

Python ile Otomatik Paket Yönetimi ve İşletim Sistemleri Araştırması

Linux (Kali) Ortamında Modern Paket Yönetim Süreçleri

Hazırlayan: Aydın Arda Demiralp



GitHub Yapısı ve Standardizasyon

Profesyonel Klasörleme

Proje, endüstri standartlarına uygun olarak modüler bir klasör yapısıyla düzenlenmiştir. Bu yapı, hem okunabilirliği artırır hem de ölçeklenebilirliği destekler.

src/: Python Kaynak Kodları

Tüm Python uygulama kodları bu dizin altında yer alır. Ana mantık ve yardımcı modüller burada bulunur, temiz ve düzenli bir kod tabanı sağlar.

researchs/: Yapay Zeka Analizleri

ChatGPT ve Gemini gibi yapay zeka modelleriyle yapılan karşılaştırmalı analizlerin sonuçları ve verileri burada hiyerarşik olarak saklanır.

docs/: Proje Rehberleri ve Dokümantasyon

Proje ile ilgili tüm rehberler, kullanım kılavuzları ve detaylı dokümantasyon bu klasörde bulunur. Kolay erişim ve anlaşılabilirlik hedeflenmiştir.

specs/: Teknik Gereksinimler

Projenin teknik şartnameleri ve gereksinim analizleri bu bölümde listelenmiştir. Geliştirme sürecinin temelini oluşturan detaylar burada tanımlanır.

Çoklu Kaynak Kullanımı ve AI Analizi

Bu projede, karşılaşılan teknik sorunları çözmek için tek bir bilgi kaynağına bağı kalınmamıştır. Farklı perspektifler ve çözümler elde etmek amacıyla çeşitli platformlardan yararlanılmıştır.

1

Çoklu AI Modeli Entegrasyonu

Researchs klasörü altında, Gemini Pro, Gemini Fast ve ChatGPT gibi önde gelen yapay zeka modellerinden elde edilen veriler hiyerarşik bir yapıda depolanmıştır.

2

Teknik Çözüm Yaklaşımlarının Analizi

Her bir yapay zeka modelinin sunduğu teknik çözüm yaklaşımları detaylı bir şekilde analiz edilerek, en uygun ve verimli çözümlerin entegrasyonu sağlanmıştır.

3

Kapsamlı Veri Saklama

Tüm araştırma verileri, gelecekteki referanslar ve proje geliřtirmeleri için düzenli bir şekilde saklanmaktadır.



Python ile Paket Yönetimi



Python Tabanlı Yönetim

Proje, Linux'un **apt** ve **dpkg** paket yönetim sistemlerini Python üzerinden kontrol etmeyi amaçlar. Bu yaklaşım, otomasyon ve esneklik sağlar.



Otomatik Paket Kurulumu

Belirtilen paketlerin otomatik olarak kurulmasını ve bağımlılıklarının çözülmesini sağlar. Bu, manuel müdahaleyi azaltır ve süreçleri hızlandırır.



Hata Yakalama ve Loglama

Paket kurulumu veya yönetim süreçlerinde meydana gelen hatalar anında yakalanır ve detaylı log kayıtları **researchs** klasörüne kaydedilir.



Sistem Güncelliği Kontrolü

Sistemin kurulu paketlerinin güncel olup olmadığını otomatik olarak kontrol eder ve güncel olmayan paketler için öneriler sunar.

Python

Python

Python

Python:(>:

```
pioenvverh1(IoP&esc 8=idag:0))>:
```

```
Linux:((ce(-1/)>:
```

```
8pthos:(/11lc)>:
```

```
Linux:((oitr==_cfctcshmi slch&Lr5)(1>);
```

```
fiemmi(latt,(1a));
```

```
pioikomi(cebah==&(Io),&alc)>:
```

```
1/);
```

```
1:
```

```
Desatrast1>:
```

```
Linux:(Kerlcal?);;
```

```
Linux:(4lofcl);;
```

```
Linux(ck/inhtcom buxtlig(gc txdt "tatitek)
```

```
Renux:(/lolc1);;
```

```
Linux(bytids,1lc);;
```

```
Dyouxi:(cbes=Iiax jtheetot:It111);;
```

```
1:
```

```
1:
```

```
Linux:((<d: eriatodk==sl&)>:
```

```
[root@nakl(dox,t=all out_dor:117)>:
```

```
Linux:(/det) (cc)>:
```

```
Liokimi(ce pbat,15nn:17)>:
```

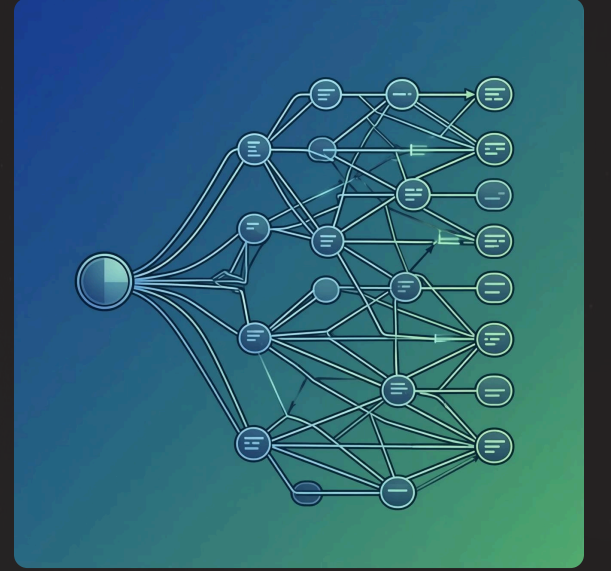
```
1:
```

```
1:
```


Versiyon Kontrolü (Git & GitHub)

Projenin başlangıcından itibaren, geliştirme süreci boyunca **Git** versiyon kontrol sistemi aktif olarak kullanılmıştır.

Bu sayede her değişiklik izlenebilir, geri alınabilir ve farklı sürümler arasında kolayca geçiş yapılabilir hale gelmiştir.



Düzenli Commit ve Push

Projenin gelişimi, düzenli olarak yapılan **commit** ve **push** işlemleriyle detaylı bir şekilde kaydedilmiştir. Bu, projenin adım adım nasıl ilerlediğini gösterir.

Açık Kaynak Erişimi

Proje, [aardademiralp/Package-Handling](https://github.com/aardademiralp/Package-Handling) adresindeki GitHub deposunda **açık kaynak** olarak sunulmuştur. Bu, topluluk katkılarına ve işbirliğine olanak tanır.