Mobil Programlama Projesi Rapor #1

İçindekiler

1 SUNUCU ÜZERİNE VE DEĞİŞİKLİKLER

1.1 TEKNİK GELİŞMELER

- DOM ağacı tasarlandı. UTF-8 dosya okuma, yazma, girdi ve çıktı işlemleri locale değişkeni ".UTF8" e ayarlanarak sağlanır. Standard kütüphanenin dar operatörleri, tek seferde bir bayt okuyup yazabilir ve CP1254 Türkçe veya varyasyonlarını kullanabilir. Herhangi bir ayar yapılmadığı takdirde, CP437 kullanılır ki bu modern sistemler için uyumsuz bir CP'dir. STD geniş operatörler, iki byte tutabildiği için UTF8 kodlama ile görev yapabilir. Geniş operatörler yalın sayı değerleri ile iyi çalışmaz. UTF16'ya denktir ama aynı zamanda UTF8 dosyalarla uyumludur. 3 ve 4 kod noktalı UTF8 karakterlerle nasıl çalıştığı test edilmedi ve bilinmiyor.
- Ayıklanan HTML dosyasının saklanması için DOM ağacı tasarlandı ve nullsafe özellik kazandırıldı.
- HTML dosyalarını okumak ve ayıklamak için tokenleştirme ve yazım kurallarına göre ayrıştırma algoritmaları geliştirildi. Tokneleştirme algoritması sadece elementlerin kendisini tek olarak tokenleştiriyor ancak html dosyasının tamamını da tokenleştirebilecek şekilde tasarlandı. Yazım kurallarına göre ayrıştırma algoritması, ayrıştırma yaparken token algoritmasını kendi içinde kullanıyor ve yardım alıyor. Ayrıştırma algoritmasında direkt olarak DOM ağacı oluşturuluyor.
- HTML dosyası okunurken raw byteları okuyor. Yani UTF8 farkındalığı yok. UTF8, CP1252 ve ASCII'nin ilk blokları için geriye dönük uyumlu olduğu için çiğ byte okumakta şimdilik sakınca görülmemektedir. Gerek duyulduğunda geniş operatörler ile okunabilir.
- Hem boşlukları hem kesme işaretlerini HTML'e kodlamadan dümdüz yazılmış. Bu nedenden dolayı HTML ayıklayıcıda çalışan tokenleştirici algoritması düzgün çalışamıyor. Birkaç değişiklik yapılacaktır.
    - <a href="https://www.geeksforgeeks.org/explore?page=1&sprint=ca8ae412173dbd8346c26a0295d098fd&sortBy=submissions&sprint_name=Beginner's DSA Sheet&itm_source=geeksforgeeks&itm_medium=main_header_outIndia&itm_campaign=DSA_Header" target="_self">
- HTML ayıklayıcı tamamlandı. 280 KB boyutundaki rastgele bir web sayfasını başarıyla ayıkladı. Hafızada yaklaşık 3-4 MB yer kaplayan bir DOM ağacı oluşturdu. DOM element sınıfının barındırdığı değişkenler dinamik hafıza tahsis eden konteyner ve string sınıflarından oluştuğu için hafızada daha fazla yer kaplamasına neden oluyor. Verilerin standart kütüphaneye ait sınıfların içinde saklanması daha güvenlidir ve geliştirme süresini azaltmaktadır. Bunların yanı sıra, optimizasyonlar ile hafıza ve işlem yükü azaltılabilir.
- HTML ayıklayıcı ve DOM ağacı için yorum satırları eklendi. DOM'da element silme fonksiyonu düzeltilecek ve HTML ayıklayıcının hataları giderilecek. Tokenleştiricinin hataları giderildi.
- Uzun ve kapsamlı gözlemler sonucunda HTML ayıklayıcının hatası olmadığı saptandı. Test aşamasında kullanılan HTML dosyasının yanlış yazıldığı ortaya çıktı. Düzgün HTML dosyaları sorunsuz bir şekilde ayıklanabilmektedir.
- DOM'da element silme fonksiyonları geliştirildi ve başarılı bir şekilde çalıştığı testler sonucunda anlaşıldı.
- POCO ile basit bir sunucu demosu oluşturuldu.

1.2 Ağ Protokolleri Hakkında ve Kütüphane Seçimi

1.2.1 Genel Bilgilendirme

- POCO kütüphanesi, kullanımı kolay olacak şekilde tasarlanmıştır. Bunun yanında birçok haberleşme protokolünü desteklese de, belgelenmesi zayıf olduğundan geliştirme sürecini daha da karmaşıklaştırmaktadır. Negatif yanları, pozitif özelliklerine ağır basmaktadır. Başka bir alternatif kütüphane bulunması projenin yararınadır.
- TCP, UDP, QUIC ve Socket gibi düşük katman protokoller; mevcut yüksek seviye protokoller ile çözülemeyecek ekstrem bir durum söz konusu değil ise kullanılmamalıdır!

1.2.2 Protokoller

1.2.2.1 WebSocket

- Sunucu-istemci arası çift yönlü asenkron (Full-Duplex), düşük gecikmeli ve gerçek zamanlı iletişim sağlar. Bağlantıları canlı tutmak için Ping-Pong yöntemi kullanır. İstemciden cevap gelmezse, belirli bir süre sonunda sunucu bağlantıyı kapatır. Bağlantı yalnızca istemci tarafından açılabilir. PNS geliştirmek için gerekli olabilir.

1.2.2.2 HTTP ve Varyasyonları

- HTTP bağlantısı (istek->cevap->bağlantı sonu) şeklinde çalışır. Sadece istemci http bağlantısı oluşturabilir. http bağlantıları tek yönlüdür (half-duplex). http Long Polling, http Streaming, Dynamic Adaptive Streaming over http (DASH) gibi iletişim yöntemleri bulunmaktadır. Bu yöntemler esasında istemcinin sürekli istek atmasına ve sunucunun cevabı parçalı paketlerle göndermesine dayalıdır. Bu şekilde gerçek zamanlı etkileşim hissi verir. Sunucu kontrol arayüzü ve olasılıkla web tarayıcı istemcileri için http web servisi geliştirmekte gereklidir.
- http/1.1; TCP üzerinden çalışır ve her seferde bir paket gönderebilir. Birden fazla gelen isteklere, sırasıyla cevap verebilir. http/2; TCP üzerinden çalışır ve çoklama ile aynı anda birden fazla isteğe sırasız cevap verebilir ancak veri akışının düzenli olması gerektiği durumlarda paket kaybı yaşanır. http/3; QUIC üzerinden, gömülü güvenlik katmanı kullanarak çalışır ve aynı anda birbirinden bağımsız ve sırasız olacak şekilde paketler gönderebilir. Çok daha hızlı ve verimlidir.

1.2.2.3 Socket.io

- Eşler arası ve sunucu-istemci şemalarının ikisini de kullanabilir. Websocket, http ve kendi ağ uygulamalarını kullanır. Gerçek zamanlı iletişim sağlar. Bağlantıları canlı tutmak için Ping-Pong yöntemi kullanır. Karşı taraftan cevap gelmezse, belirli bir süre sonunda Ping gönderen cihaz bağlantıyı kapatır.

1.2.2.4 WebRTC

- Eşler arası ve sunucu-istemci şemalarının ikisini de kullanabilir. Düşük gecikmeli ve gerçek zamanlı iletişim sağlar. ICE, STUN/TURN sunucusu ile bağlantılar sağlanır (Hole Punching).

1.2.3 Kütüphaneler ve Özellikleri

1.2.3.1 C++ REST SDK

- HTTP1.1 client/server, JSON, URI, asynchronous streams, WebSockets client, oAuth.
- Windows, Linux, OS X, Unix, iOS, ve Android.
- Tamamen ücretsiz
- Vcpkg'da mevcut

1.2.3.2 WebRTC

- WebRTC
- Windows, Linux, Android
- Tamamen ücretsiz
- Flutter için doğrudan destek (flutter-webrtc)
- C++ için doğrudan destek
- Vcpkg'da bulunmuyor

1.2.3.3 WebSocketPP

- WebSocket
- Posix/Windows, 32/64bit, Intel/ARM/PPC
- C++ Rest SDK, bu kütüphaneyi de kullanıyor.
- Flutter WebSocket için kütüphane var.
- Vcpkg'da mevcut

1.2.3.4 socket.io-client-cpp

- Socket.io
- %100 modern C++11 ile yazılmış.
- Çapraz platform. Neredeyse her yerde çalışıyor.
- Flutter pub'da mevcut.
- Vcpkg'da mevcut

1.2.3.5 POCO

- Content cache, load balancer, HTTP1.1, Socket, TCP, UDP, FTP, MTP vs.
- Çapraz platform
- Ticari amaçla kullanılması için lisans satın alınması gerekiyor.
- Ticari versiyonunda bulunan bazı özellikler, ücretsiz versiyonda bulunmuyor.
- C++ ama yarısından fazlası C'den oluşuyor :D. Modern C++ derleyicileri projenin her yerinde uyarılar veriyor.
- Kullanımı kolay ve birçok özelliği hazır olarak var
- Yetersiz dokümantasyon
- Blocking IO kullanıyor. Verimliliği düşük.
- Hata karşılamada yetersiz, programcının hataları karşılaması gerekiyor genellikle. Mesela bağlantı beklenmedik bir şekilde sonlandığında program tamamen çöküyor. Üstesinden gelindi ama başka kötü huyları var mı, bilinmediği için kullanmak riskli.
- Vcpkg'da mevcut

1.2.3.6 Proxygen

- HTTP1.1, HTTP2, HTTP3

- Modern C++ ile yazılmış
- Kısıtlı çapraz platform desteği. Windows ve Linux'da çalşıyor
- Çok fazla bağımlılığı var, derlemek uzun sürüyor ve yüksek hafıza tüketiyor
- Performans iyi
- Facebook tarafından kullanılıyor, amaca özel geliştirildiği için dokümantasyonu iyi değil
- Tamamen ücretsiz
- Geniş özelliklere sahip
- Vcpkg'da mevcut

1.2.3.7 Cpp-httplib

- http
- C++11 ile yazılmış
- Çapraz platform
- Sadece SSL desteği sunuyor
- Tek header dosyasından oluşuyor. Boyutu çok küçük ve hafif.
- Kullanımı kolay olduğu iddia ediliyor
- Blocking IO kullanıyor. Performans açısından tatmin edici sonuçlar vermeyebilir.
- Vcpkg'da mevcut