

بسم الله الرحمن الرحيم

این مستند فنی به تشریح نحوه پیاده‌سازی یک سیستم برای ثبت و ذخیره‌سازی داده‌های مربوط به حافظه (RAM) در یک پایگاه‌داده SQLite می‌پردازد. این مستند شامل دو بخش اصلی است: فایل اصلی برای ثبت داده‌های RAM در پایگاه داده و ماژولی برای بازیابی اطلاعات حافظه از سیستم عامل.

## ساختار کلی پروژه:

پروژه از دو فایل تشکیل شده است:

1. **فایل اصلی `insert_ram_data.py`**: اسکریپتی که به‌طور مستمر داده‌های RAM را بازیابی کرده و در پایگاه داده ذخیره می‌کند.
2. **ماژول `Get_Memory_Data`**: کدی که اطلاعات حافظه (Total, Used, Free) را از سیستم عامل لینوکس استخراج می‌کند.

## 1. فایل اصلی: `insert_ram_data.py`

این فایل وظیفه ثبت داده‌های مربوط به RAM را در یک پایگاه‌داده SQLite بر عهده دارد. در این فایل، عملیات استخراج داده‌های حافظه از طریق ماژول `Get_Memory_Data` صورت می‌گیرد و سپس داده‌ها در جدول مربوطه ذخیره می‌شوند.

توضیحات کد:

### • تابع `get_db_connection()`:

- این تابع اتصال به پایگاه داده را از طریق `sqlite3.connect()` برقرار می‌کند.
- فایل پایگاه داده به نام `ram_data.db` در مسیر بالاتری از مسیر جاری ذخیره خواهد شد.

### • تابع `insert_ram_data()`:

- ابتدا با استفاده از `get_ram_info()` اطلاعات RAM را از ماژول `Get_Memory_Data` دریافت می‌کند.
- سپس این داده‌ها را در جدول `ram` پایگاه‌داده با استفاده از `sql query` ذخیره می‌کند.
- پس از درج داده‌ها، اتصال به پایگاه داده بسته می‌شود.

### • حلقه اصلی:

- این اسکریپت در یک حلقه بی‌نهایت اجرا می‌شود و در هر چرخه داده‌ها را ذخیره کرده و پس از 60 ثانیه توقف، دوباره اجرا می‌شود.
- در صورت بروز خطا، پیام خطا نمایش داده می‌شود و سپس اسکریپت مجدداً به مدت 60 ثانیه متوقف می‌شود.

## 2. مازول `Get_Memory_Data`

این مازول به منظور بازیابی اطلاعات حافظه (RAM) از سیستم عامل لینوکس طراحی شده است.

توضیحات کد:

- `os.popen('free -m'):`

- این دستور، ابزار `free` در سیستم عامل لینوکس را اجرا می‌کند که اطلاعات مربوط به حافظه را به مگابایت نمایش می‌دهد.

- `process.read():`

- خروجی این ابزار به صورت رشته‌ای خوانده می‌شود.

- تجزیه اطلاعات:

- خروجی به چند خط تقسیم شده و خط دوم (`[1]lines`) شامل اطلاعات مورد نیاز می‌باشد.
- ستون‌های `used`، `total`، و `free` از خط دوم استخراج و به عنوان یک دیکشنری بازگردانده می‌شوند.

- کنترل خطا:

- در صورت بروز هرگونه خطا در اجرای دستور یا تجزیه اطلاعات، پیام خطا چاپ شده و مقدار `None` برگردانده می‌شود.

برای اجرای اسکریپت اصلی، کافی است که فایل `insert_ram_data.py` را اجرا کنید

## ۳. نکات مهم

### 1. پشتیبانی از سیستم عامل:

- مازول `Get_Memory_Data` تنها در سیستم عامل‌های مبتنی بر لینوکس که دستور `free` در آن‌ها موجود باشد، به درستی کار می‌کند.

### 2. خطایابی:

- در صورت بروز خطا، اسکریپت به طور خودکار به مدت 60 ثانیه متوقف می‌شود و سپس مجدداً تلاش می‌کند.

### 3. بهینه‌سازی:

- برای بهبود کارایی، می‌توانید زمان وقفه (`time.sleep`) را بسته به نیاز خود تغییر دهید.