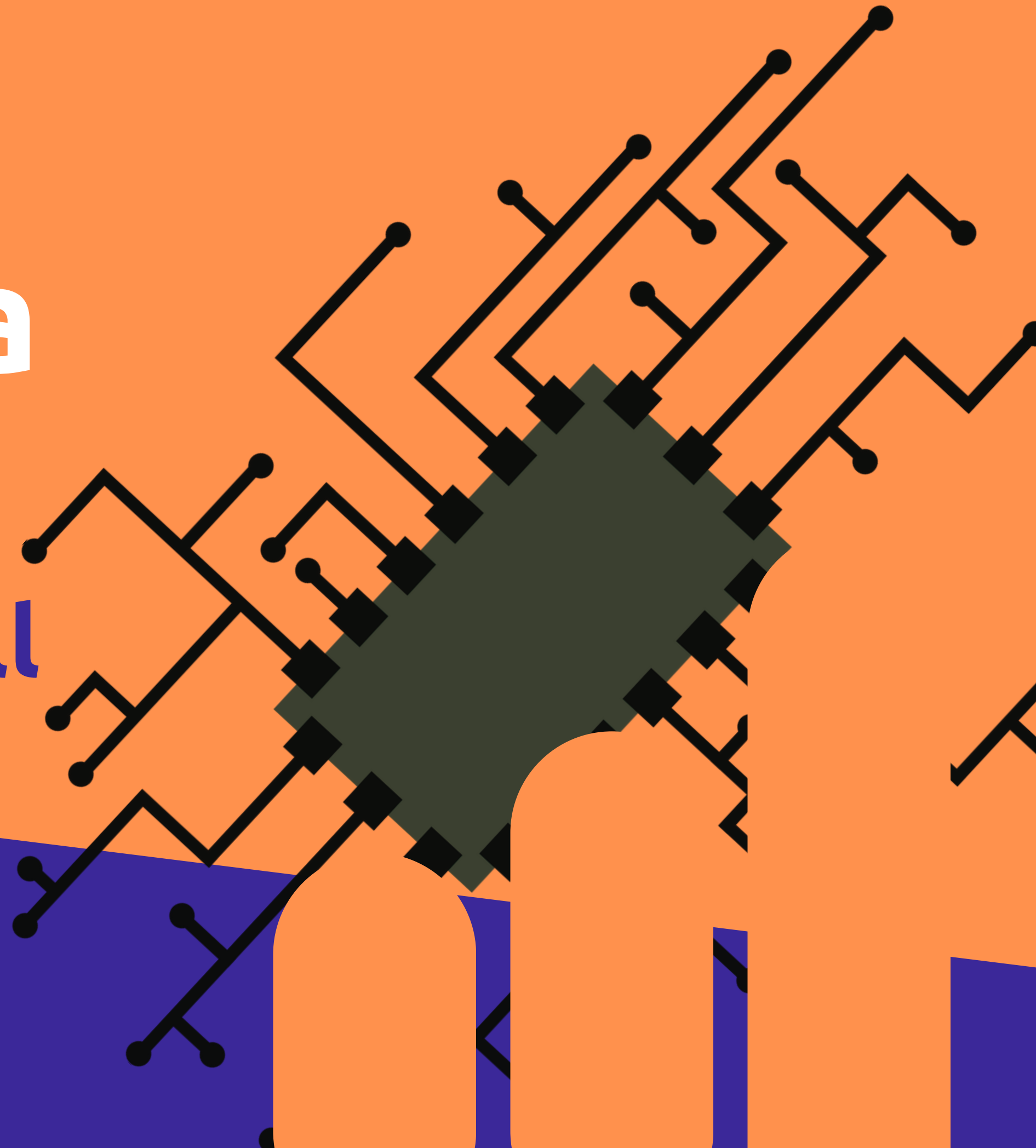


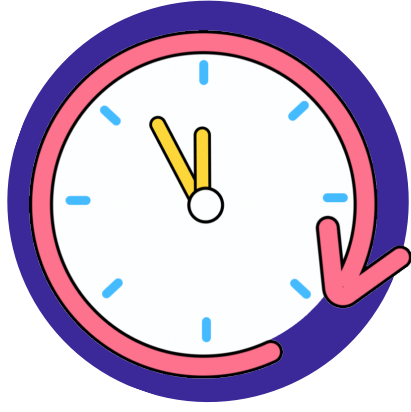
Sistem Programlama

Proje Sunumu

Multithreading Shell

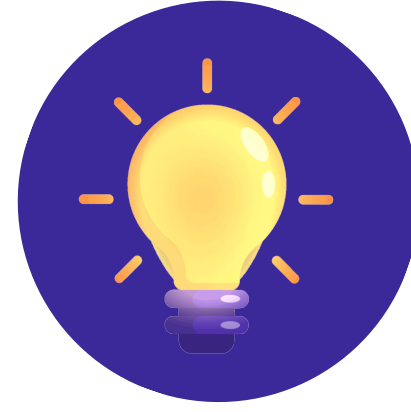


Tasarruftan Ötesi!



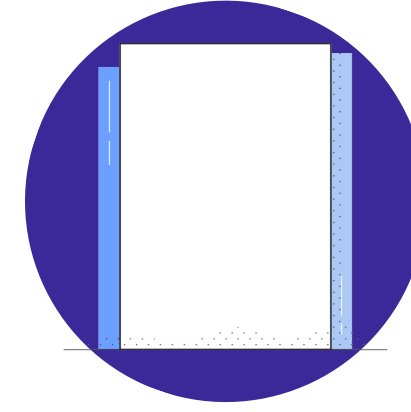
Zaman Tasarrufu

Ağ işlemleri gibi zamana bağlı komutlar hiç bu kadar hızlı olmamıştı! *Alt ve üst yapı izin verdiği ölçüde hızlanma mevcut.*



Çözüm Karmaşıklığı

Varsayılan kabuk sistemine alternatif çözümler! *`git`, `pip` gibi komutlarda şaşırtıcı sonuçlar.*



Geri Dönüş Tasarrufu

Kayıt dosyaları temelli ve daha anlamlı! *Önceki komutlara geri dönüşte varsayılan kabuk sistemlerine göre daha tasarruflu denilebilir.*

Sorunlar ve Çözümleri

Ayrıştırma

Komut girdilerinin ayrıştırılması. Varsayılan kabuk sistemi korunarak, operatörler yardımıyla komutlar birbirinden ayrıştırılmıştır. CMD sisteminde `&` operatörü kullanılırken, BASH sisteminde `;` kullanılmıştır.

Dosya Yolu

`Change Directory` olarak bilinen ve kabuk sistemlerinin vazgeçilmez komutlarından `cd` sorunları optimize edilmiştir. Çoklu iş parçacığı bu komut için aktif değildir.

Hata Yönetimi

Yazılımsal hata yönetimlerine ek, zaman karmaşıklığının yarattığı hatalar da oluşmuştur. Çoklu iş parçacığı sistemlerinde sık karşılaşılan, istenen çıktının düzeyinin belirlenmesi sorunları çözülmüştür.

Sorunlar ve Çözümleri

LOG

Multithreading sistemlerde karmaşık logging işlemleri olabilir; kabuk sistemlerinde kritik öneme sahip log parçaları, her kabuk için ayrı tutulur.

Varlıklar

Çok katmanlı projelerde, varlıklar ve özellikler belirsiz olduğundan, komutun uygulanmasından son çıktının alınmasına kadar olan süreçte sorumluluk üstlenmeliyiz.

Kurallar Bütünü

Kuralların komut özelinde değil, komutların sınıflarına göre olması gerekmektedir. Bu sebeple kurallar bütünü karmaşıklığı azaltılmıştır.

Varsayılan CMD

Zamana bağlı komutlardan `ping` kullanımıyla ilgili kısa bir örnek.



cmd.exe

```
C:\>ping -n 10 google.com
```

```
Pinging google.com [172.217.17.110] with 32 bytes of data:
```

```
Reply from 172.217.17.110: bytes=32 time=16ms TTL=56
```

```
Reply from 172.217.17.110: bytes=32 time=16ms TTL=56
```

```
Reply from 172.217.17.110: bytes=32 time=16ms TTL=56
```

```
Reply from 172.217.17.110: bytes=32 time=16ms TTL=56
```

```
Reply from 172.217.17.110: bytes=32 time=17ms TTL=56
```

```
Reply from 172.217.17.110: bytes=32 time=17ms TTL=56
```

```
Reply from 172.217.17.110: bytes=32 time=17ms TTL=56
```

```
Reply from 172.217.17.110: bytes=32 time=16ms TTL=56
```

```
Reply from 172.217.17.110: bytes=32 time=16ms TTL=56
```

```
Reply from 172.217.17.110: bytes=32 time=16ms TTL=56
```

```
Ping statistics for 172.217.17.110:
```

```
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 16ms, Maximum = 17ms, Average = 16ms
```

Süre: 9.87 saniye

Multithreading Shell

Aynı örneğin projede çalıştırılması.



`mtcmd.exe`

```
C:\>ping -n 10 google.com & ping -n 10 youtube.com
```

```
Pinging google.com [172.217.17.110] with 32 bytes of data:  
...
```

```
Ping statistics for 172.217.17.110:
```

```
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 16ms, Maximum = 17ms, Average = 16ms
```

```
Pinging youtube.com [142.251.141.46] with 32 bytes of data:  
...
```

```
Ping statistics for 142.251.141.46:
```

```
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 16ms, Maximum = 20ms, Average = 17ms
```

Süre: 10.02 saniye

Varsayılan CMD

Ağ sistemi yolu izleme üzerine örnek.



cmd.exe

```
C:\>tracert google.com
```

```
Tracing route to google.com [172.217.20.78]  
over a maximum of 30 hops:
```

1	<1 ms	<1 ms	<1 ms	HGW.LOCAL [192.168.1.1]
2	4 ms	2 ms	1 ms	172.17.1.42
3	4 ms	3 ms	4 ms	85.32.146.159.srv.turk.net [159.146.32.85]
4		1 ms	1 ms	158.122.223.31.srv.turk.net
				[31.223.122.158]
5		4 ms	6 ms	161.122.223.31.srv.turk.net
				[31.223.122.161]
6	30 ms	19 ms	19 ms	42.77.146.159.srv.turk.net [159.146.77.42]
7	16 ms	21 ms	19 ms	103.244.50.221
8	19 ms	25 ms	30 ms	193.218.0.113
9	17 ms	16 ms	17 ms	108.170.250.161
10	17 ms	16 ms	16 ms	142.250.56.111
11	16 ms	15 ms	15 ms	sof02s49-in-f14.1e100.net [172.217.20.78]

```
Trace complete.
```

Süre: 16.27 saniye

Multithreading Shell

Aynı örneğin projede çalıştırılması.



`mtcmd.exe`

```
C:\>tracert google.com & tracert youtube.com
```

```
Tracing route to google.com [172.217.20.78]  
over a maximum of 30 hops:
```

```
  1    <1 ms    <1 ms    <1 ms  HGW.LOCAL [192.168.1.1]  
...  
Trace complete.
```

```
Tracing route to youtube.com [216.58.213.110]  
over a maximum of 30 hops:
```

```
  1     1 ms    <1 ms    <1 ms  HGW.LOCAL [192.168.1.1]  
...  
Trace complete.
```

Süre: 16.54 saniye

Log Dosyası

Aynı örneğin projede çalıştırılması.



CMDLogger.txt

```
Command ID: CMD-3a0c8aed-32f3-4595-9609-06b69b6847b7
Command Date: 7.04.2023 19:44:05
Command Content: ping -n 10 google.com,ping -n 10 youtube.com
Result ID: CMD-1e0b3184-f786-4b76-bea3-6e08e95cb002
Result Content:
Pinging google.com [216.58.212.14] with 32 bytes of data:
...
Ping statistics for 216.58.212.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 15ms, Maximum = 15ms, Average = 15ms

Pinging youtube.com [142.251.140.14] with 32 bytes of data:
...
Ping statistics for 142.251.140.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 15ms, Maximum = 15ms, Average = 15ms
```

Result Running Time: 3,119 saniye

işleyiş Süreci



Shell Type

Komut istemi tipinin
(CMD/Bash)
belirlenmesi



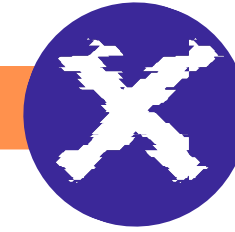
Command Division

Girdi komutlarının
ayrıştırılması



Command Type

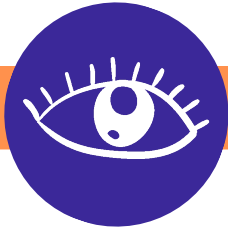
Komutun hangi tipte
olduğunun
belirlenmesi



Executing

Havuz yöntemiyle
komutların/komutun
çalıştırılması

işleyiş Süreci



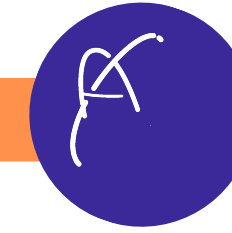
Result

Sonuçların çalışmaya
uygun formatta
gösterilmesi



Logging

Komut setine göre
çevrimdışı log sistemi



Caching

Sonucu ve log kayıtları
doğru olan
komutların ön
belleğe alınması

Avantajları

1 Ağ temelli tüm komut setlerinin daha **hızlı** ve **optimize** çalışmaması. Bu sayede büyük ölçekli komut setlerinin süre açısından avantajlı bir hale gelmesi mümkündür. Fakat internet hızının getirdiği kısıtlar bu avantajı yok sayabilir. Bu durum **sunucu port kontrollerinde** hızı arttırır.

2 **Kritik alan** uyarıları sayesinde aynı sistem kaynaklarının kullanımı minimize edilir. Doğru çalışmış komutlar ön bellekte olmasından ötürü tekrar çalıştırılınca **minimum sistem gereksinimi** ile çalışır.

3 **CMDLogger** gibi hazır programların güvenlik açıkları mevcut. Özellikle siber güvenlik alanında çalışanlar için önemli bir detaydır. Log sistemini **çevrimdışı ve yerel dosyalama sisteminde** bulundurmak güvenliği arttırmaktadır. Ayrıca komutlara ait **tarih** ve **ID** değerleri sayesinde listeleme işlemleri kolaylaşır.

Dinlediğiniz için teşekkürler!

Projeye ait tüm bilgiler için

<https://github.com/ahmetmungan/MultithreadingShell>

ziyaret etmeyi unutmayınız!

Ahmet MUNGAN

160255081

Kırıkkale Üniversitesi Bilgisayar Mühendisliği