

شرح پروژه سیستم عامل (بخش دوم)

هدف پروژه

هدف این پروژه، خواندن یک فایل داده‌ای حاوی فروش بازی‌های ویدیویی و یافتن بازی با بیشترین و کمترین فروش جهانی است که بصورت چند نخ اجرا خواهد شد تا نهایت بهره‌وری را از cpu داشته باشیم.

کتابخانه‌های استفاده شده

pandas , threading , time

نحوه کارکرد پروژه

1. با استفاده از تابع find_top فایل csv بازی‌های ویدیویی خوانده می‌شود و سپس بازی با بیشترین فروش جهانی یافت شده و نمایش داده می‌شود.
2. با استفاده از تابع find_low فایل csv بازی‌های ویدیویی خوانده می‌شود و سپس بازی با بیشترین فروش جهانی یافت شده و نمایش داده می‌شود.
3. سپس خروجی این توابع نمایش داده می‌شوند
4. در نهایت با محاسبه زمان اجرای برنامه و نمایش آن، برنامه خاتمه می‌یابد.

توضیحات خط به خط

ابتدا به وسیله دستور Import کتابخانه‌های مورد نیازمان را در فایل پایتون فراخوانی می‌کنیم

```
import pandas as pd
import threading
import time
```

در قدم بعد یک متغیر به نام start تعریف می‌کنیم تا قبل از اجرای کد زمان مربوط به آن لحظه را در خود ذخیره کند

```
start = time.time()
```

در این قسمت باید یک تابع تعریف کنیم ، نام تابع را بصورت پیشفرض find_top میگذاریم و میخواهیم این تابع مشخصات بازی که بیشترین فروش را داشته بیابد. وارد بدنه تابع میشویم و شروع به نوشتن کدهای اصلی میکنیم

ابتدا باید فایل csv را بخوانیم و آن را تبدیل به یک دیتافریم کنیم برای انجام این کار به روش زیر عمل میکنیم

```
df = pd.read_csv('./video-game-sales.csv')
```

حالا باید بررسی کنیم و ببینیم آیا داده های Null هم وجود دارند یا خیر اگر وجود داشتند آنها را برای ما حذف کند. برای اینکار از متد dropna استفاده میکنیم

```
df = df.dropna()
```

در قدم بعدی یک متغیر به نام top_seals ایجاد میکنیم تا بازی را پیدا کنیم که بیشترین فروش را داشته است برای پیدا کردن بازی که بیشترین فروش را داشته باید از ستون Global_seals مقداری را انتخاب کنیم که بزرگترین باشد برای اینکار به روش زیر عمل میکنیم

```
top_sales = df.loc[df['Global_Sales'].idxmax()]
```

حالا مقدار پیدا شده را پرینت میکنیم تا مشخصات آن بازی را به ما نمایش دهد

```
print(top_sales)
```

یک تابع دیگر تعریف کنیم ، نام تابع را بصورت پیشفرض find_low میگذاریم و میخواهیم این تابع مشخصات بازی که کمترین را داشته بیابد. وارد بدنه تابع میشویم و شروع به نوشتن کدهای اصلی میکنیم

ابتدا باید فایل csv را بخوانیم و آن را تبدیل به یک دیتافریم کنیم برای انجام این کار به روش زیر عمل میکنیم

```
df = pd.read_csv('./video-game-sales.csv')
```

حالا باید بررسی کنیم و ببینیم آیا داده های Null هم وجود دارند یا خیر اگر وجود داشتند آنها را برای ما حذف کند. برای اینکار از متد dropna استفاده میکنیم

```
df = df.dropna()
```

در قدم بعدی یک متغیر به نام low_seals ایجاد میکنیم تا بازی را پیدا کنیم که کمترین فروش را داشته است برای پیدا کردن بازی که بیشترین فروش را داشته باید از ستون Global_seals مقداری را انتخاب کنیم که کوچکترین باشد برای اینکار به روش زیر عمل میکنیم

```
low_sales = df.loc[df['Global_Sales'].idxmin()]
```

حالا مقدار پیدا شده را پرینت میکنیم تا مشخصات آن بازی را به ما نمایش دهد

```
print(low_sales)
```

هر تابع اطلاعات مختلفی را به ما میدهد، میخواهیم کد به صورت چند نخ اجرا شود یعنی هر دو تابع بصورت همزمان اجرا شوند و نهایت استفاده از cpu را ببریم و برنامه سریع تر و کارآمد تر شود

برای ایجاد نخ دو تا متغیر تعریف میکنم و هر تابع را به یکی از آنها میدیم تا نخ ما را ایجاد کند برای اینکار متد Thread را از کتابخانه threading فراخوانی میکنیم و به عنوان تارگت، نام تابع را به آن میدهیم

```
t1 = threading.Thread(target=find_top)
t2 = threading.Thread(target=find_low)
```

حالا به وسیله متد start نخ های ما شروع به کار میکنند

```
t1.start()  
t2.start()
```

باید مطمئن شویم که هر نخ ما به درستی تا انتها اجرا میشود برای اینکار متد Join را روی نخ های خود فراخوانی میکنیم

```
t1.join()  
t2.join()
```

در قدم بعد یک متغییر به نام end تعریف میکنیم تا بعد از اجرای کد زمان مربوط به آن لحظه را در خود ذخیره کند

```
end = time.time()
```

حال مدت زمان انجام شده نیز در پایان به ما نمایش داده میشود

```
print(end - start)
```