

## Gerçek Zamanlı İşletim Sistemi Kavramları

Bu ders, gerçek zamanlı işletim sistemi (RTOS) kavramlarına odaklanarak Linux'ın gerçek zamanlı çekirdek ve ilgili araçlarla nasıl kullanıldığını detaylı bir şekilde ele alacaktır.

### 1. Linux'ta Gerçek Zamanlı Çekirdek (PREEMPT-RT)

#### Avantajları ve Dezavantajları

##### Avantajları:

- **Düşük Gecikme:** Daha hızlı ve tahmin edilebilir tepki süreleri sağlar.
- **Esneklik:** Hem gerçek zamanlı hem de geleneksel iş yükleri için uygundur.
- **Topluluk Desteği:** Sürekli geliştirilen ve desteklenen açık kaynak bir projedir.

##### Dezavantajları:

- **Performans Kayıpları:** Yüksek preemption seviyesi nedeniyle, bazı geleneksel iş yüklerinde performans düşüşü yaşanabilir.
- **Uyumluluk Sorunları:** Tüm donanımlar veya sürücüler RT yaması ile uyumlu olmayabilir.
- **Yapılandırma Zorluğu:** Yüksek seviyede uzmanlık gerektirebilir.

PREEMPT-RT'nin bu avantaj ve dezavantajları, kullanım senaryolarına bağlı olarak dikkatlice değerlendirilmelidir.

Gerçek zamanlı Linux, belirli işlerin belirli süreler içinde tamamlanmasını garanti eden bir sistemdir. PREEMPT-RT, Linux'ı gerçek zamanlı hale getirmek için geliştirilmiş bir yama setidir.

#### Temel Kavramlar

- **Deterministik Davranış:** Sistem, her zaman tahmin edilebilir sürelerde yanıt verir.
- **Preemption (Kesme):** İşlemcide çalışan bir işlem, daha yüksek öncelikli bir işlem geldiğinde kesilebilir.
- **Kernel Space vs User Space:** Gerçek zamanlı işler genellikle kernel seviyesinde gerçekleştirilir.

#### PREEMPT-RT Kullanımı

PREEMPT-RT, standart Linux çekirdeğine eklenerek daha iyi bir zamanlama ve daha düşük gecikme sağlar. PREEMPT-RT'yi etkinleştirmek için aşağıdaki adımlar izlenebilir:

##### 1. PREEMPT-RT Yamalarını Yükleyin:

- PREEMPT-RT yamalarını Linux çekirdeği ile uyumlu olarak indirin:
- wget  
<https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/<version>/patch-<version>-rtXX.patch>

##### 2. Linux Çekirdeği Kaynak Kodunu İndirin:

- Çekirdek kaynak kodunu alıp yama ile entegre edin:
- `git clone https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git`
- `cd linux`
- `patch -p1 < ../patch-<version>-rtXX.patch`

### 3. Konfigürasyon Yapın:

- "make menuconfig" ile yapılandırma ekranını açın ve "Fully Preemptible Kernel (RT)" seçeneğini etkinleştirin:
- `make menuconfig`

### 4. Çekirdeği Derleyin ve Yükleyin:

- Yeni çekirdeği derleyip sisteme yükleyin:
- `make -j$(nproc)`
- `sudo make modules_install`
- `sudo make install`

### 5. Sistemi Yeni Çekirdek ile Başlatın:

- Sistem başlatma yöneticisinde (GRUB) yeni çekirdeği seçerek yeniden başlatın.

### 6. Doğrulama Yapın:

- Çekirdeğin yüklendiğini doğrulamak için:
- `uname -r`

PREEMPT-RT içeren bir sürüm numarası görmelisiniz. standart Linux çekirdeğine eklenerek daha iyi bir zamanlama ve daha düşük gecikme sağlar.

## 2. Gerçek Zamanlı Linux Ayarları ve Kullanım Alanları

### Kullanım Alanları

- **Endüstriyel Otomasyon:** Robotik sistemler, üretim hatları ve sensör bazlı uygulamalar için gerçek zamanlı Linux kullanılabilir. Örneğin, bir otomotiv fabrikasında robotik kolların hassas bir şekilde senkronize çalışması gereklidir.
- **Ses ve Video Akışı:** Canlı yayın platformlarında veya ses işleme yazılımlarında düşük gecikmeli medya işleme önemli bir rol oynar. Örneğin, müzik prodüksiyonunda gecikmesiz bir ses kaydı sağlanabilir.
- **Medikal Cihazlar:** Kalp atış monitörleri gibi anlık veri işleme gereksinimi olan cihazlar gerçek zamanlı Linux'a dayanır.
- **Telekomünikasyon:** Baz istasyonlarının veri paketlerini işlemeye yönelik düşük gecikme gereksinimleri için.

- **Havacılık ve Uzay:** Uçak kontrol sistemlerinde veya uydu iletişimlerinde gerçek zamanlı Linux tercih edilir. Örneğin, uçak içi eğlence sistemleri.

### Temel Ayarlar

- **CPU Frekans Sabitleme:** Gücünüzü maksimum performans için ayarlayın:  
Örnek; `sudo cpufreq-set -g performance`
- **Öncelikli Görevler:** Belirli bir süreci gerçek zamanlı yapmak:  
Örnek; `chrt -f 99 <PID>`

### 3. Real-Time Süreçlerin Planlanması ve Önceliklendirilmesi

Linux'ta süreç planlama, gerçek zamanlı öncelikler kullanılarak yapılabilir.

#### Planlama Politikaları

- **SCHED\_FIFO:** İlk giren ilk çıkar.
- **SCHED\_RR:** Yuvarlak robin.

#### Örnek Kod

Bir sürecin planlama politikasını kontrol etme ve değiştirme:

```
import os
import subprocess

# Süreç ID'sini al
pid = os.getpid()

# Süreç önceliğini belirle
subprocess.run(["chrt", "-f", "99", str(pid)])
```

#### 4. RT Çekirdek Derleme ve Yükleme

PREEMPT-RT yamalı bir çekirdek derlemek için adımlar:

##### Adımlar

1. **Kaynak Kodlarını Alın:**
2. `git clone https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git`
3. **PREEMPT-RT Yamasını Ekleyin:**
4. `patch -p1 < patch-<version>.patch`
5. **Çekirdeği Derleyin ve Kurun:**
6. `make menuconfig`
7. `make -j$(nproc)`
8. `sudo make modules_install`
9. `sudo make install`

##### Sık Karşılaşılan Sorunlar ve Çözümleri

1. **"Hatalı Yama Uygulaması":**
  - **Sorun:** Yama uygularken "hunk failed" hatası alabilirsiniz.
  - **Çözüm:** Yamayı doğru sürümle eşleştirdiğinizden emin olun. Ayrıca yamayı uygulamadan önce "git status" ile dosya durumunu kontrol edin.
2. `git reset --hard`
3. `patch -p1 < patch-<version>.patch`
4. **"Derleme Hataları":**
  - **Sorun:** Derleme sırasında "missing dependency" hataları ile karşılaşabilirsiniz.
  - **Çözüm:** Tüm gerekli paketlerin kurulu olduğundan emin olun:
  - `sudo apt-get install build-essential libncurses-dev bison flex libssl-dev`
5. **"Çekirdeğin GRUB'da Görünmemesi":**
  - **Sorun:** Yeni çekirdek GRUB menüsünde görünmüyorsa.
  - **Çözüm:** GRUB'u güncelleyin ve yeniden başlatın:
  - `sudo update-grub`
  - `sudo reboot`
6. **"Sistem Çökmesi":**
  - **Sorun:** Yeni çekirdekle sistem başlatılamıyorsa.
  - **Çözüm:** GRUB üzerinden eski bir çekirdeği seçerek başlatın ve yeni çekirdeğin yapılandırmasını gözden geçirin.

## 5. Performans Ölçüm Araçları ve Gecikme Testi

### Gecikme Testi İçin "cyclicttest"

"cyclicttest" aracı, sistemin gerçek zamanlı görevler için ne kadar hızlı tepki verebildiğini ölçmek için kullanılır.

#### Kurulum

"cyclicttest" aracı aşağıdaki gibi kurulabilir:

```
sudo apt install rt-tests
```

#### Kullanım

Temel bir gecikme testi için örnek komut:

```
sudo cyclicttest -t1 -p99 -n -i100 -l10000
```

Bu komut:

- **-t1**: Tek bir iş parçası oluşturur.
- **-p99**: İşlem önceliğini 99 olarak ayarlar (yüksek öncelik).
- **-n**: Gerçek zamanlı modda çalışır.
- **-i100**: İşlem aralığını 100 mikro saniye olarak belirler.
- **-l10000**: 10.000 döngü boyunca test yapar.

#### Çıktı Örneği

```
# /dev/cpu_dma_latency set to 0us
```

```
policy: fifo: loadavg: 0.00 0.00 0.00 1/80 3442
```

```
T: 0 ( 3442) P:99 I:100 C: 10000 Min:   5 Act: 10 Avg: 12 Max:  25
```

- **Min**: Minimum gecikme (mikrosaniye).
- **Act**: Son ölçülen aktif gecikme.
- **Avg**: Ortalama gecikme.
- **Max**: Maksimum gecikme.

#### Sonuçların Yorumlanması

- **Min ve Avg Değerleri**: Sisteminizin normal çalışma koşullarında gecikme performansını gösterir. Düşük değerler, iyi bir performansı ifade eder.
- **Max Değeri**: Nadiren meydana gelen en kötü senaryodaki gecikmeyi temsil eder. Bu değer, gerçek zamanlı görevlerin kritik zaman sınırlarını aşıp aşmayacağını belirlemek için önemlidir.
- **Öneri**: Max değeri yüksekse, sisteminizde aşağıdaki optimizasyonları gözden geçirin:

1. **Gerçek Zamanlı Ayarları Kontrol Edin**: CPU frekans sabitlemesi veya öncelikli işlem ayarları yapılmış mı?

2. **Arka Plan İşlemlerini Azaltın:** Sistem kaynaklarını tüketen gereksiz hizmetleri devre dışı bırakın.
3. **Donanım Uyumunu Gözden Geçirin:** RT çekirdek ile kullanılan donanımın gerçek zamanlı performansı desteklediğinden emin olun.

### **Daha İleri Testler**

"cyclictest" çıktısının görselleştirilmesi veya loglanması daha ayrıntılı analiz için değerlendirilebilir. Örneğin:

```
sudo cyclictest -t2 -p90 -n -i200 -l5000 > cyclictest_output.txt
```

### **Gecikme Testi İçin "cyclictest"**

"cyclictest" aracı gecikmeyi ölçmek için kullanılabilir.

### **Kurulum**

```
sudo apt install rt-tests
```

### **Kullanım**

```
sudo cyclictest -t1 -p99 -n -i100 -l10000
```

Bu komut, 10.000 tekrar için gecikme ölçümü yapar.