

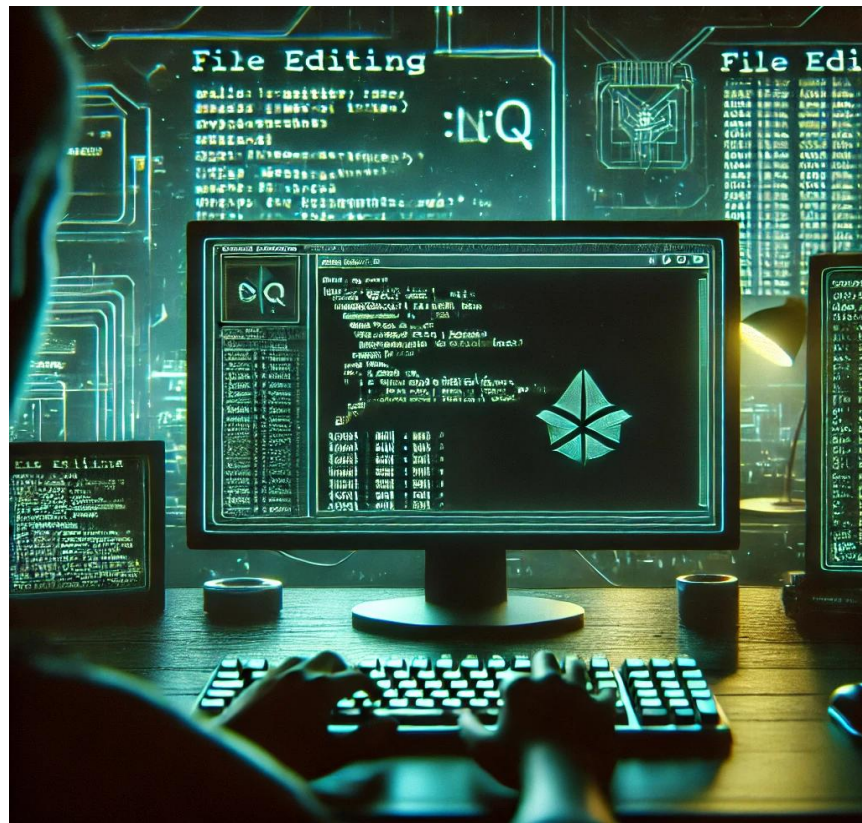


پروژه دوم برنامه نویسی پیشرفته

سید ابوالقاسم میرروشندل و آرش یوسفی

ترم 14032

پای نانو PyNano



مقدمه

نانو Nano یک ویراشگر متن سبک است که در محیط های لینوکس و یونیکس برای ویرایش فایل های متنی استفاده می شود. این ویراشگر مبتنی بر ترمینال است و بدون نیاز به محیط گرافیکی کار می کند، بنابراین برای کاربران و برنامه نویسانی که با سرور ها و مدیریت فایل های سیستمی سرو کار دارند بسیار مفید است.

در لینوکس با وارد کردن دستور زیر محتویات داخل فایل filename به نمایش گذاشته می شود و قابل ویرایش است.

```
nano filename.txt
```

مباحث: مباحث پروژه ۱، برنامه سازی شی گرا، کار با فایل ها و دایرکتوری ها

هدف پروژه

این پروژه به شما فرصت می دهد تا اصول برنامه نویسی شی گرا OOP را پیاده سازی کنید.

• ساختار کلاس ها و اشیاء Classes & Objects

شما باید هر بخش از ویراشگر متن (مانند مدیریت فایل، مدیریت کاربران، پردازش متن و کنترل ورودی) را در یک کلاس جادوگانه پیاده سازی کنید. این کار باعث سازماندهی و افزایش خوانایی کد شما می شود.

• متود های کلاسی و استاتیک Static & Class Methods

در آبجکت یا کلاس شما که نیاز به پیداسازی یک سری الگوریتم و منطق دارد شما متود هایی تعریف میکنید، به عنوان مثال متود edit_file() در کلاس ادیتورتون.

• مدیریت استثناها Exception Handling

هنگام خواندن یا ذخیره سازی فایل ها باید از بلاک های try-except استفاده شود تا در صورت بروز خطا (به عنوان مثال وجود نداشتن فایل) برنامه کرش نکند و خروجی معتبری به کاربر بدهد.

• کدنویسی ماژولار Modular Coding

به جای نوشتن همه کد ها در یک فایل، کد را به ماژول های مستقل تقسیم کنید. به عنوان مثال ساختن فایل های مختلف فقط برای کلاس ها. Editor.py/main.py/etc.

• برنامه نویسی مبتنی بر دستورات Command-Based Programming

پروژه باید بر اساس یک سری دستورات که داخل ترمینال کاربر وارد میکند باید کار کند و گزینه های انتخابی مانند منو یا حتی رابط گرافیکی ندارد.

توضیح پروژه

بعد از اجرای برنامه خود در shell یا ترمینال IDE خود، کاربر باید با وارد کردن دستور لاگین و پاس دادن یوزر مربوطه، پسورد خود را وارد کند و تایید هویت بشود.

به عنوان مثال:

```
python3 main.py
>> login root
Please state your password:
// enter password
>> login as root successful!
```

بعد از تایید هویت شما قابلیت دسترسی به چندین نوع دستور را خواهید داشت: cd, nano, ls, del و pwd

- با دستور nano و پاس دادن فایل مورد نظر به آن میتوانید محتوای فایل مورد نظر خود را مشاهده کنید و در صورت داشتن مجوزهای لازم، آن را ویرایش بدهید.
- با دستور cd می‌توانید بین دایرکتوری‌ها جابجا شوید.
- با دستور ls فایل‌های موجود در دایرکتوری فعلی را مشاهده کنید.
- با دستور pwd مسیر دایرکتوری فعلی خود را ببینید.
- با دستور del و پاس دادن فایل مورد نظر، در صورت داشتن مجوز، آن فایل پاک می‌شود.

به عنوان مثال:

```
>> pwd
Current directory is C:\Users\ASUS\Desktop\Ashkan
>> ls
These files and folders exist in this directory:
file1.txt
folder1
>> cd folder1
Directory changed to C:\Users\ASUS\Desktop\Ashkan\folder1
>> pwd
Current directory is C:\Users\ASUS\Desktop\Ashkan\folder1
>> cd ..
Directory changed to C:\Users\ASUS\Desktop\Ashkan\
>> nano file1.txt
// mechanism for editing
```

توضیح تکمیلی

سیستم یوزر:

برای سیستم یوزر خود باید یکسری کاربر پیش فرض (با رمز عبور های پیش فرض) برای برنامه ی خود داشته باشید. این کاربر های پیش فرض شامل support, guest و root خواهند بود.

- در صورتی که با root وارد شدید دسترسی خواندن، حذف و ویرایش فایل را دارید.
- با وارد شدن به عنوان support اجزای خواندن و ویرایش فایل را دارید اما نمیتوانید فایل حذف کنید.
- و در صورتی که با guest وارد شدید فقط حق خواندن فایل هارا دارید.
- لازم به ذکر است که برنامه ی شما باید یک دستور logout هم داشته باشد.

راه اندازی ثبت نام و دسترسی دادن از اهداف پروژه نیست.

دستورات مدیریت directory:

همانطور که گفته شد برنامه‌ی شما باید مجهز به دستوراتی باشد که بتوانید بین فولدر ها و دایرکتوری‌ها جابجا شوید. شما به‌طور پیش‌فرض در همان فولدر پروژه‌ی خود خواهید بود.

برای پیاده‌سازی این بخش از ماژول OS کمک بگیرید.

باز کردن فایل:

در صورتی که فایل وجود نداشته باشد و کاربر وارد شده root باشد باید فایل مورد نظر در آن directory ساخته شود. توجه کنید که محتویات فایل باید به صورت تمیز داخل ترمینال نشان داده شود و داخل خود ترمینال شما ویرایش هایتان را انجام بدهید. (بدین منظور نیست که فایل داخل سیستم شما باز شود).

ویرایش فایل:

در صورتی که کاربر دسترسی ویرایش داشته باشد می‌تواند فایل را ویرایش دهد. ویراشگر شما باید یک مکاننما یا cursor داشته باشد که با arrow key های کیبوردتان قابل جابجایی باشد و قابلیت تایپ و استفاده از backspace را داشته باشد.

ذخیره سازی فایل:

در صورتی که ویرایش شما با فایل به اتمام رسید با فشار دادن CNTRL + X برنامه باید خارج شود. قبل از خارج شدن در صورت save نبودن تغییرات باید برنامه هشدار بدهد. با وارد کردن دستور CNTRL + O تغییرات شما save میشود.

دیلیت و پاک کردن فایل:

کاربر در صورتی که root باشد میتواند با وارد کردن دستور del filename.txt آن فایل را پاک کند.

پروژه شما باید حتما نکات و اصول شیءگرایی و اهداف را رعایت کند وگرنه نمره‌ای به پروژه تعلق نمی‌گیرد.

امتیازی:

- پیاده سازی undo / redo: پیاده سازی سیستم بازگردانی تغییرات CNTRL+Z و انجام مجدد آن CNTRL+Y. پیشنهاد میشود اگر قابلیت را پیداسازی کنید از Stack یا پشته استفاده کنید.

- **قابلیت جستجوی متن در فایل:** با استفاده از Regex کلمه ی مدنظر خودتان را در داخل فایل سرچ کنید و در صورت موجود بودن آن، آن کلمه هایلات شود. (میتوانید از کد های ANSI برای رنگ آمیزی متن استفاده کنید).
- **Autosave:** برنامه شما این قابلیت را داشته باشد که در حین ویرایش تغییرات یک Thread جداگانه ویرایش های شما را مدام ذخیره کند که در صورتی که برنامه ناگهان بسته شد تغییرات ذخیره بمانند و هنگام باز شدن برنامه برای همان فایل مشخص، برنامه از کاربر برای ادامه تغییرات سوال کند.

ثبت پروژه

پروژه خود را در فایل فشرده Zip با فرمت زیر بنویسید و آن را در کوئرا ارسال کنید.

APPROJ2_FirstNameLastName_StudentID.zip

نکات عمومی پروژه‌ها

هرگونه ایده و خلاقیت جدید، تنها با به کارگیری مباحث پروژه و سرفصل‌های مرتبط با این درس و با هماهنگی با دستیاران آموزشی، تا ۲۵ درصد نمره مثبت به همراه خواهد داشت.

پروژه شما تنها در صورتی مشمول **نمره مثبت** می‌شود، که بخش اصلی پروژه را بیش از **۹۵ درصد** تکمیل کرده باشید.

کپی و استفاده کورکورانه از منابع اینترنتی، یا ابزارهایی نظیر Chat-GPT ممنوع است و شامل جریمه خواهد بود.

پروژه تنها به صورت انفرادی قابل انجام بوده و وجود هرگونه تشابه مشکوک بین دو کد یا عدم تسلط به کد در طول ارائه، **شامل جریمه خواهد بود.**

نام گذاری صحیح، کامنت نویسی مناسب، رعایت دندانده‌گذاری، و خوانا بودن کد **الزامی است.**