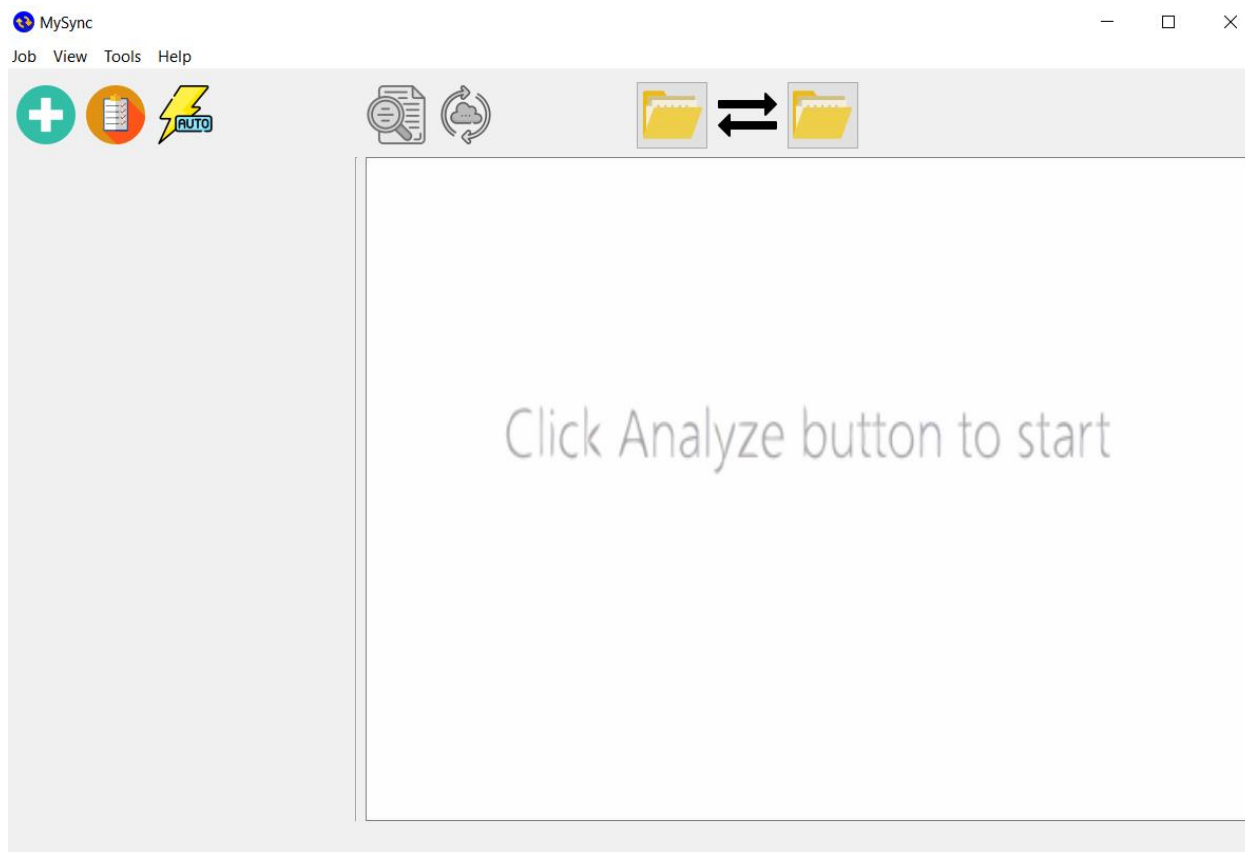
همچنین در حین انجام سینک کردن، آن قسمت که تمام فولدر ها و فایل های دایرکتوری داده شده را نمایش میداد، به صورت آنلاین و لحظه ای تغییرات آن فولدر ها را نمایش میدهد، و آنها را در لحظه آپدیت کرده و نحوه سینک کردن را به وضوح میتوان دید.

که به عبارتی آن فایل ها و فولدر هایی که در نهایت رو صفحه نمایش داده میشوند، همان فایل ها و فولدر های نهایی و سینک شده اند.

## نحوه کار با نرم افزار

-در ابتدا صفحه ای باز میشود که در آن میتوان **directory** ها را وارد کرد و این کار با کلیک کردن بر روی آیکون پوشه صورت میگیرد.

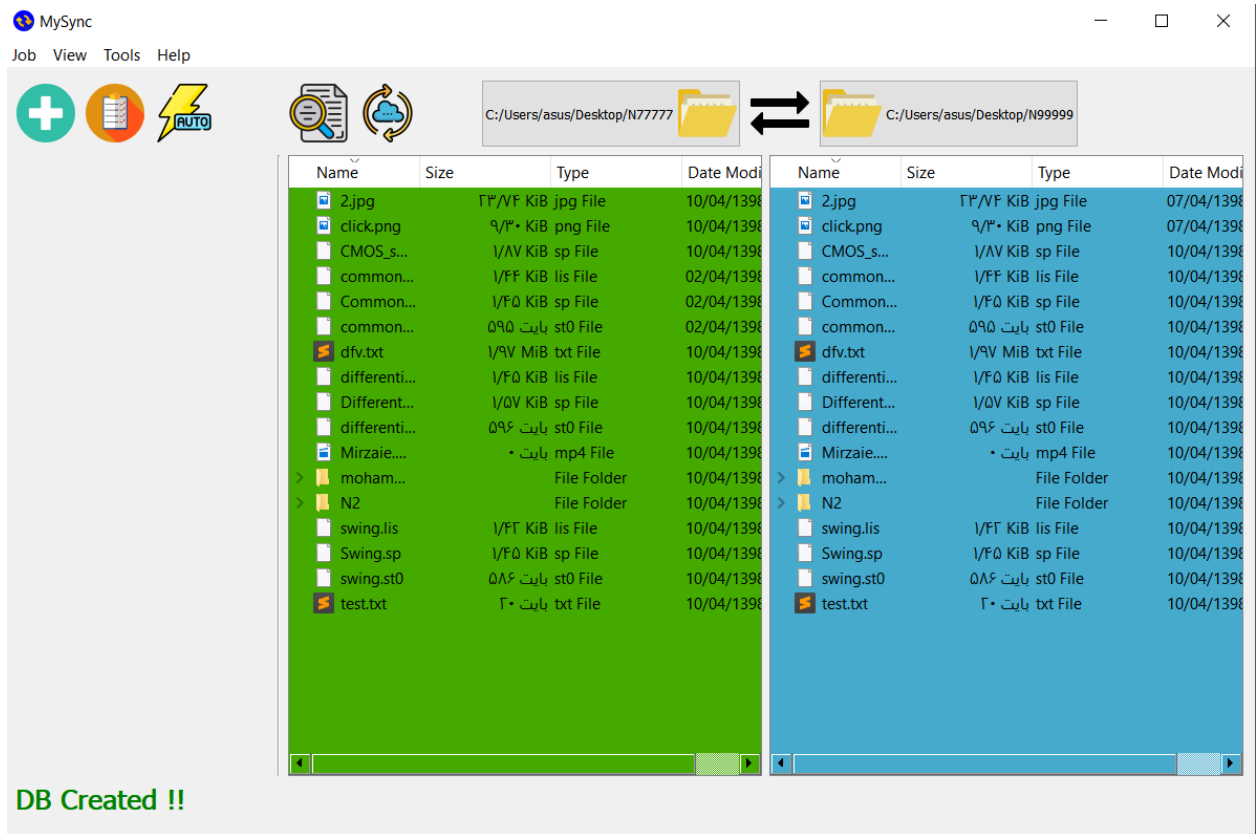
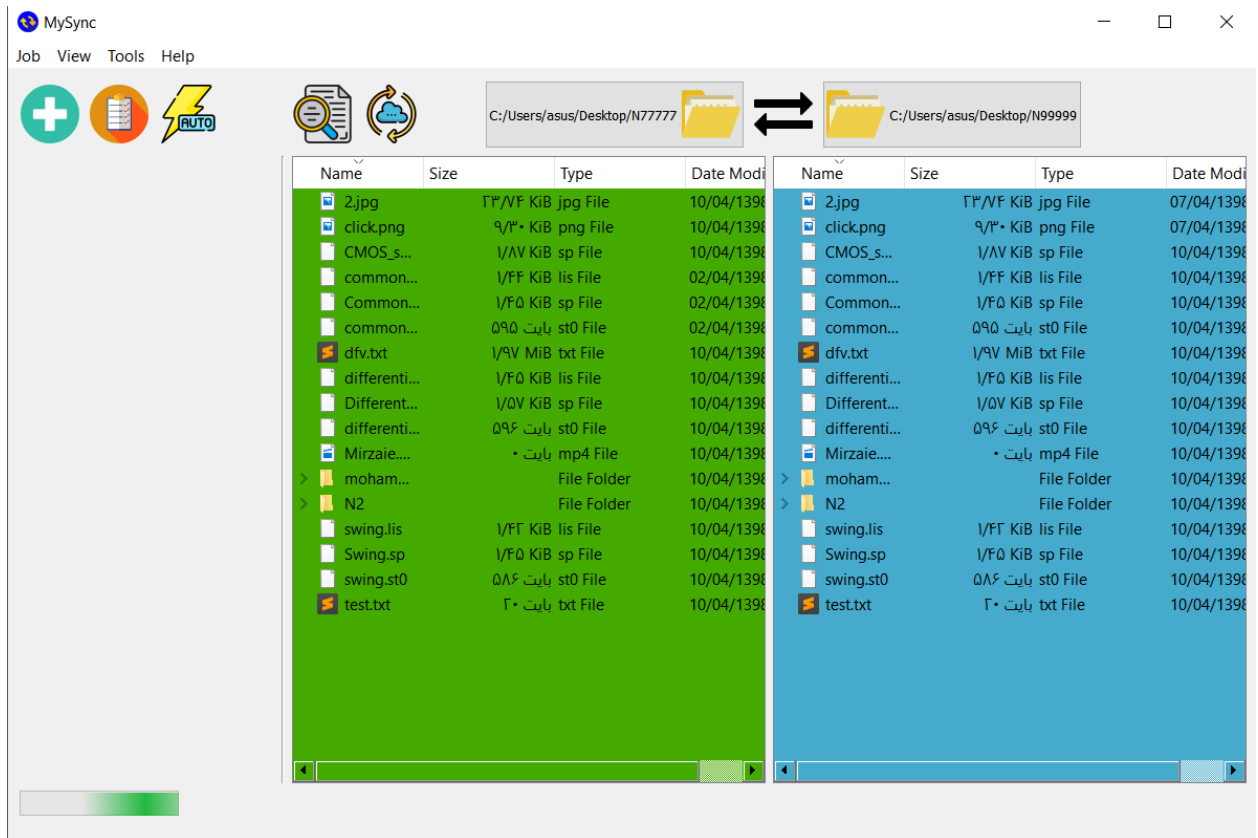
-در ابتدا آیکون مربوط به آنالیز و **sync** خاموش است و پس از وارد کردن **directory** به رنگ روشن در می آید.



سپس با زدن دکمه آنالیز لیست **directory** های وارد شده نمایش داده می شود. که این لیست به صورت یک **tree** است و با کلیک کردن بر روی **folder** ها میتوان محتویات داخل آنها را مشاهده کرد.

پس از تشکیل **data base** عبارت گوشه پایین سمت چپ یعنی **db created** نمایش داده میشود.

همچنین در هنگام ساختن **table** یا عملیات **sync** کردن نوار باری در پایین نمایش داده میشود.



همچنین مانند عملیات ساخت data base، پس از پایان عملیات sync عبارت synced نمایش داده میشود.

