

# WebAssembly

Harun OKTAY  
Alper ONRAT

## Özet

WebAssembly, tarayıcılarda yüksek performanslı kod çalıştırmak için kullanılan alt seviye bir programlama dilidir. Bu teknoloji sayesinde hızlı web uygulamaları geliştirilebilmektedir. Ancak, WebAssembly yalnızca web amaçlı kullanılmak zorunda değil; güvenlik, performans ve taşınabilirlik gibi birçok farklı amaç için de kullanılabilir. Stack tabanlı bir yapıyı kullanarak işlemleri stack üzerindeki değerlerle gerçekleştirir. C++ ve Rust gibi dillerle entegre edildiğinde, bu dillerin kodlarını WebAssembly'nin ikili formatına dönüştürüp çalıştırabilir, böylece C++ ile yazılmış araçların web üzerinden kullanılmasını sağlar. Ayrıca, WebAssembly'nin web dışında kullanılacak birçok özelliği mevcuttur. WebAssembly Text Formatı (WAT) ile doğrudan alt seviyede yazılabilir ve doğrudan assembly formatında yazılan fonksiyonlar genellikle daha verimli çalışır. Bu özellikleri sayesinde WebAssembly'nin aviyonik sistemler ve araçlar için de uygulanabilir olduğunu düşünüyorum.

## WebAssembly Nedir?

WebAssembly (Wasm), web uygulamalarında yüksek performanslı kod çalıştırmak için kullanılan düşük seviyeli bir ikili formattır. WebAssembly, web tarayıcıları ve diğer ortamlarda hızlı ve verimli bir şekilde çalışacak şekilde tasarlanmıştır. Günümüzde kullanılan ana tarayıcılar (Chrome, Firefox, Edge) WebAssembly kodunu destekler ve doğrudan çalıştırabilir; bu, JavaScript'e göre daha hızlı performans ve daha düşük başlatma süreleri sağlar. WebAssembly, genellikle C, C++, Rust ve diğer dillerden derlenen kodları destekler; böylece bu dillerle yazılmış uygulamalar tarayıcı ortamında sorunsuz bir şekilde çalıştırılabilir.

WebAssembly, taşıma ve güvenlik açısından önemli avantajlar sunar. WebAssembly modülleri, platformdan bağımsız bir biçimde çalışabilir; yani aynı modül farklı işletim sistemlerinde ve donanımlarda aynı şekilde çalışır. Ayrıca, WebAssembly kodu sandbox ortamında çalıştığı için güvenlik risklerini en aza indirir. Bu teknoloji, aynı zamanda büyük ve karmaşık uygulamaların daha küçük ve yönetilebilir parçalara ayrılmasını sağlayarak performansı artırır. WebAssembly'nin bu özellikleri, onu sadece web uygulamaları için değil, oyunlar, veri işleme, görüntü işleme ve diğer performans-kritik uygulamalar için de uygun bir seçim haline getirir.

Benim en mantıklı bulduğum yöntem ise, en çok kullanılan döngülerde JavaScript kodunu döndürmek yerine WebAssembly kullanmaktır. Bu, kodun hızını ve verimliliğini önemli ölçüde artırabilir. Optimize ettiğimiz bir döngünün her adımı milisaniye seviyesinde zaman kazandırsa bile, bu döngü bir milyon kez dönse, bu durum bize dakikalar kazandırabilir. Bu kayıp günlük kullanımlar için normal karşılanabilse bile, savunma, sağlık ve aviyonik gibi alanlarda milisaniyeler bile çok değerlidir.

## WebAssembly'nin Motor Kontrol Araçlarına Uygulanması

WebAssembly, aviyonik sistemlerde kullanılan motor kontrol araçlarının performansını artırmak ve fonksiyonelliğini genişletmek için etkili bir teknoloji olabilir. Burada kullanılan; motor parametrelerini değiştiren, güncelleyen ve gözleyen, Qt ve C++ kullanılarak geliştirilen bu program Webassembly geliştirmeye oldukça uygun olabilir. Bu programı WebAssembly ortamına taşımanın, çeşitli avantajlar sağlayabileceğini düşünüyorum. Bu avantajları şu şekilde sıralayabilirim:

**Performans:** Aviyonik sistemlerin, milisaniyeler düzeyinde hassasiyet ve hızlı tepki süreleri gerektirdiğini gözlemledim. WebAssembly, alt seviye bir dil olarak yüksek performans gösterebildiğini söylemişim. C++ oldukça hızlı çalışsa da bu programı WebAssembly'e geçirdikten sonra WAT formatında fonksiyonlar ekleyerek daha da hızlı hale getirebileceğini düşünüyorum. Bu da motor parametrelerinin daha hassas ve hızlı bir şekilde güncellenmesini sağlayabilir.

**Alt Seviye Veri İşleme:** WebAssembly, düşük seviyeli veri işleme yeteneklerine sahiptir ve bu, motor parametrelerinin ayrıntılı ve doğrudan bir şekilde işlenmesine olanak tanır. WebAssembly'nin CAN protokolü ile doğrudan entegrasyonu üzerine çalışmalar günümüzde mevcut olmasa da, WebAssembly'nin alt seviyede veri işleme kapasitesinin, motor verilerinin etkili bir şekilde yönetilmesine katkı sağlayabileceğini düşünüyorum. Arama, sıralama, Checksum gibi fonksiyonların hesaplaması WAT formatında yazılabilir diye düşünüyorum.

**Web Ortamı:** WebAssembly, Qt arayüzlerinin web tarayıcılarında güvenli bir ortamda çalışmasını sağlar. Bu özellik, motor parametrelerini uzaktan izlemek ve yönetmek için erişilebilir ve güvenli bir ortam sunar. Bu sayede, motor kontrol arayüzüne yerel bir tarayıcı üzerinden erişim sağlanabilir. Eğer kablosuz sinyal bağlantısı kurulabilirse uçak havadayken de iletişim kurulabilmesi mümkün olabilir.

**Taşınabilirlik:** Qt, kod düzeyinde platformlar arası taşınabilirlik sağlar, ancak WebAssembly bu taşınabilirliği daha da artırabilir. WebAssembly modülleri, farklı işletim sistemlerinde ve tarayıcılarda tutarlı bir şekilde çalışabilir, bu da uygulamanın çeşitli ortamlarda daha kolay dağıtılmasını ve kullanılmasını sağlar.

**Güvenlik:** WebAssembly, sandbox ortamında çalışarak güvenlik risklerini minimize eder. Bu, hassas motor verilerinin işlendiği aviyonik sistemlerde güvenlik standartlarını karşılamak için önemlidir. Kodun izole bir ortamda çalışması, güvenlik açıklarını azaltır ve uygulamanın genel güvenliğini artırır.

**Gömülü yazılım İçin Faydalar:** WebAssembly, gömülü cihaz üreticileri için çeşitli avantajlar sunar. Qt uygulamalarını WebAssembly hedef platformu için yeniden derleyerek web tarayıcısında çalıştırmak mümkündür. Ayrıca, JavaScript ve WebAssembly bileşenlerini birleştirerek mevcut Qt uygulamasının bazı parçalarını yeniden kullanmak, geliştirme sürecini önemli ölçüde azaltabilir ve UI öğelerinin farklı platformlarda kullanılabilirliğini artırabilir. WebAssembly kullanan uygulamalar, tarayıcıda sıfır kurulum ile çalışarak mobil cihazlarda uygulama mağazalarını atlatma imkanı sağlar.

**Uzaktan Kullanıcı Arayüzleri ve Merkezi İzleme:** WebAssembly, uzaktan kullanıcı arayüzleri ve merkezi izleme çözümleri için mükemmel bir seçenektir. Mevcut Qt uygulama kodunu yeniden kullanarak, gömülü sistem üreticileri uzaktan ekranlar veya mevcut ekranların yerini alacak çözümler geliştirebilir. Ayrıca, WebAssembly, merkezi izleme ve yüksek hesaplama kaynakları gerektiren yapay zeka algoritmalarını desteklemek için de uygundur.

**Kullanıcı Arayüzü Prototip İşbirliği:** WebAssembly, geliştirme sürecinde uygulamaları paydaşlarla kolayca paylaşmayı sağlar. Bir Qt uygulamasını yeniden derleyip web sunucusuna yükleyerek URL'yi paylaşmak, geri bildirim toplamayı kolaylaştırır. Bu, tamamlanmamış donanımları paylaşma veya uygulama mağazası süreçlerine girme ihtiyacını ortadan kaldırır.

WebAssembly'nin, mevcut kodu yeniden kullanma ve yüksek performanslı işlemler için avantajlar sağlaması mevcut yazılımlarda avantaj sağlayabilir. Bu teknoloji, yeni yenilikler için kapıları açarak, motor kontrol sistemlerinin daha verimli ve erişilebilir hale gelmesini sağlayabilir.

## Qt Basit bir arayüz Uygulamasının WebAssembly'e Geçirilmesi

Şu an üzerinde çalışılan araç çok karmaşık olduğu için ekstra uğraş gerektirecektir ama basit bir Qt arayüz uygulamasını WebAssembly ortamına taşımak için aşağıdaki adımlar izlenebilir:

**1. Qt ve Emscripten Kurulumu:** İlk olarak, Qt'nin WebAssembly destekleyen sürümünü ve Emscripten'i kurmanız gerekir. Emscripten, C++ kodlarını WebAssembly modüllerine derlemeye olanak tanıyan bir araçtır.

**2. Proje Ayarları:** Qt projenizi, WebAssembly hedef platformu için yapılandırmanız gerekir. Bu, 'CMakeLists.txt' veya 'qmake' dosyalarındaki derleme ayarlarının WebAssembly'e uygun şekilde düzenlenmesini içerir.

**3. Kodun Derlenmesi:** Qt uygulamanızı Emscripten kullanarak derleyin. Emscripten, C++ kodunuzu WebAssembly modülüne dönüştürecek ve gerekli JavaScript bağlayıcı kodunu oluşturacaktır. Bu işlem genellikle komut satırı aracılığıyla gerçekleştirilir.

**4. Web Ortamına Yerleştirme:** Derlenen WebAssembly modülünü ve JavaScript bağlayıcı kodunu bir web sunucusuna yerleştirin. Bu modüller, bir web tarayıcısında çalışacak şekilde yüklenebilir.

**5. Test ve Optimize Etme:** Uygulamanızı çeşitli web tarayıcılarında test edin ve performansını optimize edin. WebAssembly kodu tarayıcıda çalışırken, bazı optimizasyonlar gerekebilir.

Bu adımlar, Qt arayüz uygulamasının WebAssembly ortamına taşınmasını ve web tabanlı bir uygulama olarak çalıştırılmasını sağlar.

## WebAssembly'nin Stack Mantığı ve Temel Sözdizimi

WebAssembly, stack tabanlı bir hesaplama modeline sahiptir. Bu modelde, tüm işlemler bir yığın (stack) üzerinde gerçekleştirilir. Veriler stack'e push edilir ve işlemler (örneğin toplama veya çarpma) stack'teki değerlerle gerçekleştirilir.

Aşağıda, WebAssembly'nin stack mantığını ve temel sözdizimini göstermek için 'multiply' (çarpma) fonksiyonunun bir örneği verilmiştir. Bu örnekte iki sayıyı çarpan bir fonksiyon tanımlanmıştır.

### WebAssembly Text Formatı (WAT)

WebAssembly'nin Text Formatı (WAT), WebAssembly kodunun insan tarafından okunabilir bir biçimidir. Aşağıda 'multiply' fonksiyonunun WAT formatında tanımı verilmiştir:

```
(module
  (func $multiply (param $a i32) (param $b i32) (result i32)
    ;; Her iki parametreyi stack'e push eder
    local.get $a
    local.get $b
    ;; Çarpma işlemini yapar
    i32.mul
  )
  ;; Fonksiyonu dışa aktarır
  (export "multiply" (func $multiply))
)
```

### Açıklama

1. **Fonksiyon Tanımı:** - (func \$multiply (param \$a i32) (param \$b i32) (result i32)): multiply adında bir fonksiyon tanımlar. Bu fonksiyon iki i32 (32-bit tamsayı) parametre alır ve bir i32 sonucu döndürür.

2. **Parametreleri Stack'e Push Etme:** - local.get \$a ve local.get \$b: Bu komutlar, fonksiyona geçirilen a ve b parametrelerini stack'e getirir. WebAssembly'de, local.get komutu belirli bir yerel değişkeni stack'e alır.

3. **İşlem:** - i32.mul: Stack'teki iki üstteki değeri (yani a ve b) alır ve çarpar. Sonuç stack'e push edilir.

4. **Fonksiyonu Dışa Aktarma:** (export "multiply" (func \$multiply)): Bu komut, multiply fonksiyonunu dışa aktarır ve WebAssembly modülünde erişilebilir hale getirir.

## Sonuç

WebAssembly, gömülü sistemlerde ve aviyonik motor kontrol araçlarında önemli bir teknoloji olabilir. Performans, taşınabilirlik ve güvenlik avantajları ile WebAssembly, mevcut Qt uygulamalarını ve diğer gömülü sistemleri optimize ederek performansı artırabilir. Gömülü cihaz endüstrisinde WebAssembly'nin benimsenmesi, kod yeniden kullanımını ve yüksek performanslı uygulamaları destekleyerek önemli faydalar sağlayacaktır.

## References

- [1] WebAssembly. (2023). WebAssembly Documentation. Retrieved from <https://webassembly.org/docs/>
- [2] WebAssembly. (2023). WebAssembly Specification. Retrieved from <https://webassembly.org/spec/>
- [3] Schneider, P. (2022). Does WebAssembly Matter for Embedded System Makers? Retrieved from <https://www.qt.io/blog/does-webassembly-matter-for-embedded-system-makers>
- [4] Qt Company. (2021). Qt for WebAssembly. Retrieved from <https://doc.qt.io/qt-5/wasm.html>
- [5] Qt Company. (2021). Qt and WebAssembly: Opportunities for Embedded Development. Retrieved from <https://www.qt.io/blog/qt-and-webassembly-opportunities-for-embedded-development>
- [6] Haas, A. (2020). WebAssembly Performance: An Overview. Retrieved from <https://arxiv.org/abs/2003.07918>
- [7] Hutter, H. (2021). Understanding WebAssembly: A Tutorial with Examples. Retrieved from <https://hutter.dev/understanding-webassembly/>
- [8] Kilian, C., Braun, M. (2023). WebAssembly in Avionics: Decoupling Software from Hardware. Retrieved from [https://www.researchgate.net/publication/375485920\\_WebAssembly\\_in\\_Avionics\\_Decoupling\\_Software\\_from\\_Hardware](https://www.researchgate.net/publication/375485920_WebAssembly_in_Avionics_Decoupling_Software_from_Hardware)