

# CSE 437 REAL TIME SYSTEM ARCHITECTURES

## HOMEWORK 1

Batuhan TOPALOĞLU

151044026

RAPOR

MyTimer sınıfı ITimer arayüzünde bulunan fonksiyonların gerçekleşmesi için oluşturulmuş bir sınıftır. MyTimer sınıfının ITimer arayüzünde bulunan fonksiyonlar hariç kullanıcının kullanımına açık başka bir fonksiyonu bulunmamaktadır.

MyTimer sınıfı kendi üzerine yüklenen işleri halletmek için kullanacağı thread'i kendi oluşturmaktadır o yüzden kullanıcının bu konuda bir işlem yapılmasına gerek yoktur.

MyTimer sınıfının registerTimer fonksiyonlarının varyasyonlarına verilen callback fonksiyonlarının gerçekleşme sürelerinin programın akışını bozmamak üzere küçük işler yapan ve muhakkak bir şekilde return ederek geri dönen fonksiyonlar olduğu varsayılmış. Aksi halde MyTimer sınıfının thread'i o fonksiyonda takılı kalacaktır.

MyTimer'ın çalışma şeklini genel hatları ile aktaracak olursak, ilk olarak constructor ile kendi üzerine düşen işlemleri halletmek üzere( queue traverse, run callbacks) bir thread oluşturur.

MyTimer sınıfı registerTimer fonksiyonu her çağrıldığında parametreler ile bilgileri kullanarak içerisinde kurulu tüm timerların olduğu queue'ya ekler. Bu queue içerisinde struct 'entry' tipinde veriler tutar. Struct entry içerisinde ise timer ile ilgili bilinmesi gereken bulunur(timepoint,perioda,...). Timer thread'i bu queue üzerinde sürekli gezerek vakti gelen timerların callback'lerini çalıştırır. Timer'ın tipine göre queue dan çıkarılıp çıkarılmayacağına karar verir ve ileride tekrar çağıracaksa işlediği timer'ı queue'nun başından alıp en sonuna koyar.

Queue üzerindeki hem registerTimer fonksiyonları hem de MyTimer'ın thread fonksiyonu olan 'loop' çalıştığı için iki taraf arasında bir senkronizasyon mekanizma bulunması gerekiyordu. Bunu mutex, condition variable kullanılarak çözüldü.

Queue boş olduğu sürece yani kurulu herhangi bir timer olmadığı zamanlarda MyTimer thread'i bir condition variable ile beklemeye alınır ve thread uyumuş olur. Bu condition variable'ı spurious wakeup'tan korumak için ek olarak bir boolean 'ready' değişkenini kullanır.

Queue'ya erişimi kontrol etmek içinde mutex kullanılır.

MyTimer sınıfının fonksiyonları test etmek için bir test dosyası hazırlandı. Burada registerTimer fonksiyonları farklı parametreler ile çağrılmaktadır. Callback fonksiyonları terminale yazan basit lambda expressionlar verilmiştir.

Projenin dosya hiyerarşisi:

```
| - batuhan_topaloglu_151044026
| - CmakeLists.txt
| - test.cpp          /*sınıfın test dosyası burada olmalı*/
| - includes
|   | - MyTimer.hpp
|   | - ITimer.hpp
| - sources
|   | - MyTimer.cpp    /*MyTimer sınıfının implementasyonu burada
| - build
|   | - bin
|     | - MyTimer.exe  /*test.cpp'nin executable'ı MyTimer buraya
çıkartılır.*/
|   | - MakeFile       /*makefile burada bulunur */
```

1-Projenin build edilmesi için ana klasörün altındaki **build** klasörü içerisinde terminal açılarak “**cmake ..**” komutu çalıştırılmalıdır.

2- MakeFile **build** klasörü altında oluşur. Terminalde build dizininde “**make**” komutu çalıştırılır.

3- **build** klasörü altındaki **bin** klasörü içerisinde executable olarak MyTimer dosyası oluşacaktır. Bu test.cpp'nin executable'ıdır. O dizinde “**./MyTimer.exe**” komutu çalıştırıldığında proje çalışmaya başlayacaktır.