Çok İşlemci Üzerinde Çalışan Yazılımlar için Hata Yönetim Altyapısı

Mustafa YAMAN

Mikrodalga ve Sistem Teknolojileri (MST) Grubu,

ASELSAN AŞ, Ankara

e-posta: yaman@aselsan.com.tr

Özetçe

ASELSAN MST grubunda radar ve elektronik harp projelerindeki gömülü uygulamalar için çok işlemcili donanım mimarileri kullanılmaktadır. Bu donanımlar üzerinde koşan yazılımların hatasız çalışması çok önemlidir. Tasarım aşamasında alınan önlemlere rağmen yazılım veya donanım kaynaklı çıkabilecek hataların algılanması, teşhisi, düzeltilmesi ve raporlanması için bu yazılımlar ile birlikte çalışabilecek ve yazılımların yapısını değiştirmeyecek Hata Yönetim amaçlı yazılım modülleri tasarlanmaktadır. Bu bildiride çok işlemcili yapılarında kullanılabilen ve performansı olumsuz yönde etkilemeyen bir Hata Yönetim altyapısı anlatılmaktadır.

1. Giriş

Sistemlerde yazılım kaynaklı hataların oluşmasını engellemek için yazılım geliştirme süreçleri, yazılım mimarileri ve tasarımları geliştirilmektedir. Ayrıca kullanıcı gereklerinin iyi analiz edilmesiyle ve anlaşılır kodlar yazmakla hataların oluşması engellenmeye çalışılmaktadır [1]. Bu çabalarla, yazılım kaynaklı hatalar azaltılabilse bile tamamen engellenememektedir.

Yazılım güvenilirliğinin ölçüsü, yazılımın herhangi bir hata durumuna düşmeden tanımlanan görevlerini yerine getirme olasılığıdır. Bu olasılığı artırmak ve daha güvenli yazılımlara ulaşmak için çeşitli geliştirme süreçleri, tasarım, kodlama ve test teknikleri geliştirilmektedir. Hata durumuna düşen bir yazılımın kendini toparlayabilmesi de önemli bir ölçüttür. Kendini toparlayabilme, yazılımın ciddi ve geçiştirilemeyen bir hataya düşmesi durumunda sıfırlanıp yeniden başlatılması, ilklendirilmesidir. Kendini toparlayabilme konusunda, yazılımın kaldığı yerden çalışmaya devam etmesi hedeflenmektedir.[2]

Çalışma sırasında oluşabilecek hataları algılayıp, sistemin çalışmasını etkilemeden önlem alabilen kısacası kendini toparlayabilen, hata yönetim yapılarına ihtiyaç duyulmaktadır. Sistemlerde hatanın yönetilmesi 5 aşamada gerçekleştirilmektedir [3]. Bu aşamalar şunlardır:

- Tespit: Tespit aşamasında, hatalar bulunur ve hata yöneticisine iletilir. Hatanın teşhisi ile ilgili herhangi bir işlem yapılmaz.
- Teşhis: Teşhis aşamasında, hatanın nereden kaynaklandığı ve hangi birimleri etkilediği belirlenir.
- İzolasyon: İzolasyon aşamasında hatanın sistemin çakılmasına yol açmamasını sağlayacak önlemler alınır. İzolasyon ile sistemin tam olarak çalışması garanti edilmez.
- Giderme-Toparlanma: Giderme-Toparlanma aşamasında sistemin beklenen davranışına dönmesi sağlanır.
- Onarma: Onarma aşamasında, sistemin tüm artık birimleri ile birlikte tüm kabiliyetlerine ulaşması sağlanır.

Yukarıda anlatılan aşamalarda her işlemde bir üst seviyeye veya log sunucusuna raporlama yapılmaktadır.

ASELSAN MST grubunda geliştirilen ve bu bildiride anlatılan hata yönetim tasarımında, hatanın algılanması, raporlanması ve sistemin kısa süre içinde kaldığı yerden devam edecek şekilde yeniden başlatılması hedeflenmiştir. Çoğu sistemde donanım arızası durumunda bu arızayı tolere edecek yedek donanım bulunmamaktadır. Hata Yönetim tasarımında sadece yazılım kaynaklı hatalara ve anlık donanım hatalarına önlem alınabildiği düşünülmektedir. Bu sebepten dolayı kalıcı donanım arızaları bu bildiri kapsamında değildir.

Radar ve Elektronik Harp Projelerindeki gömülü uygulamalarda performans ihtiyaçlarından dolayı çok işlemcili yapılar kullanılmaktadır. Projelerin ihtiyaçlarına göre üzerinde gömülü yazılımlar koşan işlemci sayısı değişmektedir. Bildirinin ilk bölümünde Çok İşlemcili Mimari ile bu yazılımlarla birlikte çalışmak üzere ihtiyaç duyulan Hata Yönetim yapısının gerekliliği anlatılacaktır. İkinci bölümde, çok işlemcili mimari üzerinde koşan Hata Yönetim tasarımı detaylandırılıp gömülü yazılım mimarisindeki yerinden bahsedilecektir. Üçüncü bölümde örnek senaryolar üzerinden işleyiş aktarılacaktır.

2. Hata Yönetim Gerekleri

Hata Yönetim gereklerini tartışmadan önce mevcut çok işlemcili mimariyi incelemek gerekmektedir.

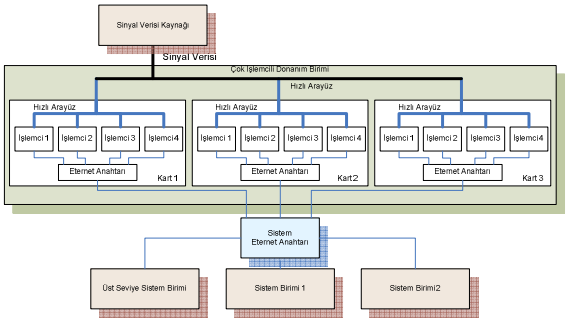
2.1. Çok İşlemcili Mimari

Örnek birçok işlemcili donanım mimarisi Şekil 1'de gösterilmiştir. Şekilde çok işlemcili donanımın yanı sıra diğer sistem birimleri için örnek de verilmiştir. Çok işlemcili donanım mimarisinde bulunan özelikler şunlardır:

- Tüm işlemcilerin birbirine erişmesini sağlayan ve hızlı ve yoğun veri alışverişine izin veren hızlı ara yüz bulunmaktadır.
- Tüm işlemcilerin ethernet ara yüzü bulunmaktadır.
- Tüm işlemciler birbirleriyle ve diğer sistem birimleri ile UDP veya TCP/IP protokolü ile haberleşebilmektedir.

Hızlı ara yüz özellikleri şunlardır:

- Çoğunlukla sinyal işleme amaçlı kullanılan çok işlemcili yapıda, sinyal kaynağından gelen sinyal verileri işlemciler tarafından seri hızlı veri yolu ile alınmaktadır.
- İşlemciler arasında da Ethernet arayüzü dışında hızlı haberleşme için arayüz bulunmaktadır.



Şekil 1: Çok İşlemcili Donanım Mimarisi

2.2. Hata Yönetim Gerekleri

Hata Yönetimi tasarımında temel alınan gerekler aşağıdaki gibi tanımlanmıştır. (Şekil 2)

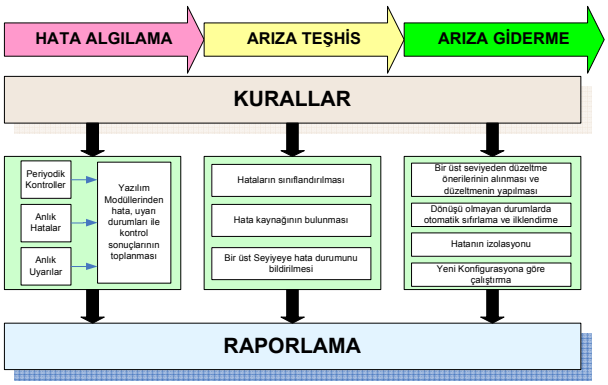
- Hata Yönetim kapsamındaki yazılım modüllerinin modüler olması
- İşlemciler üzerinde yerel olarak bulunacak Hata Yönetim modüllerinin, belirlenen Sinyal İşleme Mimarisine göre tasarımı yapılmış yazılımlara entegrasyonunda, yazılımları etkilememesi
- Çok işlemcili yapılarda farklı noktalarda oluşan hata durumlarının tek bir merkezde toplanması
- Hata Yönetimle ilgili her adımda Log sunucusuna açıklayıcı bilgilerin kayıt amaçlı gönderilmesi
- Hata durumlarını bildiremeyen işlemcilerin son hata durumlarını kalıcı belleklerine yazması
- Gelen hataların sınıflandırıp yorumlaması ve gerekiyorsa bir üst seviye sistem birimine durumun bildirilmesi
- Hata durumlarına göre, işlemcilerin veya kartların sıfırlaması ve ilklendirmesi
- Hata durumuna göre konfigürasyon değişikliği yapılması ve bu konfigürasyona göre tüm yazılım modüllerinin çalıştırılması
- Hata Yönetim, hata algılama, arıza teşhis, arıza düzeltme ve raporlama işlevlerinin yerine getirilmesi

3. Hata Yönetim Tasarımı

3.1. Yazılım Mimarisi

İşlem yoğun sinyal işleme uygulamalarında çoğunlukla çok işlemcili yapılar kullanılmaktadır. Hata Yönetim tasarımı, Şekil 3'de gösterilen, ASELSAN MST gurubunda geliştirilen Sinyal İşleme Yazılımları Mimarisi [4] içine yerleşebilecek şekilde tasarlanmıştır.

Şekil 3'deki Sinyal İşleme Yazılım Mimarisinde, İşletim Sistemi, Soyutlama, Servis, Uygulama ve Algoritma katmanları ile tüm katmanlara hizmet verebilen Destek katmanı bulunmaktadır. Ayrıca sistemdeki işlemci, veriyolu benzeri tüm donanım bileşenlerini temsil eden ve herhangi bir yazılım barındırmayan bir Donanım katmanı tanımlanmıştır. [4]



Şekil 2: Hata Yönetim Gerekleri

Mimaride, Hata Yönetim ile ilgili modüller Uygulama ve Destek Katmanlarında yer almaktadır.

ASELSAN MST gurubunda Sinyal İşleme Yazılım Mimarisine uygun olarak sistem yazılım modülleri tasarlanmaktadır. Yazılım modüllerinin işlemciler üzerine yerleşimi analiz çalışmaları sonucunda belirlenmektedir. Tasarımlarda, performans kriterleri göz önünde bulundurularak modüllerin işlemciler üzerine dağıtılmasının dinamik olarak belirlenmesi hedeflenmektedir.

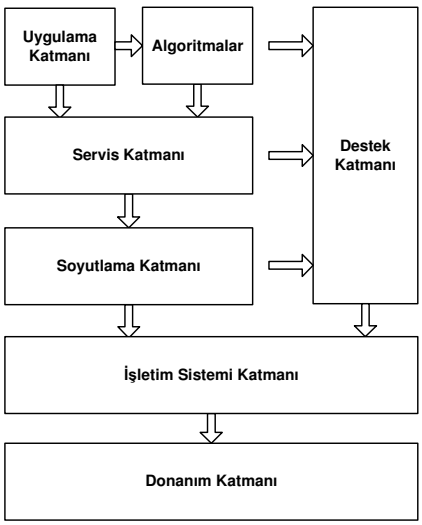
3.2. Hata Yönetim Tasarımı

Şekil 4'de gösterildiği gibi Hata Yönetim tasarımı ile birlikte tüm işlemcilerde Hata Kontrol Modülü olarak adlandırılan bir modül çalışmaktadır.

Hata Kontrol Modülleri, bulundukları işlemcide çalışan diğer modüller ile ilgili durumları ve işlemcinin durumunu monitör etmek üzere tasarlanmıştır. Hata Kontrol modülü jenerik tek bir modüldür ve çalıştığı işlemcideki yazılım modüllerini bir konfigürasyon dosyasından okuyarak veya yazılım modüllerinin kendisine kayıt yaptırmasıyla çalıştığı işlemci hakkında bilgi edinmektedir.

Çok işlemcili yapıda yer alan bir işlemcide veya ethernet ile bağlı bulunan başka bir sistem biriminde çalışmak üzere Hata Yöneticisi tasarlanmıştır. Hata Yöneticisi, çok işlemcili alt sistemde yer alan bir işlemcide çalışabileceği gibi sistemde yer alan Sistem Yöneticisinde veya kullanıcı terminalinde de çalışabilmektedir.

Bu bölümde, Giriş bölümünde bahsedilen Hata Yönetimi aşamalarının tasarımda nasıl gerçeklendiği anlatılacaktır.



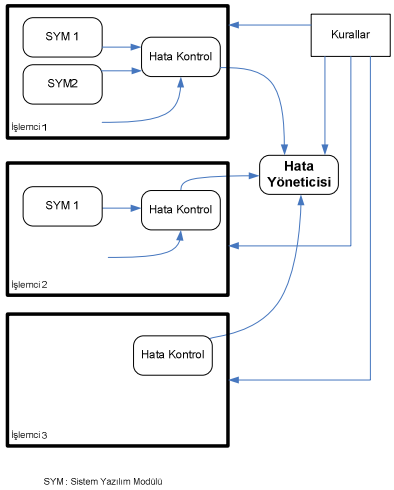
Şekil 3: Yazılım Mimarisi

3.2.1. *Hata Algılama*

Hata Yönetiminin ilk aşaması oluşan hatanın algılanmasıdır. İşletim sistemlerinde, yazılım kütüphanelerinde, yazılım ve donanım sürücülerinde yer alan fonksiyonlarda hata durumlarında mutlaka hata dönüşleri yapılmaktadır. Dönen bu hatalar Hata Yöneticisine aktarılmaktadır. Bunun dışında yazılımlar içindeki sınır kontrollerinde, gelen mesajların kontrolünde veya tutarlılık kontrollerinde oluşan hata veya uyarı durumlar da Hata Yöneticisine bildirilmektedir.

Hata Kontrol Modülü çalıştığı işlemcide çalışan tüm yazılımlardan haberdar olmalı ve durumlarını periyodik olarak sorgulayıp Hata Yöneticisine bildirmektedir. Hata Kontrol Modülü, işletim sistemi yeteneklerini kullanarak işlemcide çalışan görevlerin durumlarını sorgulamaktadır. Sistemde çalışan yazılım modüllerinde hataların ne olduğu ve hangi durumlarda yöneticiye haber verilmesi gerektiği kurallar dosyasında bulunmaktadır. Açılışta kurallar dosyası hata kontrol modülü tarafından okunmaktadır.

Sonuç olarak çok işlemcili yapıdaki işlemcilerde çalışan tüm modüllerdeki hatalar, uyarılar ve durumlar Hata Yöneticisinde toplanmaktadır.



Şekil 4: Çok İşlemcili Yapıda Yazılım Modülleri

3.2.2. *Arıza Teşhis-Arıza Giderme*

Hata Yöneticisi, işlemcilerde koşan yazılımlar tarafından algılanan tüm hataları, uyarıları ve çalışma durumlarını değerlendirerek ne yapılması gerektiğine karar vermektedir. Kararları verirken kurallar dosyasından yararlanmaktadır. Kurallar dosyasında aşağıdaki bilgiler yer almaktadır.

- Hata ve uyarıların gruplandırılması,
- Hata durumlarının arızalarla eşlenmesi
- Arıza durumunda arıza çeşidine göre izole edilecek birimler
- Arızalar karşısında uygulanacak arıza giderme yöntemleri
- Arıza durumunda bir üst seviye sistem birimine bildirilecek mesaj

Hata Yöneticisi gelen hata ve uyarılara göre kurallar dosyası içeriğine göre aşağıdaki işlemleri uygulayabilir.

- Operatöre gösterilmek üzere hata ve uyarıların terminale gönderilmesi
- Bir üst seviye sistem birimine sistemin kapatılması için mesaj gönderilmesi
- Problemli kartların sıfırlanması ve sistemin çalışmasını etkilemeden yeniden çalışmaya başlaması (izolasyon)
- Problem olan yazılım modülünün yeniden başlatılması (Özel şartlarda geçerli olabilir) (izolasyon)
- Problem olan işlemcinin yeniden başlatılması (Özel şartlarda geçerli olabilir) (izolasyon)

3.2.3. Raporlama

Hata Yönetimin önemli aşamalarından biri de oluşan hataların, yapılan teşhis ve düzeltmelerin raporlanmasıdır. Raporlama iki şekilde yapılabilir: Bir üst seviye sistem birimine bilgilerin aktarılması, bilgilerin bir Log Sunucusuna aktarılması. Raporlama için bu yöntemlerden biri veya her ikisi de kullanılabilir.

Sistemde oluşan hataların sonra analiz edilebilmesi için log tutma işlevini kullanmak faydalı olacaktır.

Hata Yönetim tarafından raporlama yapılamadığı durumlarda hata bilgilerinin kaybolmaması için de önlem alınmaktadır. İşlemci kartlarında bulunan ve kapatılıp açılma ile silinmeyen bellek alanları bu bilgilerin kaydedilmesi için kullanılabilmektedir. Buradaki amaç, hata oluşması durumunda bir düzeltme yapıldıysa ve bu durum raporlanamadıysa bir sonraki problemsiz açılışta durumun raporlanmasıdır.

4. Hata Yönetim İşleyişi

Hata Yönetim yapısının işleyişi iki başlık altında anlatılacaktır: Açılış İşlemleri ve Normal Çalışma Modu.

4.1. Açılış İşlemleri

Açılışta, merkezi bir diskte sistem konfigurasyon parametreleri ve işlemcide çalışacak modülleri tanımlayan dosyalar bulunmaktadır. Çok işlemcili yapıda koşan tüm yazılım modülleri bu dosyalara göre işlemcilere yüklenmekte ve yine bu dosyalarda yer alan parametreleri kullanarak çalışmaktadır. Açılışta sırasıyla aşağıdaki işlem adımları gerçekleştirilmektedir.

- Tüm işlemcilerde Hata Kontrol modülü çalışır. Hata Kontrol modülü, bulunduğu işlemcideki diğer yazılım modüllerine Hata Yöneticisine ve Log Sunucusuna erişim hizmeti verir.
- Modüllerdeki açılış işlemleri tamamladığında Hata Yöneticisine rapor gönderilir.
- Hata Kontrol İşlevleri
    - Her işlemcide bir Hata Kontrol modülü çalışır.
    - Hata Kontrol modülü Hata Yöneticisi ile bağlantı kurar.
    - Çalıştığı işlemcideki diğer modüllerin durumunu periyodik olarak kontrol eder ve Hata Yöneticisine raporlar. (Hata olmasa da Kalp Atışı Mesajı amaçlı mesaj gönderilir)
    - Hata Kontrol modülü bir önceki çalışmada bildirilememiş hataları Hata Yöneticisine raporlar.
- Hata Yöneticisi İşlevleri
    - Belirli bir zaman içinde tüm Hata Kontrol modüllerinin bağlanmasını ve raporların tamamlanmasını bekler.
    - Tamamlanmadığı durumda hata teşhis ve düzeltme işlemlerini gerçekleştirir.
    - Bir önceki çalışmada raporlanamamış hataları raporlar.

Açılış sırasında yukarıdaki işlevler gerçekleştirildikten sonra normal çalışma moduna geçilir.

4.2. Normal Çalışma Modu İşlemleri

Normal çalışma modu, açılışını tamamlayan tüm modüllerin bulunduğu modu anlatmaktadır. Çalışma modunda çıkan tüm hatalar ve uyarılar Hata Yöneticisine bildirilir.

Normal Çalışma modunda Hata Kontrol İşlevleri:

- İşlemcilerde çalışan görevlerin sürekli kontrolünü yapar.
- Hata bulunsun veya bulunmasın Hata Yöneticisine periyodik raporlama yapar
- Hata Yöneticisine raporlanamayan hataları saklar

Normal Çalışma modunda Hata Yöneticisi İşlevleri:

- Gelen hatalara göre kurallar dosyasında tanımlanmış kararları verir.
- Düzeltilemeyecek hata oluştuğu durumları bir üst seviye sistem birimine bildirir.
- Raporlayamadığı durumları saklar.

Hata durumunda izolasyon yapılıp sadece sistemin belirli bir biriminin sıfırlanması gerekiyorsa bir takım önlemler alınmalıdır. Alt birimler bir üst seviye sistem biriminden aldığı parametrelere göre çalışmaktadır. Sıfırlama sonrası kaldığı yerden çalışmaya devam edebilmesi için aldığı tüm parametreleri enerji kesilmeden yapılan sıfırlama sonrası bozulmayacak bir bellek alanında saklamalıdır. Enerji kesilmeden yapılan sıfırlama sonrası bu veriler okunarak, çalışmaya devam edilebilmektedir.

5. Sonuçlar

Bu bildiride, Çok İşlemci Üzerinde Çalışan Yazılımlar için Hata Yönetim tasarımı ve işleyişi anlatılmıştır. Hata Yönetim tasarımı için öncelikle gerekler belirlenmiş ve çok işlemcili donanım mimarisi göz önünde bulundurulmuştur. Donanım sağladığı avantajlar da kullanılarak hata oluştuğunda algılayabilecek, teşhis yapabilecek ve düzeltmeleri uygulayabilecek bir yapı tasarlanmıştır. Tasarımda, hata algılamadan düzeltmenin yapılmasına kadarki tüm aşamalarda sistemde bulunan ilgili birimlere raporlama yapılması hedeflenmiştir. Raporlama yapılamadığı durumlarda da bir sonraki problemsiz açılışta durumun raporlanması sağlanmıştır. Sistemin hata sonucunda düzeltmenin ardından kaldığı yerden çalışmaya devam edebilmesi için de gerekli önlemler alınmıştır.

Hata Yönetim tasarımı ile kazanılan en önemli yeteneklerden biri de hata raporlarını kullanarak yazılımın güvenilirliğinin ölçülebilmesidir.

Hata yönetim yapısının uygulandığı projelerde yazılım geliştirme aşamasında da büyük fayda sağlanmıştır. Çok işlemcili mimaride koşan yazılım modüllerinin durumları hakkında sürekli bilgiler toplamak ve hataların monitör edilmesi yazılım kaynaklı hataların bulunmasında katkı

sağlamaktadır. Aynı zamanda birçok modülden yakın zamanlarda çıkan hataların toplanması hatalar arasındaki ilişkilerin tespitinde önemli rol oynamaktadır.

6. Kaynakça

[1] Pullum, L.L., Software Fault Tolerance Techniques and Implementation, Artech House, MA, 2001

[2] Shooman, M.L., Reliability of Computer Systems and Networks, John Wiley & Sons, New York, 2002

[3] Anderson, T., Grabbie, T., Hammersley, J., Providing Open Architecture High Availability Solutions, High Availability Forum, 2001

[4] Acar, D., Erdoğdu, S., Dökmen, A., Şengül, M., Yaman, M., Sinyal İşleme Yazılımları için Mimari Tasarımı, UYMK, 2006