

INF 443
Dağıtık Sistemler ve Uygulamaları
Ödev V

Serhan Daniş

27.10.2017

İçindekiler

1 Basit Mesajlaşma	2
2 Uygulama Protokolü	2
2.1 İstemciden sunucuya mesaj gönderme	3
2.1.1 Yeni bağlantı kurma	3
2.1.2 Bağlantıyı kapat	4
2.1.3 Kullanıcı listesi sorgulama	4
2.1.4 Bağlantı Kontrol	4
2.1.5 Genel mesaj	4
2.1.6 Özel mesaj	5
2.1.7 Hata durumları	5
2.2 Sunucudan istemciye mesaj gönderme	5
2.2.1 Genel mesaj	5
2.2.2 Özel mesaj	6
2.2.3 Sistem mesajı	6
2.2.4 Bağlantı Kontrol	6
3 Metin tabanlı istemci yazma	6

1 Basit Mesajlaşma

Bu çalışmada bilinen mesajlaşma uygulamalarının (Whatsapp, Messenger, Skype, Signal, Telegram, ICQ, IRC) arkasında kullanılan gelişmiş protokollerinin çok basit bir halini uygulayacağız, başka bir deyişle mesajlaşma sunucusu ve istemcisi yazıyor olacağız. Yazacağınız yazılım ikilisi (sistem) aşağıdaki fonksiyonları içerecektir:

- Sunucu ve istemci arasında alt seviye haberleşmeyi sağlamak için bir uygulama protokolü tasarlamak gerekiyor.
- Sistem tek chat odasından oluşacak. Bağlanan kullanıcılar istediklerinde kimlerin bağlanmış olduğunu görebilecekler.
- Gönderilen genel mesajlar bütün bağlı kullanıcılara gönderilecek.
- Kullanıcılar kişiye özel mesaj gönderebilecek. Bu mesaj sadece hedefteki kullanıcıya iletilecek.
- İstemci arayüzünde mesaj gönderme ve mesajları görmek için iki ayrı bölge olacak.
- Sunucu bağlantıları ya da istenilen diğer sistem mesajlarını bir log dosyasına ve/ya ekrana raporlayacak.

2 Uygulama Protokolü

Sunucu ve istemci arasında alt seviye haberleşmeyi sağlamak için bir uygulama protokolü kullanılması gereklidir. Dünyaca kullanılan ve bir konsorsiyum tarafından kabul edilmiş protokol standartları RFC denilen dokümanlarda anlatılmaktadır. Biz çok daha basit sadece sizin için türetilmiş bir protokolü inceleyeceğiz. Burada bütün mesajlar RFC standartlarından uzak sade bir dille anlatılıyor. Bizim amacımız daha çok bu sistemlerin nasıl çalıştığını kavramak olduğu için standart olmayan bir protokol mesaj kümesini inceleyeceğiz. Standart olmamasına rağmen bu protokol mesajları veya komutları halihazırda çalışan bir sistemin komutlarıdır. Sistemin detayları bir sonraki bölümde verilmektedir.

Yukarıda sayılmış gelişmiş sistemlerde (Whatsapp, Skype vb.), bu protokollerin önüne görsel arabirimler eklenir. Bu arabirimlerde bir tuşa basma arka tarafta bir protokol mesajının gönderilmesine neden olabilir. Hatta arabirim sayfasının kapatılması başka bir protokol mesajına denk geliyor olabilir. Ayrıca bu protokol mesajları günümüz mesajlaşma sistemlerinde şifreli gönderilmektedir.

Bizim uygulama protokolümüzde iki bağlantı yönü, dolayısıyla iki farklı protokol yapısına ihtiyacımız var. Biri istemciden sunucuya, diğeri sunucudan istemciye giden mesajlardır. Bunlar aynı görünseler de biraz farklı çalışmaktadırlar (Mesela SAY veya MSG mesajları buna örnektir).

İstemci-sunucu yönlü mesajlaşma soru cevap şeklinde olacaktır ve her mesaj satır sonu karakteri (\n) ile bitirilecektir. Komut kökleri üç karakterden oluşacaktır. Aşağıda protokolün kısa özeti (Tablo 1) ve daha sonra her komutun nasıl kullanılması gerektiği biraz daha detaylı yazılmıştır.

2.1 İstemciden sunucuya mesaj gönderme

Tablo 1: İstemci sorguları için protokol özeti

İstek	Parametre	Cevap	Parametre	Tanım
USR	nickname	HEL	nickname	Yeni kullanıcı kabulü
		REJ	nickname	Yeni kullanıcı reddi
QUI		BYE	nickname	Çıkış
LSQ		LSA	nickname:nick...	Kullanıcı listesi isteme
TIC		TOC		Bağlantı testi
SAY	message	SOK		Genel mesaj gönderme
MSG	nickname:message	MOK		Özel mesaj gönderme
		MNO	nickname	Kullanıcı bulunamadı
Komut		ERR		Hatalı komut
Komut		ERL		Giriş yapılmadı

2.1.1 Yeni bağlantı kurma

Bağlantı kurduktan sonra sunucuya kullanıcı ismi aktarılacaktır. Kullanıcı ismi boşluksuz Türkçe karakter içermeyen bir yapıda olmalıdır. Sunucunun kullanıcı ismini beğenmediği durumlarda veya kullanıcı ismi mevcutsa sunucu girişi reddedip bağlantıyı kapatmalıdır. (<...> şeklindeki yazım değişkenleri belirtmektedir.)

Sorgu: istemci→sunucu

USR <nickname>

Kullanıcıyı kabul et:

Cevap: sunucu→istemci

HEL <nickname>

Böyle bir kullanıcı mevcutsa reddet:

Cevap: sunucu→istemci

REJ <nickname>

2.1.2 Bağlantıyı kapat

Sorgu: istemci→sunucu

QUI

Cevap: sunucu→istemci

BYE <nickname>

2.1.3 Kullanıcı listesi sorgulama

Sorgu: istemci→sunucu

LSQ

Cevap: sunucu→istemci

LSA <colon separated nicknames>

Örnek Cevap: sunucu→istemci

LSA crazy_boy82:stayla90

2.1.4 Bağlantı Kontrol

Sorgu: istemci→sunucu

TIC

Cevap: sunucu→istemci

TOC

2.1.5 Genel mesaj

Sorgu: istemci→sunucu

SAY <message>

Cevap: sunucu→istemci

SOK

2.1.6 Özel mesaj

Sorgu: istemci→sunucu

MSG <nickname>:<message>

Cevap: sunucu→istemci

MOK

Gönderilecek kullanıcı mevcut değilse:

Cevap: sunucu→istemci

MNO <nickname>

2.1.7 Hata durumları

İstemciden sunucuya giriş yapmadan giriş yapma isteği (USR) dışında bir sorgu iletilirse sunucu cevap olarak aşağıdaki mesajı gönderecektir:

Cevap: sunucu→istemci

ERL

İstemciden sunucuya giden ve protokole uygun olmayan mesajlarda istemci aşağıdaki mesajı dönecektir:

Cevap: sunucu→istemci

ERR

2.2 Sunucudan istemciye mesaj gönderme

İstemci→sunucu yönlü mesajlaşma soru cevap şeklinde olacaktır ve her mesaj satır sonu karakteri (\n) ile bitirilecektir. Komut kökleri üç karakterden oluşacaktır. Mesaj detayları aşağıda belirtilmektedir.

2.2.1 Genel mesaj

sunucu sorgusu

SAY <nickname>:<message>

istemci cevabı

SOK

Tablo 2: Sunucu sorguları için protokol özeti

İstek	Parametre	Cevap	Parametre	Tanım
SAY	message	SOK		Genel mesaj iletme
MSG	nickname:message	MOK		Özel mesaj iletme
SYS		YOK		Sistem mesajı iletme
TIC		TOC		Bağlantı testi
Komut		ERR		Hatalı komut

2.2.2 Özel mesaj

sunucu sorgusu

MSG <nickname>:<message>

istemci cevabı

MOK

2.2.3 Sistem mesajı

sunucu sorgusu

SYS <message>

istemci cevabı

YOK

2.2.4 Bağlantı Kontrol

Sorgu: istemci→sunucu

TIC

Cevap: sunucu→istemci

TOC

3 Metin tabanlı istemci yazma

Bu bölümde “telnet” yerine kullanacağınız metin tabanlı bir istemci yazmanız beklenmektedir. İstemci sunucu ile konuşmak için yukarıdaki protokolü kullanırken, kullanıcıdan bilgi almak ve kullanıcıya mesaj iletme için ayrı ve daha basit bir protokol kullanacaktır.

İstemci arayüzünde basit bir komut satırı bulunduğu farzedilir. Bu komut satırı mesaj veya komut yazmaya yarar. Aşağıda yazılabilecek komutlar tanımlanmıştır:

- /n <nickname>: Yeni kullanıcı tanımlama.
- /l: Kanal (sistem) listesi isteme.
- /q: Kanalı (sistemi) terketme.
- /m <nickname> <message>: Özel mesaj gönderme.

Başına “/” koyulmayan her şey genel mesaj olarak algılanacaktır. Komut satırından okunan veriler giden işleyiciden (`outgoing_parser`) geçirilip sokete yazılacak protokol verisi üretilir ve `senderQueue`ya yazılır.

Her istemcinin toplamda üç tane thread’i olması gerekmektedir:

1. `LoggerThread`: Sunucu ile ilgili durumları bir dosyaya kaydeder.
2. `SenderThread`: Sokete veri koyup sunucuya göndermek için kullanılır. Sadece `senderQueue` üzerinden mesaj bekler, gelen verileri `outgoing_parser()`’dan geçirip oluşturulan protokol mesajlarını sokete yazar.
3. `ReaderThread`: Soketten gelen verileri okur, ilgili veri dağıtımlarını yapar. Gelen verileri `incoming_parser()` diye bir method ile işler. Sunucuya verilecekse `senderQueue`’ya yazar ki `SenderThread` karşıya gönderebilsin. Ekranı veri basılacaksa uygun formata dönüştürüp ekrana yazar.