

Codezero Gömülü İşletim Sistemi Çekirdeği ve Sanallaştırma Platformu

Bahadır Bilgehan Balban¹

¹ B Labs Ltd. Firma Kurucu ve Teknik Sorumlusu, Bursa

¹e-posta: bbalban@b-labs.co.uk

Özetçe

Günümüz gömülü sistemleri özelleşmiş yapıları itibariyle özel ihtiyaçlara hitabeden işletim sistemi çekirdeklerine ihtiyaç duymaktadırlar. Bu sebeple sadece temel sistem işlevlerini içeren ve her gömülü sistemde ihtiyaç duyulmayacak dosya sistemi, hafıza yöneticisi, ağ iletişim katmanı gibi işlevlerden arındırılmış mikroçekirdek mimarileri önem kazanmıştır. Buna ek olarak gelişmiş gömülü sistemlerin işlemci ve hafıza kapasitelerinin artması sonucu, birden fazla yazılım bölümünün aynı gömülü sistem üzerinde bağımsız olarak çalışması mümkün hale gelmiştir. Bu iki olgu gözönüne alınarak, gelişmiş gömülü sistemlerin tamamına hitabedecek şekilde sadece temel sistem işlevlerini içeren Codezero isimli bir mikroçekirdek tasarlanmıştır. Bu mikroçekirdek aynı zamanda bir gömülü sistemi birbirinden izole edilmiş hafıza ve işlemci bölümlerine ayırarak hipervisör olarak adlandırılan bir sanallaştırma platformu olmayı da hedeflemektedir. Son olarak Codezero üzerinde çalışacak şekilde hafıza yönetimi ve dosya sistemi işlevlerini sağlayan POSIX arayüzüne sahip sistem servisleri geliştirilmiştir. Bu sistem servisleri aracılığıyla Codezero'nun tam teşekküllü bir işletim sistemi görevini de yerine getirmesi öngörülmektedir.

1. Giriş

Gömülü bilgisayar sistemlerinde kullanılan yonga ve veriyolu gibi temel donanım öğeleri her sistemde sisteme özgü olarak farklı bir mimari ile inşa edilmektedir. Buna ek olarak gömülü sistemlerin uygulama alanları da oldukça özelleşmiş olup, sistemden sisteme farklılık göstermektedir. Dolayısıyla bu sistemleri destekleyecek işletim sistemi çekirdeklerinin bu farklılıklara hitabedebilecek yapıya sahip olmaları gerekir. Bu açıdan bakıldığında işletim sisteminin birçok işlevini bir bütün olarak ortaya koyan tekil çekirdek tabanlı işletim sistemleri dezavantajlı konuma düşmektedirler. Örneğin bir dosya sistemine ihtiyaç duymayan, ya da mevcut bulunan hafıza yönetiminden daha farklı bir hafıza yönetim şekline ihtiyaç duyan gömülü sistemler için tüm bu işlevleri bütünlük olarak bulunduran tekil çekirdekler uygun düşmemektedirler. Bu sebeple işlevleri sadece hafıza alanları yaratılması, görev kontrolü ve görevler arası iletişim gibi temel ve soyut işlevlere indirgenmiş, Codezero isimli bir mikroçekirdek geliştirilmiştir. Yapısı itibariyle hem çekirdek küçülerek daha basit ve kararlı hale gelmiş, hem de bahsedilen temel işlevlere gereksinim duyan tüm gömülü sistemlerde kullanılmaya uygun düşecek evrensel bir mimari ortaya çıkmıştır. Codezero mikroçekirdeği, basitlik ve arılık olgularına sahip L4 mikroçekirdek arayüzünü[1,2] taban alarak silbaştan yazılmış,

esnek yapıya bir gömülü işletim sistemi çekirdeği olmayı hedeflemektedir.

Gömülü sistemler günümüz itibariyle işlemci gücü ve hafıza kapasitesi açısından gitgide gelişerek, eklenen donanım yongalarının da gelişmesiyle birlikte işlem kapasitesi yüksek, çok amaçlı ve oldukça karmaşık bir yapı kazanmaktadır.

Bu hususlar dikkate alındığında, bir gömülü donanım platformunda birden farklı yazılım ögesinin çalışması ve kontrollü bir şekilde yönetilmesi söz konusu olmuştur. Çoklu uygulamalara örnek olarak birden fazla işletim sisteminin aynı sistemde çalışması ya da bir özelleşmiş uygulamanın bir işletim sistemi yanında çalışması, ve işletim sistemi uygulamalarıyla iletişimde olması sayılabilir.

Öte yandan karmaşıklaşan gömülü sistemlerde bulunan birbirinden farklı donanım yongalarının ortak bazı işlevlerinin tek bir elden kontrollü olarak yönetilmesi de ortaya çıkan önemli gereksinimlerdenidir. Örneğin aynı sistemde bulunan birden fazla yonganın güç yönetimi, başlangıç yönetimi, erişilebilirlik ya da diğer bir deyişle güvenlik yönetimi gibi alanlarda tek bir elden yönetilmesi söz konusu olmuştur.

Codezero mikroçekirdeği yukarıda bahsedilen hususlar doğrultusunda, gömülü sistemin hafıza ve işlemci kaynaklarını birden fazla bölmeye bölerek her bir bölmede sanallaştırılmış bir işletim sistemi ya da bir gömülü uygulama yığınının bağımsız olarak koşmasını sağlayacak bir sanallaştırma platformu olarak tasarlanmıştır. Aynı zamanda donanım yongalarının bahsedilen güç, güvenlik ve başlangıç yönetimi alanlarında ortak yönetimi de gözetilmiş, bu temel işlevlerin yer almasına fırsat sağlayacak genel bir arayüz ile geliştirilmiştir.

Günümüzde, Codezero mikroçekirdeğine benzer işlevlerle tasarlanan sanallaştırma platformları bulunmaktadır. Codezero mikroçekirdeğinin bu platformlardan önemli bir farkı, kullanım alanının sadece işletim sistemi sanallaştırılmasından ibaret olmamasıdır. Örneğin, Linux ve benzeri bir işletim sistemi çekirdeğinin Codezero üzerinde sanallaştırılması mümkün olmakla beraber, silbaştan bir işletim sisteminin geliştirilmesi de mümkün kılınmıştır. Bu amaç doğrultusunda POSIX arayüzünün[3] önemli bir kısmını destekleyen hafıza yöneticisi ve dosya sistemi servisleri Codezero üzerinde çalışmak üzere geliştirilmiş ve kullanıma sunulmuştur.

2. Mikroçekirdek Yapısı ve Temel Özellikleri

Codezero mikroçekirdeği ve yeni kuşak mikroçekirdeklerin karakteristik özellikleri ortak olarak bu bölümde anlatılmıştır.

Mikroçekirdek yapısına sahip çekirdeklerde işletim sisteminin en temel işlevler dışında kalan tüm işlevleri mikroişlemci tarafından desteklenen yönetici modunun dışında kalan kullanıcı modunda yer alır. Örneğin hafıza yöneticisi, dosya sistemi yöneticisi, aygıt sürücüler bu kategoride bulunan işlevlerdir. Bu işlevler mikroçekirdek mimarisinde çekirdeğin dışında bulunduğundan, çekirdek tarafından kontrollü bir şekilde koşturulurlar ve sistemin işleyişini olumsuz yönde etkileyemezler. Sonuç olarak sistemin kararlılığı artmış olur.

Gömülü sistemler göz önüne alındığında, mikroçekirdek tabanlı işletim sistemlerinin bahsedilen karakteristik özellikleri daha çok önem kazanır. Çünkü gömülü sistemlerde hem sistem kararlılığı önem taşır, hem de kullanılacak işlevler arasında seçim yapabilmek özgürlüğü bu sistemlerin kendine has özelleşmiş mimarilerine uygun düşer.

Codezero mikroçekirdeğinde temel işlevler olarak adres alanlarının yönetimi, görev yönetimi ve görevler arası iletişim ele alınmıştır. Bu işlevler dışında örneğin hafızanın hangi algoritmalar vasıtasıyla uygulamalara dağıtılacağı kullanıcı alanında yer alacak bir veya birden fazla hafıza yöneticisinin özelleşmiş yönetim şekline bırakılmıştır.

Codezero mikroçekirdeğinin arı yapısı L4 mikroçekirdek mimarisi ele alınarak silbaştan tasarlandığından, yeni kuşak tasarım olarak adlandırılabilir. Örnek olarak geçmişte denenmiş olan düşük performanslı, kullanışsız ve karmaşık iletişim metotları içeren Mach[4] ve benzeri eski kuşak mikroçekirdek tasarımları ile benzerlik göstermez.

2.1. Teknik Özellikler

Codezero Mikroçekirdeği, sanal hafıza desteği bulunan gömülü sistemler için geliştirilmiştir. Sanal hafıza desteği ve korumalı hafıza sisteminin varlığı gömülü sistem üzerinde bulunan uygulamaların kontrollü olarak çalışması ve daha da önemlisi mikroçekirdeğin işleyişinin ve dolayısıyla sistemin bütününde kararlılığın korunması açısından önem taşır.

Codezero Mikroçekirdeği bellibaşlı birkaç sistem çağrısını destekler. Bu çağrılar sayesinde:

- Görevler yaratmak, değiştirmek, durdurmak ve başlatmak
- Adres alanları oluşturmak ve bu alanlara görevler eklemek
- Görevler arasında kontrollü olarak haberleşmek
- Sanal hafıza adresleri atamak
- Aynı adres alanında yeralan görevler arasında kilitli senkronizasyon sağlamak
- İşlemcinin önbellek (cache), adres çeviri önbelleği (TLB) gibi birimleri üzerinde kontrol sağlamak

- Gömülü platforma özgün olarak platformda bulunan donanım birimleri üzerinde erişim, güvenlik, güç sarfiyatı ve birim başlangıcı gibi genel konularda hakimiyet sağlamak

mümkündür.

Codezero mikroçekirdeği gerçek zamanlı uygulamalar dikkate alınarak geliştirilmiştir. Örneğin sistem çağrıları kesmeler ile kesilebilir olarak tasarlanmıştır. Bu sayede sistem çağrısında bekleme durumunda bulunan bir görevin kesme ile kesilerek çağrıyı terk etmesi mümkün olur. Mikroçekirdek aynı zamanda *preemptif koşma* özelliğine sahiptir. Bu sayede herhangi bir görev sistem çağrısında koşuyor olsa dahi zaman dilimi dolduğu takdirde görev zamanlayıcısı tarafından alıkoymulabilir. Mikroçekirdekte çok işlemcili sistemlerde koşma yetisi de bulunmaktadır. Buna ek olarak çekirdek minimum düzeyde işlev içerdiğinden yapılan işlemlerin çok azı kilitleme gerektirir. Bu sayede sistem işleyişinde kilit alınarak preemptif koşmanın engellendiği süreler çok kısa tutulmuştur.

Codezero mikroçekirdeğinde görevler arasında temel iletişim randevu sistemi ile senkron olarak yapılmaktadır. Dolayısıyla görevlerin göndermek istedikleri mesajlar çekirdek içinde yer tutmaz, ayrıca senkron görev işleyişi korunarak sistemin kararlı olması sağlanır.

Görevler arası iletişimi sağlamak üzere *UTCB* olarak adlandırılan bir görev mesaj alanı bulunmaktadır. Bu alan her görev için belli bir adres aralığında atanmıştır ve kullanıcı alanından erişilebilir konumdadır. Görevler göndermek istedikleri mesajları tanımlayarak bu alanları doldururlar ve mikroçekirdek sistem çağrıları aracılığıyla iletişim kurmak istedikleri görevlerle haberleşirler.

Mesajların küçük olması durumunda hafızaya dokunulmadan sadece saklayıcılar üzerinden aktarımı sayesinde mikroçekirdeklerde performans düşüklüğü yarattığı düşünülen görev iletişim mekanizması da iyileştirilmiştir.

Codezero Mikroçekirdeği ARM gömülü mikroişlemcilerini desteklemektedir. *K&R C¹* programlama dili kullanılarak oldukça uyarlanabilir bir biçimde tasarlandığından sanal hafıza desteği bulunan diğer işlemci mimarilerine de taşınabilir olma özelliğine sahiptir.

Yazılım lisansı açısından bakıldığında, Codezero projesi *GPLv3*[5] lisansı ile dağıtım opsiyonu sayesinde geliştiriciler açısından uygun bir gömülü yazılım geliştirme aracı olmayı amaçlamaktadır.

3. Sanallaştırma Platformu

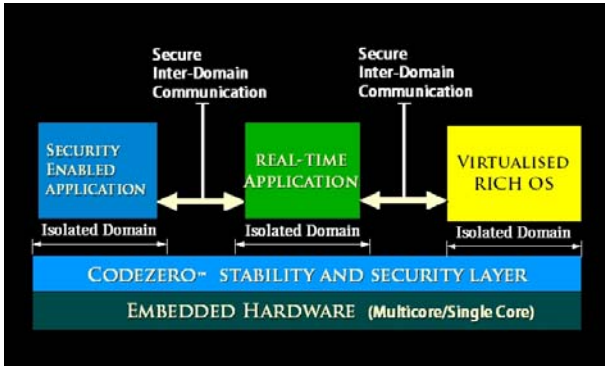
Günümüz gömülü sistemlerinde yaygınlaşan bir uygulama, Hipervisör adlı bir mini-çekirdek aracılığıyla sistemin hafıza ve işlemci kaynaklarının *bölme* olarak adlandırılan birden fazla bölüme ayrılması ile, bu bölmelerde farklı uygulamaların çalıştırılması şeklindedir. Bu yöntemin gömülü sistemler açısından birçok faydası bulunmaktadır. İlk olarak gömülü sistem tekil ve karmaşık bir işletim sistemi çekirdeği yerine hipervisörün yönetiminde çalıştığından çökmelerden korunur. Hipervisör yazılımı bu işlevine ilişkin olarak *güvenilir yazılım*

¹ Kernighan and Ritchie, *The C Programming Language*, 3rd Edition

tabanı olarak adlandırılır. Hipervisör oldukça basit ve küçük olarak tasarlandığından, hatalı işlem yapma olasılığı oldukça düşüktür. Sonuç olarak bir güvenilir yazılım tabanı olarak Hipervisör sayesinde gömülü sistemin kararlılığı korunur.

Hipervisörlerin diğer bir işlevi, gömülü sistemde yaratılan bölmelerin güvenli ve kontrollü bir şekilde koşturulması ve bölmeler arası kontrollü haberleşmenin sağlanmasıdır. Bu bir bakıma bölmeler arasında mantıksal birer bariyer yaratılması anlamına gelir. Sonuç olarak hipervisörün doğru ve kararlı çalışması esastır.

Hipervisör yazılımı sayesinde bağımsız ve kendi bölmesinde izole edilmiş olarak çok çeşitli yazılım servisleri çalıştırılabilir. Şekil 1'de de örneklendirildiği gibi, Linux ve benzeri zengin özelliklere sahip ancak güven seviyesi düşük bir işletim sisteminin yanında, yüksek güvenli bir uygulamanın çalıştırılması mümkündür. Uygulama örneği olarak telif hakları yönetim yazılımları, ödeme yazılımları ve gerçek zamanlı ağ iletişim yazılımları verilebilir.



Şekil 1: Codezero üzerinde sanallaştırılmış zengin işletim sistemi ile yüksek güvenli ve gerçek-zamanlı uygulamalar

Hipervisör yazılımları gömülü sistemlerdeki yönetici işlemlerini yapmaya yetkin tek yazılım olduklarından, aynı işlemleri yapmak ve sistemde bütünsel hakimiyet sağlamak üzere yazılmış bulunan işletim sistemi çekirdeklerinin yönetici işlemlerini hipervisör yazılımı üzerinden gerçekleştirmesi beklenir. Örneğin sayfa tablolarının değiştirilmesi, yeni adres alanları yaratılması, işlemci önbelleği üzerinde değişiklikler yapılması bu kategorideki işlemlerdir. Sonuç olarak Hipervisör yazılımı üzerinde müşterek olarak çalışacak olan çekirdeklerin bu olanağa fırsat verecek şekilde değiştirilmesi ve taşınması gerekmektedir. Bu işlem *Sanallaştırma* olarak adlandırılır.

Codezero mikroçekirdeği, işletim sistemi servislerinin geliştirilmesi için taban bir sistem olmakla beraber, yukarıda bahsedilen şekilde başka işletim sistemlerinin donanımdan soyutlanarak sanallaştırılması için de uygun bir yazılım tabanı oluşturmaktadır. Örneğin sanallaştırma için gerekli olan yönetici işlemlerini yerine getirmek üzere 2.1 bölümünde bahsedilen sistem çağrılarını kullanmak yeterlidir. Bu amaçla Linux işletim sistemi çekirdeğini Codezero mikroçekirdeği

üzerinde çalıştırmak üzere değiştirme ve taşıma çalışmaları başlamıştır.

Hipervisörlerde bulunan güvenilir yazılım tabanı yetilerini sağlamak açısından Codezero mikroçekirdeği arı tasarımı itibarıyla uygun mimari yapıya sahiptir. Ancak kararlılık açısından iyileştirme çalışmasına gidilerek bu özelliğin sağlanması planlanmaktadır.

Sonuç olarak Codezero mikroçekirdeği, hem hafıza bölüşüm ve birden farklı hafıza yöneticisine fırsat veren bir sistem olarak, hem işletim sistemi sanallaştırması için gerekli sistem çağrılarını sahip olarak, hem de basit ve arı yapıdan kaynaklanan kararlılık yetileri sayesinde gömülü sanallaştırma platformları arasında sağlam bir yer edinmeye aday bir sistemdir.

4. Gömülü Sistem Yöneticisi

Günümüzde oldukça karmaşılaşan ve heterojen mimarilere doğru yol alan gömülü sistemlerde, sistemde bulunan çok amaçlı ve farklı özelliklere sahip donanım birimlerinin tek elden kontrollü bir biçimde yönetilmesi önemli bir gereksinim haline gelmiştir. Codezero Mikroçekirdeğinin amaçladığı bir diğer rol ise gömülü sistemlerde bulunan bu donanım birimlerinin temel ve ortak işlevler açısından yönetimini üstlenmektir.

Mikroçekirdeğin gömülü uygulama ve işletim sistemlerini güvenli ve kontrollü bir şekilde sistemde koşturmasının yanında, donanım birimleri üzerinde de temel ve ortak işlevler açısından kontrolünün olması doğal bir yönelimdir. Gömülü sistemlerde yönetilecek temel işlevler 4 ana kategoride sıralanabilir:

- Güç sarfiyat yönetimi
- Donanım erişimi ve güvenlik yönetimi
- Donanım hata bildirim yönetimi
- Donanım başlatma ve senkronizasyon yönetimi

Bu bahsedilen işlevler günümüz işletim sistemlerinde aygıt sürücüler tarafından bağımsız olarak yürütülmektedirler. Örneğin her bir donanımın güç yönetimi, o donanıma has bir yöntem ile sağlandığından, birbirinden farklı birçok yönetim mekanizması ortaya çıkmıştır. Buna ek olarak bu mekanizmaların uygulamaların en üst seviye katmanlarından kontrol edilmesi karmaşıklığa sebep olmaktadır.

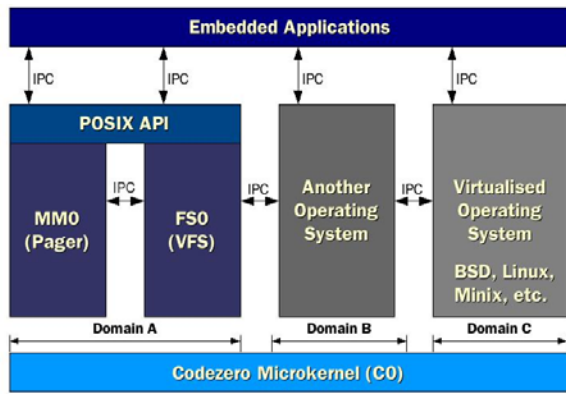
Buna karşılık bu özelliklerin tek bir elden yönetilmesi ve uygulamaların doğrudan donanım yöneticisi ile iletişim kurulabilmesi daha arı ve kolay kullanım sağlayacak bir yaklaşımdır. Örneğin Linux işletim sisteminin Codezero üzerinde sanallaştırılması durumunda, bir Linux uygulamasının güç sarfiyatını azaltmak üzere donanım yöneticisi olarak doğrudan Codezero mikroçekirdeği ile iletişim kurmasını sağlamak mümkündür.

Bahsedilen temel donanım yönetim işlevlerini yerine getirmek üzere Codezero mikroçekirdeğinde bir platform sistem çağrısı bulunmaktadır. Bu çağrının, gömülü platformda bulunacak ve yukarıda sayılan işlevleri yerine getirecek bir donanım yönetim yongasına hükmederek, ilgili işlevleri yerine

getirmesi planlanmıştır. Bu amaçla, yongayı sağlayacak donanım firması ile ortak olarak somut tasarım çalışmaları yürütülmektedir.

5. Gömülü İşletim Sistemi

Codezero mikroçekirdeği önceki bölümlerde bahsedilen hafıza bölüşüm ve sanallaştırma platformu olma özellikleri yanında, bulundurduğu temel sistem işlevleri sayesinde bir işletim sisteminin silbaştan geliştirilmesine de fırsat verir. Şekil 2'de de örneklendirildiği gibi, Codezero aracılığıyla gömülü sistemde yaratılmış bölmelerden birinde, işletim sistemi görevi görecek bir hafıza yöneticisi ve bununla birlikte çalışan dosya sistemi yöneticisi, aygıt sürücülerini gibi görevler de geliştirmek mümkündür.



Şekil 2: Codezero üzerinde A bölgesinde yer alan POSIX servisleri ve sistemdeki diğer bölmeler

Codezero'nun bu alandaki yetilerini göstermek amacıyla, POSIX arayüzünün küçük fakat önemli bir kısmını destekleyen MM0 isimli bir hafıza yöneticisi ve FS0 isimli bir sanal dosya sistemi görevi geliştirilmiştir.

5.1. MM0 Hafıza Yöneticisi

MM0 hafıza yöneticisi modern bir POSIX işletim sisteminden beklenenecek hafıza yönetim yetilerine sahip bir hafıza yöneticisidir. Bu görev, sayfa önbelleği ve uygulamalara sayfa atamalarından sorumludur. Örneğin sayfa aksamalarını ele alır, fiziksel sayfaları ve onların dosyalar ve görevler ile olan bağıntılarını elde tutar. *FreeBSD* işletim sisteminde yer alan hafıza objelerine benzer bir yapı örnek alınarak silbaştan geliştirilen bu görev sayesinde uygulamaların koşturulması esnasında talebe bağlı sayfalama (demand paging) tekniğini kullanmak da mümkün olmuştur.

5.2. FS0 Sanal Dosya Sistemi

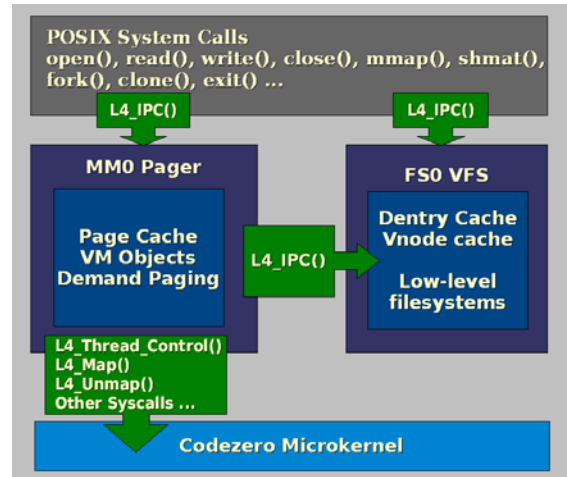
FS0 isimli sanal dosya sistemi görevi, MM0 ile müşterek olarak aynı bölme içinde çalışmak üzere tasarlanmıştır. Ancak bu görev kendi adres alanında çalışır ve MM0 ile görevler arası iletişim mekanizması sayesinde iletişim kurar.

FS0 modern POSIX tabanlı işletim sistemlerinde bulunan dosya yönetim sistemleri ile benzer yetilere sahip bir dosya yöneticisidir. Dizin önbelleği ve dosya önbelleğinden sorumlu bu görev, farklı dosya sistemi yapılarının hemen hemen hepsinde bulunan ortak özelliklerini üst bir katmanda genelleştirerek, farklı dosya sistemlerine ortak bir sanal arayüz ile erişimi sağlar. Örneğin MM0 hafıza yöneticisi, sayfa önbelleğinde bulunmayan dosya sayfalarını bu görevden edinerek işlevini yerine getirir.

5.3. Çekirdek ve POSIX Servisleri Etkileşimi

Önceki bölümlerde belirtildiği üzere, Codezero Mikroçekirdeği aracılığıyla gömülü sistemde bölme olarak adlandırılan bölümler oluşturulur ve her bir bölmede farklı bir uygulama ya da uygulama grubunu çalıştırmak mümkündür. Yukarıda bahsedilen işletim sistemi servisleri Codezero üzerinde bulunacak tek bir bölmede çalışmak üzere tasarlanmıştır. Örneğin her bir bölmede bir tane bulunması beklenen hafıza yöneticisi görevini bu bölmede MM0 görevi üstlenmiştir.

Codezero mikroçekirdeği sadece temel işlevleri sağladığından, bölmelerdeki yönetim şekilleri hakkında hiçbir varsayım ya da bilgi içermez. Örneğin önceden belirlenmiş bir sanal hafıza ve fiziksel hafıza alanı MM0 görevine atanmıştır ve bu alanların nasıl yönetileceği MM0 görevine bırakılmıştır. MM0 görevi Şekil 3'te gösterildiği gibi, gelen POSIX sistem çağrı talepleri doğrultusunda hafıza atamalarını yapmak, adres alanları ve görevler yaratmak amacıyla mikroçekirdek sistem çağrılarını başvurur. Bahsedilen sistem çağrıları yönetim amaçlı olduğundan güvenliği sağlamak adına mikroçekirdek tarafından sadece bölmede hafıza yöneticisi olarak atanmış görevin bu çağrıları yapması kabul edilir.



Şekil 3: Çekirdek ve POSIX servisleri etkileşimine yakından bakış

Sistemde bulunan kullanıcı uygulamaları, MM0 ve FS0 görevleri ile mikroçekirdek tarafından sağlanan görevler arası iletişim mekanizmasını kullanarak haberleşirler.

Bir çalışma örneğini ele almak gerekirse, herhangi bir uygulama tarafından *execve()* POSIX çağrısı yapıldığında, MM0 görevi bir iletişim çağrısı ile haberdar edilir. MM0 görevi, mikroçekirdeğin sunduğu yönetici sistem çağrıları sayesinde çağrıyı yapan görevi sonlandırarak FS0 görevinden koşturulacak yeni programın dosyasını, yine iletişim çağrısı sayesinde ister. Sonraki aşamada MM0 tekrar mikroçekirdek yönetici çağrılarını kullanarak yeni programı başlatır.

6. Sonuç

Gömülü sistemlerin işlem gücü ve kapasitesinin artması sonucu, bu sınıfta bulunan bilgisayar sistemlerinin hayatın her alanına yayılması sözkonusu olmuştur. Bu sistemlerin güçlenmesi ve karmaşık hale gelmesi sonucu son zamanlarda ortaya çıkan gereksinimleri karşılayacak çok yönlü işletim sistemleri ve yazılımlara ihtiyaç duyulmaya başlanmıştır. Bu ihtiyaçları karşılamak üzere özgün bir mimari ele alınarak Codezero mikroçekirdeği projesi ortaya çıkmıştır.

Codezero projesi sanallaştırma özellikleri, gömülü platform yönetim özelliği ve işletim sistemi özelliği ile çok amaçlı bir sistem yazılımı olma yolunda ilerleme kaydetmektedir. Özellikle gömülü sistemlerin kararlılığı, kolay kullanımı ve esnekliğine yönelik işlevleri olmasıyla projenin dünya çapında yer edinecek değerli bir proje haline gelmesi beklenmektedir.

Codezero projesinin el bilgisayarları, cep telefonları, savunma sanayii, sağlık sektörü, güvenlik uygulamaları gibi birçok alanda kullanılabilmesi mümkündür. Açık kaynak lisans opsiyonu sayesinde elde edilen bilgi birikiminin geliştiriciler ile paylaşılması da projenin erişilebilirliği açısından önemli bir husustur.

Sonuç olarak dünya ile beraber ülkemizde de stratejik önemi bulunan sistem yazılımları alanında Codezero projesinin önemli bir boşluğu doldurması beklenmektedir.

7. Kaynakça

- [1] System Architecture Group, University of Karlsruhe, *"The L4Ka::Pistachio Microkernel, White Paper, 2003"*
- [2] National ICT Australia, *"NICTA L4-embedded kernel reference manual version N1, October 2005"*
<http://ertos.nicta.com.au/software/kenge/pistachio/latest/refman.pdf>
- [3] IEEE Standards Association, *"Portable Operating Systems Interface"*
<http://www.standards.ieee.org/regauth/posix>
- [4] Black, D.L., Golub, D.B., Julin, D.P., Rashid, R.F., Draves, R.P., Dean, R.W., Forin, A., Barrera, J., Tokuda, H., Malan, G., and Bohman, D. *"Microkernel Operating System Architecture and Mach"* Proceedings of the USENIX Workshop on Micro-Kernels and Other Kernel Architectures, April 1992.
- [5] GNU Foundation *"GNU General Public License v3, 2007"*. <http://www.gnu.org/gpl-3.0.html>