

INF443

Dağıtık Sistemler Proje Tasarım Raporu:

Sunucu Eklentileri Ve Değişiklikleri

Ali Kaan Özden
16401781

Yeni & Değiştirilmiş Protokol Mesajları:

NIC→REG

Açıklama: NIC protokol mesajı kullanıcının kendi rumuzunu sunucuya tanıtması için kullanılmaktaydı. Sunucunun şuan istenen halinde kullanıcı adı ve şifre olacağı için ve bu iki veri kalıcı olarak tutulduğu için bu komut REG ile değiştirildi. REG komutu aynı zamanda oturumunu açmış bulunan kullanıcının şifresini değiştirmek için de kullanılabilir.

1. İstemciden sunucuya oturum açılmamışken REG mesajı:

Sorgu: istemci→sunucu

REG <nickname>:<password>

- Kullanıcı adı ve şifrenin doğru şekilde belirtildiği ve eşsiz oldukları senaryoda:

Cevap: sunucu→istemci

REGOK <nickname>

WEL <nickname>

- Kullanıcı mevcut ise reddet:

Cevap: sunucu→istemci

REJ <nickname>

- Kullanıcı şifresi yanlış şekilde girildiyse (sadece nümerik karakter yerine harf kullanımı vb.):

Cevap: sunucu→istemci

REJPSW <password>

2. İstemciden sunucuya oturum açılmışken REG mesajı:

Sorgu: istemci→sunucu

REG <password>

- Şifre koşullarına uymuyorsa:
Cevap: sunucu→ istemci
REJPSW <password>
- Şifre koşullarına uyuyorsa:
Cevap: sunucu→ istemci
REGOK <password>

LOGIN

Açıklama: REG mesajı ile beraber kaydı alınan kullanıcıların sunucuya tekrar bağlandıklarında kayıtlı hesapları ile bağlanabilmeleri için yeni bir komuta ihtiyaç duyulmaktadır. LOGIN ile kullanıcı kendi hesabına giriş sağlayabilecektir.

1. İstemciden sunucuya oturum açılmamışken LOGIN mesajı:

Sorgu: istemci→sunucu
LOGIN <nickname:<password>

- Başarılı giriş:
Cevap: sunucu→ istemci
WEL <nickname>
- Kullanıcı adı veya şifre yanlış ise:
Cevap: sunucu→ istemci
REJ

2. İstemciden sunucuya oturum açılmışken LOGIN mesajı:

Sorgu: istemci→sunucu
LOGIN <nickname> <password>

Bu durumda otomatik olarak oturumu açık olan hesaptan çıkış yapılır:

Cevap: sunucu→ istemci
BYE <old-nickname>

Protokol (1.) maddede olduğu gibi devam eder.

LOGOFF

Açıklama: Varolan oturumdan çıkış yapmak için eklenmiştir.

1. İstemciden sunucuya oturum açılmamışken LOGOFF mesajı:

Sorgu: istemci→sunucu
LOGOFF

Cevap: sunucu→ istemci
NACC

2. İstemciden sunucuya oturum açılmışken LOGOFF mesajı:

Sorgu: istemci→sunucu
LOGOFF

Cevap: sunucu→ istemci
OKL

RLS

Açıklama: Sunucunun yeni versiyonunda birden fazla chat room olacağı için o anda var olan odaların listesini çıkarmak için RLS komutu eklendi. İstemcinin herhangi bir oturum açıp açmamasından bağımsız olarak RLS komutu oda listesini istemciye getirecektir.

Sorgu: istemci→sunucu
RLS

Cevap: sunucu→ istemci
LSTR <room_name 1>:<room_name 2>: ... :<room_name n>

GNL (updated)

Açıklama: GNL eski sunucu versiyonunda odadaki tüm kullanıcılara mesajı iletme için vardı. Artık birden fazla oda olacağı için GNL ile hangi odanın kullanıcılarına iletileceğini belirtmek gereklidir.

1. İstemciden sunucuya oturum açılmamışken GNL mesajı:

Sorgu: istemci→sunucu
GNL <room_name> : <message>

Cevap: sunucu→ istemci
NACC

- İstemciden sunucuya oturum açılmışken GNL mesajı:

Sorgu: istemci→sunucu

GNL <room_name> : <message>

- Kullanıcı belirtilen odaya giriş yapmışsa:

Cevap: sunucu→istemci

OKG <room_name>

- Kullanıcı belirtilen odaya giriş yapmamışsa:

Cevap: sunucu→istemci

NAR <room_name>

ENTER

Açıklama: Kullanıcının odalara girebilmesi için tanımlanmıştır.

- İstemciden sunucuya oturum açılmamışken ENTER mesajı:

Sorgu: istemci→sunucu

ENTER <room_name>

Cevap: sunucu→istemci

NACC

- İstemciden sunucuya oturum açılmışken ENTER mesajı:

Sorgu: istemci→sunucu

ENTER <room_name>

- Oda bulunduysa:

Cevap: sunucu→istemci

OKEN

Mesaj: sunucu → diğer istemciler

NTF <nickname> entered <room_name>

- Oda bulunmadıysa:

Cevap: sunucu→istemci

NOPR <room_name>

EXIT

Açıklama: Kullanıcının odalardan çıkabilmesi için tanımlanmıştır.

1. İstemciden sunucuya oturum açılmamışken EXIT mesajı:

Sorgu: istemci→sunucu

EXIT <room_name>

Cevap: sunucu→istemci

NACC

2. İstemciden sunucuya oturum açılmışken EXIT mesajı:

Sorgu: istemci→sunucu

EXIT <room_name>

- Belirtilen odaya daha önceden girilmemiş ise:

Cevap: sunucu→istemci

NENT <room_name>

- Belirtilen odaya daha önceden girilmiş ise:

Cevap: sunucu→istemci

OKEX <room_name>

Mesaj: sunucu → diğer istemciler

NTF <nickname> exited <room_name>

GLS (updated)

Açıklama: Kullanıcı listelemek için kullanılan GLS sorgusu artık konuşma odasını parametre olarak almakta ve odaya girilmediği durumda liste çıkmamakta. Aynı zamanda listelenen kullanıcıların yetkileri da listelenmektedir.

1. İstemciden sunucuya oturum açılmamışken GLS mesajı:

Sorgu: istemci→sunucu

GLS <room_name>

Cevap: sunucu→istemci

NACC

- İstemciden sunucuya oturum açılmışken GLS mesajı:

Sorgu: istemci→sunucu

GLS <room_name>

- Belirtilen odaya giriş yapılmışsa:

Cevap: sunucu→istemci

LSTU <nickname 1: authority> : <nickname 2:authority> : ... : <nickname n:authority>

- Belirtilen odaya giriş yapılmamışsa:

Cevap: sunucu→istemci

NENT <room_name>

RLSE

Açıklama: Kullanıcının içinde bulunduğu odaları listelemek için tanımlanmıştır.

- İstemciden sunucuya oturum açılmamışken RLSE mesajı:

Sorgu: istemci→sunucu

RLSE

Cevap: sunucu→istemci

NACC

- İstemciden sunucuya oturum açılmışken GLS mesajı:

Sorgu: istemci→sunucu

RLSE

Cevap: sunucu→istemci

LSTRE <room_name 1>:<room_name 2>: ... :<room_name n>

CREATR

Açıklama: Kullanıcının yeni oda kurabilmesi için tanımlanmıştır.

- İstemciden sunucuya oturum açılmamışken CREATR mesajı:

Sorgu: istemci→sunucu

CREATR <room_name>

Cevap: sunucu→istemci

NACC

2. İstemciden sunucuya oturum açılmışken CREATR mesajı:

Sorgu: istemci→sunucu

CREATR <room_name>

- Belirtilen oda ismi odalar listesinde yoksa:
Cevap: sunucu→istemci
OKCR <room_name>
- Belirtilen oda ismi odalar listesinde zaten var ise:
Cevap: sunucu→istemci
REJ <room_name>

KICK

Açıklama: Kullanıcının yöneticisi olduğu odadan başka bir kullanıcıyı atabilmesi için tanımlanmıştır.

1. İstemciden sunucuya oturum açılmamışken KICK mesajı:

Sorgu: istemci→sunucu

KICK <room_name>:<nickname>

Cevap: sunucu→istemci

NACC

2. İstemciden sunucuya oturum açılmışken KICK mesajı:

Sorgu: istemci→sunucu

KICK <room_name>:<nickname>

- Belirtilen oda bulunmakta ise:
 - Kullanıcı belirtilen odanın yöneticisi ise:
 - Atılmak istenen kullanıcı belirtilen odaya üye ise:
Cevap: sunucu→istemci
OKKICK <nickname>

Mesaj: sunucu→ atılan kullanıcı

KICKED <room_name>

Mesaj: sunucu→ diğer kullanıcılar

NTF <nickname> kicked from <room_name>

- Atılmak istenen kullanıcı belirtilen odaya üye değil ise:

Cevap: sunucu→ istemci

NOP <nickname>

- Kullanıcı belirtilen odanın yöneticisi değil ise:

Cevap: sunucu→ istemci

NOWN <room_name>

- Belirtilen oda bulunmamakta ise:

Cevap: sunucu→ istemci

NOPR <room_name>

CLSR

Açıklama: Kullanıcının yöneticisi olduğu odayı kapatabilmesi için tanımlanmıştır.

1. İstemciden sunucuya oturum açılmamışken CLSR mesajı:

Sorgu: istemci→sunucu

CLSR <room_name>

Cevap: sunucu→ istemci

NACC

2. İstemciden sunucuya oturum açılmışken CLSR mesajı:

Sorgu: istemci→sunucu

CLSR <room_name>

- Belirtilen oda bulunmakta ise:
 - Kullanıcı belirtilen odanın yöneticisi ise:

Cevap: sunucu→ istemci

OKCLS <room_name>

Mesaj: sunucu→ diğer kullanıcılar

NTF <room_name> is closed by <owner-nickname>

Mesaj: sunucu→ diğer kullanıcılar

EXITED <room_name>

- Kullanıcı belirtilen odanın yöneticisi değil ise:

Cevap: sunucu→ istemci

NOWN <room_name>

- Belirtilen oda bulunmamakta ise:

Cevap: sunucu→ istemci

NOPR <room_name>

Aşağıdaki tabloda yeni eklenen protokol mesajları ve eski protokol mesajlarının özeti bulunmaktadır.

İstek	Parametre	Cevap	Parametre	Tanım
REG	<nickname>:<password>	REGOK	<nickname>	Yeni hesap kabulü
			<password>	Yeni şifre kabulü
		REJ	<nickname>	Yeni hesap reddi
		REJPSW	<password>	Yanlış şifre tanımı
		WEL	<nickname>	Başarılı giriş
LOGIN	<nickname>:<password>	WEL	<nickname>	Başarılı giriş
		REJ		Kullanıcı adı veya şifre yanlış
		BYE	<old-nickname>	Oturum açıkken yeni hesap yaratılırsa eski hesap kapanışı
LOGOFF		NACC		Oturum açılmamış
		OKL		Oturum kapandı
RLS		LSTR	<room_name 1>:<room_name 2>:...	Oda listesi
GNL	<room_name> :<message>	NACC		Oturum açılmamış
		OKG	<room_name>	Mesaj gönderildi
		NAR	<room_name>	Belirtilen odaya giriş yapılmamış
ENTER	<room_name>	NACC		Oturum açılmamış
		OKEN		Başarılı oda giriş
		NTF	<nickname> entered	Diğer kullanıcılara bildir

			<room_name>	
		NOPR	<room_name>	Oda bulunamadı
EXIT	<room_name>	NENT	<room_name>	Odaya daha önceden giriş yapılmamış
		OKEX	<room_name>	Odadan çıkış onayı
		NTF	<nickname> exited <room_name>	Çıkış bildirimi
GLS	<room_name>	NACC		Oturum açılmamış
		LSTU	<nickname 1: authority> : <nickname 2:authority> : ...	Kullanıcı listesi
		NENT	<room_name>	Odaya daha önceden giriş yapılmamış
RLSE		NACC		Oturum açılmamış
		LSTRE	<room_name 1>:<room_name 2>: ...	Dahil olunan oda listesi
CREAT R	<room_name>	NACC		Oturum açılmamış
		OKCR	<room_name>	Oda kuruluş onayı
		REJ	<room_name>	Oda ismi alınmış
KICK	<room_name>:<nickname>	NACC		Oturum açılmamış
		OKKICK	<nickname>	Kullanıcı başarı ile atıldı
		KICKED	<room_name>	Atılan kullanıcıya kick bildirimi
		NTF	<nickname> kicked from <room_name>	Diğer kullanıcılara kick bildir
		NOP	<nickname>	Atılmak istenen kullanıcı bulunamadı
		NOWN	<room_name>	Oda yöneticisi değil
		NOPR	<room_name>	Oda bulunamadı
CLSR	<room_name>	NACC		Oturum açılmamış
		OKCLS	<room_name>	Oda kapanışı onaylandı
		NTF	<room_name> is closed by <owner-nickname>	Oda kapanışı bildirimi
		EXITED	<room_name>	Kapanan odanın kullanıcıları çıkarıldı
		NOWN	<room_name>	Oda kapatmak için yetki yok
		NOPR	<room_name>	Oda bulunamadı
QUI		BYE	[nick_name]	Sunucu bağlantısı koparıldı
PIN		PON		Pin cevabı
PRV	<nick_name>:<message>	OKP		Özel mesaj onayı

		NOP	<nick_name>	Özel mesaj için kullanıcı bulunamadı
KOMUT		ERR		Hatalı komut

Yeni & Değiştirilmiş Veri Yapıları:

Yeni yapıları belirlemeden önce sunucunun bir önceki versiyonunda bulunan yapılardan bahsederek:

```
clientDictionary = {} # clientConnection : [ReadThread, WriteThread, connectionQ]
userNames = {} # userName : clientConnection
userNamesLock = threading.Lock()
clientDictLock = threading.Lock()
loggerQ = queue.Queue()
```

Yukarıdaki yapılar bulunmaktadır.

Artık tek bir konuşma odası olmadığı için konuşma odalarını tutan ayrı bir dictionary gereklidir.

```
room_dict = {room_name: [owner, user_list]}
```

Aynı zamanda kullanıcılar artık kayıtlı oldukları için userNames üzerinde çeşitli değişiklikler yapılması gerekmektedir ve yeni sözlüklere ihtiyaç bulunmaktadır .

```
clientDictionary = {clientConnection : [ReadThread, Writethread, connectionQ]}
user_authentication = {username:password}
users_with_conections = {username:clientConnection}
users_with_roomOwnership = {username:room_name}
users_with_roomMembership = {username:room_name}
```

Yukarıdaki sözlük yapıları ile istenilen kullanıcı hakkında istenilen veriye ulaşmak mümkün. Fakat sözlüklere erişimin kullanıcılar ile ilgilenen threadler arasında senkron bir şekilde gerçekleşmesi gerektiği için yeni lock değişkenlerine ihtiyaç bulunmaktadır. Aşağıda verilen tüm lock değişkenlerine ihtiyacın olup olmadığını ancak kodu yazarken anlamam mümkün, bazılarını çıkarabilirim.

```
user_auth_lock = threading.Lock()
users_with_conn_lock = threading.Lock()
users_with_roomOwn_lock = threading.Lock()
room_dict = threading.Lock()
```

Çalışma Senaryoları:

- a. Yeni kullanıcı kaydında artık NIC yerine REG komutu kullanılıyor. REG mesajı istemci tarafından gönderildiğinde ilgili thread ön koşulları kontrol ettikten sonra *user_authentication* , *users_with_conections* alanlarına sırayla erişerek gerekli verileri senkronluğa dikkat ederek giriyor ve kullanıcı ana sisteme alıyor. *clientDictionnary* alanı sonsuz sunucu döngüsünde ilgili thread istemciye atanırken işleniyor.
- b. Sunucuya giriş yapmak için bağlandıktan sonra LOGIN mesajı kullanılarak istenilen oturuma erişim sağlanıyor. Bu sırada sorumlu thread *user_authentication* sözlüğünde parametreleri doğruladıktan sonra *users_with_conections* sözlüğünde o andaki bağlantıyı giriş yapılan oturum ile eşleştiriyor.
- c. Konuşma odalarını listelemek için istemci RLS mesajını gönderdikten sonra ilgili thread *room_dict* sözlüğünden anahtarları çekerek ilgili kuyruğa senderThread tarafından yollanması için liste halinde koyuyor.
- d. Konuşma odası eklemek için kullanıcının oturum açmış olması şartını sağladıktan sonra CREATR ile istenilen oda ismini vererek odayı konuşma odasını eklemesi mümkün. Bu esnada *room_dict* sözlüğünden o isme sahip başka bir oda olup olmadığı kontrol ediliyor ve eğer bulunamazsa oda kurma işlemi onaylanıp aynı sözlüğe yeni oda ekleniyor. Diğer durumda ise kullanıcıya REJ ile oda isminin alındığı söyleniyor.
- e. Konuşma odasına giriş yapmak için ENTER <room_name> komutunu kullanmak gerekiyor. İlgili oda ismi *room_dict* içinde bulunduğu takdirde bu odanın kullanıcı listesine yeni kullanıcı ekleniyor ve işlem tamamlanıyor. Oda ismi bulunamazsa NOPR başarılı giriş olursa OKEN mesajları geri döndürülüyor. OKEN mesajının ardından oda üyelerine yine aynı sözlük üzerinden erişilerek NTF mesajı bildirim amaçlı gönderiliyor.
- f. Konuşma odasına genel mesaj atmak için GNL <room_name> komutu kullanılmalı. Bir önceki GNL versiyonundan farklı olarak artık oda ismi belirtmek zorunlu. Bir önceki maddede olduğu gibi ilgili sözlükten oda ismi aranıyor, bulunursa ve ilgili kullanıcı o odaya daha önceden giriş yapmış ise odanın diğer kullanıcılarına mesaj iletiliyor. Odaya daha önceden giriş sağlanamadığı takdirde NAR mesajı, oda bulunmadığı takdirde NOPR mesajı geri gönderilmektedir. Odaya mesaj yollamak için aynı sözlükte bulunan kullanıcı adları çekilip bu kullanıcıların bağlı olduğu connection değişkenlerini bulunduran sözlüğe yani *users_with_conections* ve *clientDictionnary* erişimler sağlanarak ilgili thread kuyruklarına mesajlar bırakılıyor.

- g. GLS olarak bir önceki versiyonda kullandığımız kullanıcı listeleme komutu artık ilgili odaya giriş yapmış olma ön koşulunu istemektedir. Yukarıdaki maddelerdeki gibi *room_dict* sözlüğünden ilgili doğrulama işlemleri gerçekleştirildikten sonra liste istekte bulunan istemciye geri döndürülüyor. İstemci odaya daha önce girmemiş ise NENT cevabı ile sunucu tarafından bu durum bildirilmektedir.
- h. RLSE komutu ile içinde bulunulan konuşma odalarını listelemek mümkün. Bunun için kullanıcının ilk önce oturum açmış olması gerekmekte. Bu koşul sağlanmadığı takdirde NACC mesajı, sağlandığı takdirde *users_with_roomMembership* sözlüğünden bulunan odalar listelenmiş bir halde LSTRE ile kullanıcıya geri döndürülmektedir.
- i. Konuşma odasından çıkmak için EXIT komutunu kullanmak gereklidir. İlgili odaya zaten giriş yapılmamışsa NENT, giriş yapılmışsa ve başarı ile çıkmışsa OKEX mesajları ile sunucu geri cevap verir. OKEX mesajının gönderildiği durumda NTF ile diğer kullanıcılara bu aksiyon bildirim olarak gönderilir. (e) maddesinde belirtilen sözlük erişimleri aynı şekilde ilgili thread tarafından yapılır.
- j. Konuşma odasından bir kullanıcıyı atmak için KICK komutu kullanılır. Bunu yapabilmek için kullanıcının ilgili odanın sahibi olması gerekir. Bu koşulu doğrulamak için ilgili thread *users_with_roomOwnership* sözlüğüne danışır. Odadan atılmak istenen kullanıcı *room_dict* sözlüğünde bulunamazsa NOP mesajı döndürülür. Eğer kullanıcıyı atmaya çalışan istemci oda sahibi değilse NOWN mesajı döndürülür. Bir kullanıcı birden fazla odanın sahibi olabildiği için KICK komutuna parametre olarak kullanıcı adıyla beraber oda ismi de verilmelidir. Oda ismi *room_dict* sözlüğünde bulunamazsa NOPR mesajı döndürülür. Kullanıcın tüm koşulları sağladığı durumda oda sahibine OKKICK onay mesajı dönerken atılmış olan kullanıcıya KICKED <room_name> mesajı döner ve diğer tüm oda üyelerine NTF ile bu aksiyon bildirilir. Atılma işlemi başarı ile gerçekleştiği takdirde *room_dict* sözlüğünde ilgili oda bulunur ve kullanıcı listesinden atılmış olan kullanıcı çıkartılır.
- k. Konuşma odasına bir kullanıcının girişini engelleme yani BAN atma komutu bu raporda BONUS olarak verildiği için belirtilmemekte. Eğer zamanım yeterse proje kodunu yazarken bu implementasyonu da yapacağım.
- l. Konuşma odasını kapatmak için (j) maddesinde olduğu gibi odanın sahibi olmak gereklidir. Bu koşullar (j) maddesindeki sağlandığı zaman CLSR ile oda sahibi odayı kapatabilir. Bu durumda ilgili sözlük olan *room_dict*'ten oda silinirken aynı zamanda *users_with_roomOwnership* ve *users_with_roomMembership* sözlüklerinden ilgili odaya dair kullanıcıların kayıtları silinir. İşlem başarı ile gerçekleştiğinde oda sahibine OKCLS mesajı dönerken diğer oda üyelerine NTF ile oda sahibinin odayı kapattığı bildirilir ve ardından EXITED <room_name> ile odadan çıktıklarını belirtir. Koşullar sağlanamadığı takdirde (j) şıkkındaki aynı exception mesajları döndürülür.

- m. Başka kullanıcıya özel mesaj atmak için kullanılan PRV komutu bir önceki versiyon ile aynıdır. Tek değişen şey: GLS ile tüm kullanıcı listesini almak mümkün olmadığı için kullanıcı adı bilinmediği takdirde gönderilecek kullanıcı ile ilk önce aynı odada bulunmak ve o odadayken GLS ile kullanıcı listesini çekmek gereksinimin ortaya çıkmasıdır. Proje kodunu yazarken bu durumu kolaylaştıracak bir protokol eklemesi yapabilirim.

