

CS360 Lab #8 -- Threaded Jtalk

- [Jim Plank](#)
- [CS360](#)

Jtalk'u threaded program'a dönüştürüyoruz ve server'ı güçlendiriyoruz.

Öncelikle dizini inceleyin, [/home/huangj/cs360/labs/lab8/](#) burada iki program göreceksiniz.

Birincisi olan jtalk.c., jtalk için 'client'ınız olacak. 'Command line'da 'host' ve 'port number' yer alıyor. Öncelikle bir 'user name' soruyor, daha sonra 'host/port number'da yer alan 'socket' için bir bağlantı istiyor. Bağlantı yapıldığı anda ayrı 'threadler'birleşir. Ana 'thread' satırları standart 'input'tan okur, isimleri onların başına ekler ve 'send string' kullanarak onları 'server'a gönderir. 'send_string'in yaptığı şey şudur: 'string'in uzunluğunu gönderir ve daha sonra kendisini gönderir. Bu, 'server'ın, yeni satırlar aramaya gerek kalmadan 'input'ları ayrıştırmasını kolaylaştırır.

Diğer 'thread' (**from_socket()**) 'string'leri 'socket'ten okur ve standart bir 'output'a yazar. Bunu yapmak için **read_string()** kullanır. Tahmin edebileceğiniz üzere, **read_string()** bir 'string'in uzunluğunu ve daha sonra 'string'in kendisini okur.

'Socket' veya 'standart input' kapandığında **jtalk** ortaya çıkar.

cat_server.c. ise bir 'socket' hizmeti veren basit bir programdır. Bağlantı yakaladığı anda tüm 'input'u kolaylıkla bağlantıdan bağlantıya gönderip alır. Jtalk'u, bir makine üzerinde **cat_server**'ı çalışırken ve jtalk bağlıken test edebilirsiniz. Jtalk penceresine yazdığınız her şey yazılanların başına ilave edilen isim ile çağrılmak zorundadır . Devam etmeden önce jtalk ve cat_server hakkında herşeyi anladığınızdan emin olun.

jtalk_server

Evet, şimdi işiniz bir jtalk'ın 'threaded server'ı olan jtalk server'ı yazmak. jtalk_server 'thread'leri kullanmalıdır, **select()** kullanmamalıdır. **Jtalk_server**, 'command line'da 'host' ve 'port'alacak ve bu host/port üzerinde bir 'socket' hizmete sunacaktır. 'Socket' hizmete girdiği anda, herhangi bir sayıda **jtalk** client bağlanabilir ve bir client ne yazarsa diğer tüm clientler tarafından görülür. Eğer bir client çıkar veya diğer clientler katılırsa herşey çalışır olmalıdır.

'Threads' hayatınızı kolaylaştırır. **accept_connection()**'ı çağıran bir 'thread'iniz olmalı. Bağlantı yakaladığı anda bu bağlantıyı sunan başka bir 'thread'i tetikler. Açıkçası, tüm 'thread'ler tarafından paylaşılan birkaç data yapısına ihtiyaç duyacaksınız böylece herhangi bir 'thread'tüm bağlantılara yazılabilir.

Bu data yapısını koruyan bir ‘mutex’e ihtiyacınız olacağını unutmayın. Bunun sebebi, bir ‘thread’ tüm bağlantılarınıza yazıyor olacakken bir diğer ‘thread’in eş zamanlı olarak çıkış yapacak olması. Bunu aklınızdan çıkarmayın. Ayrıca bu ‘sigpipe’ ile ne yapacağınızı da düşünün.

Son olarak ``jtalk console"u çalıştıran ekstra bir ‘thread’e sahip olmalısınız. Bu ‘thread’ ‘standart output’a bir prompt yazar ve sonrasında ‘standart input’taki komutları kabul eder.

1. ‘Standart input’ kapalıysa ‘server’ı yok edebilir.
2. Command ``TALKERS" ise ‘server’a bağlı olan tüm kişilerin listesini yazdırmalıdır.
3. Command ``ALL" ise, ‘server’ın sahip olmuş olduğu tüm bağlantıların bir listesini yazdırmalıdır.

Çalışan bir uygulamayı şurada bulabilirsiniz: [/home/cs360/lab8/](#) Kodlamaya başlamadan önce deneyerek neler olduğunu kolaylıkla görebilirsiniz.

Hints (İpuçları)

Açıkçası, **accept_connection()**’ı çağıran bir ‘thread’e, konsol için bir ‘thread’e ve her bağlantı için ayrı bir ‘thread’e ihtiyacınız olacak. Ben, bağlantı sayısı ile şifrelenen kırmızı-siyah ağaç (her bir yeni bağlantı yapıldığında görevlendirilecek şekilde) ve mevcut/geçmiş tüm bağlantılar için ikinci bir kırmızı-siyah ağaç kullandım.

Hiçbir zaman **pthread_join()**’i çağırmadım ancak bir bağlantıyı koptuğu zaman **pthread_exit()**’i kullandım.

Print zamanlamaları için **ctime()**’ı kullanın.

Doğru şekilde disconnect olmak için **send_string()** v.b. modifiye etmeniz gerekli. Buna rağmen **jtalk**’u değiştiremezsiniz -- sunucunuz benim sağlamış olduğum jtalk ile çalışıyor olmalı.

Şunu farz etmelisiniz ki eğer bir ‘socket’te **SIGPIPE** yakalarsanız, bu soketteki tüm **read()** çağrıları anında sıfır değerine ulaşacaktır. Eğer tüm yapıyı doğru şekilde kurarsanız bu hayatınızı kolaylaştıracaktır.

Aşamalı olarak çalışıp aşamalı olarak test edin.