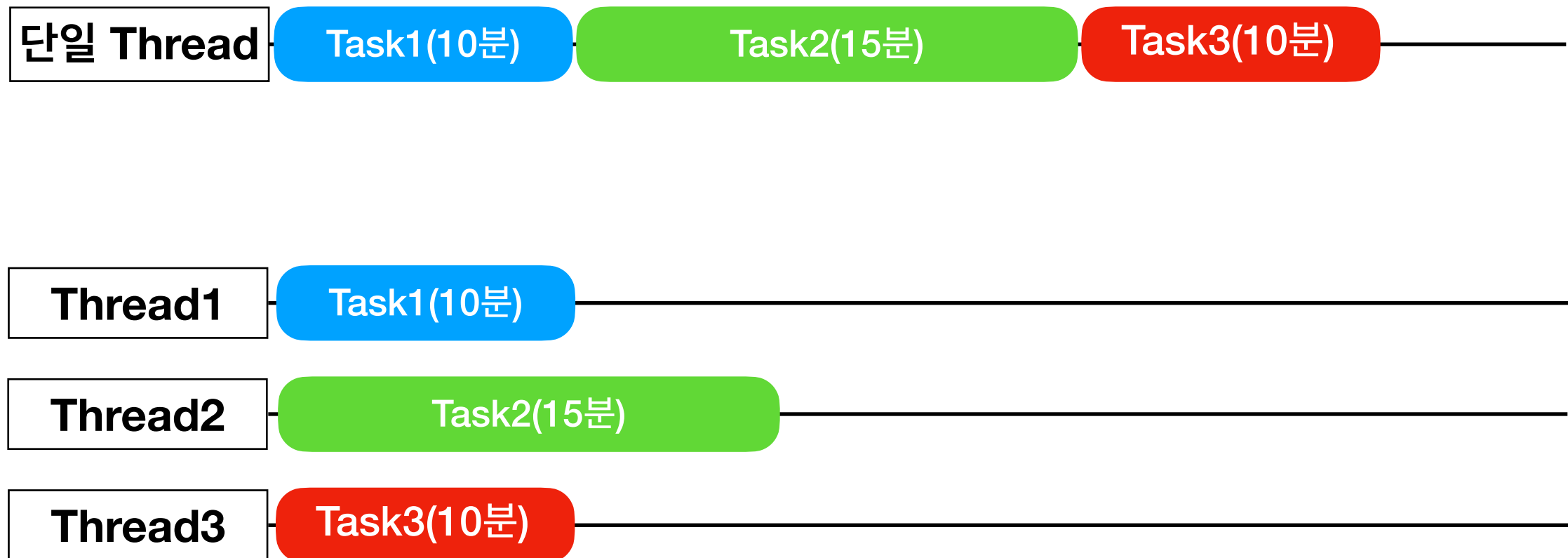


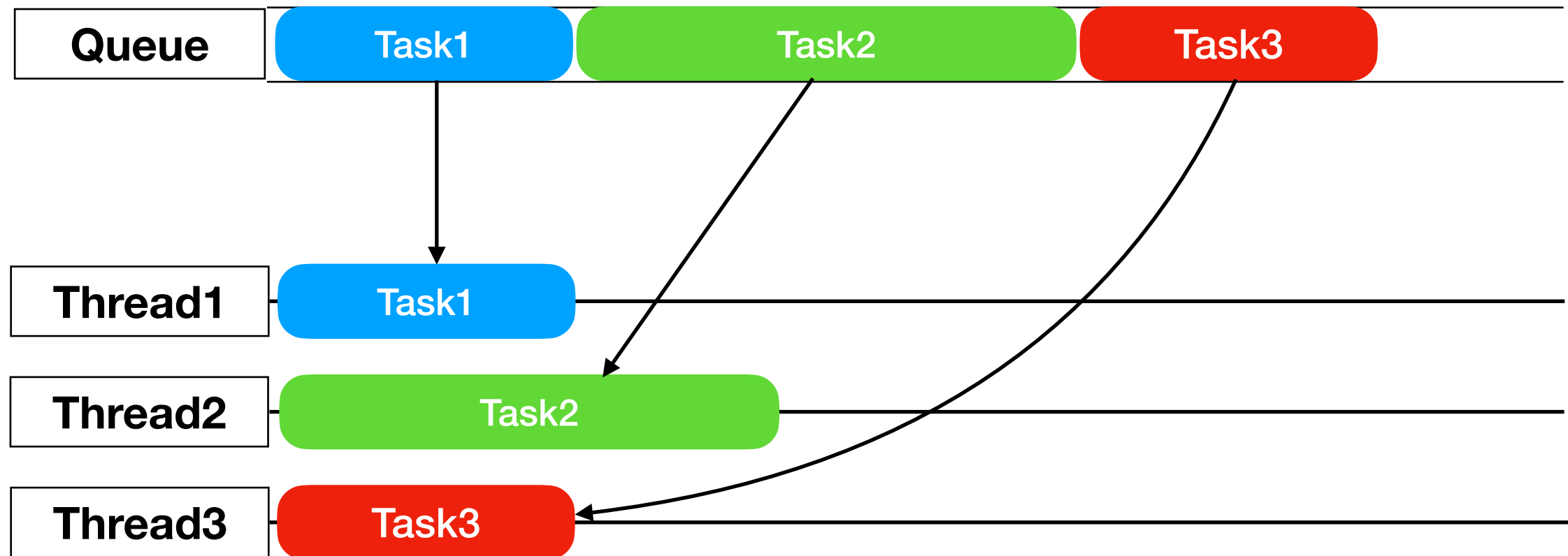
Grand Central Dispatch(GCD)

Thread



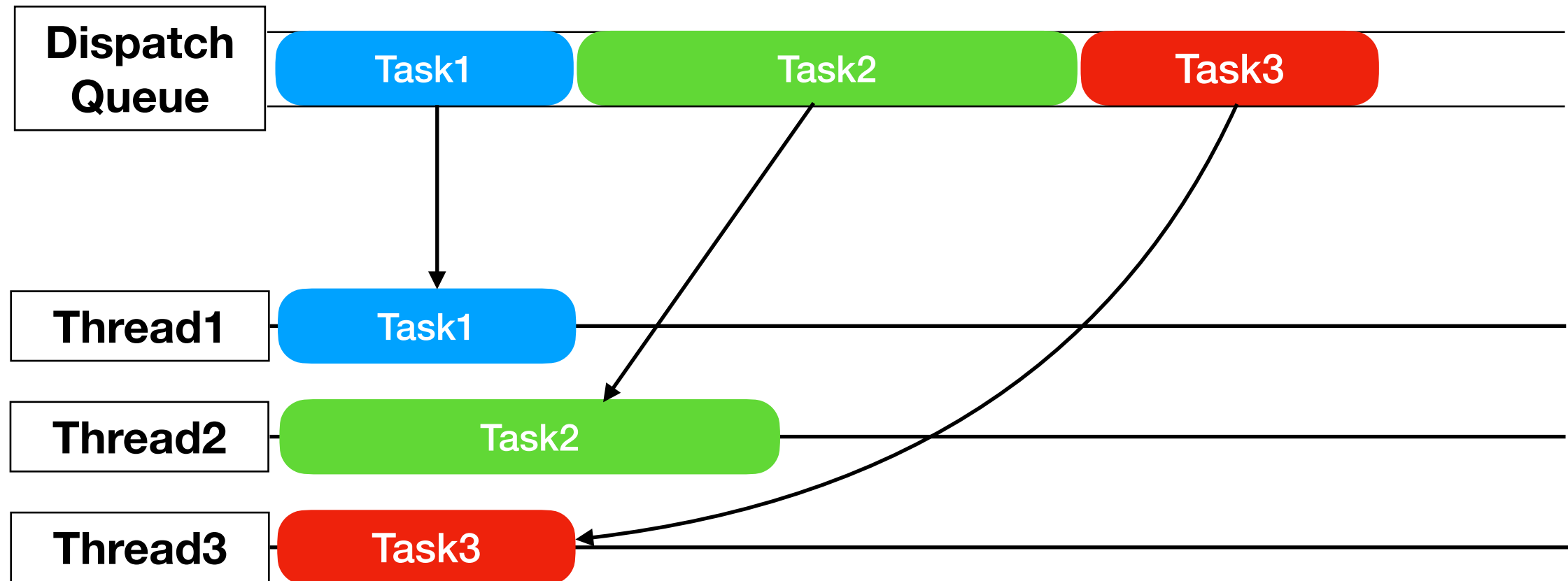
Queue

iOS에서는 작업(Task)를 Queue에 보내기만 하면 알아서 분배



Dispatch Queue

Queue에 작업을 보내면 GCD가 작업을 분배
GCD가 사용하는 Queue의 이름이 Dispatch Queue



Dispatch Queue

DispatchQueue의
종류

작업의 방식
동기, 비동기

```
DispatchQueue.main.async {  
    self.tableView.reloadData()  
}
```

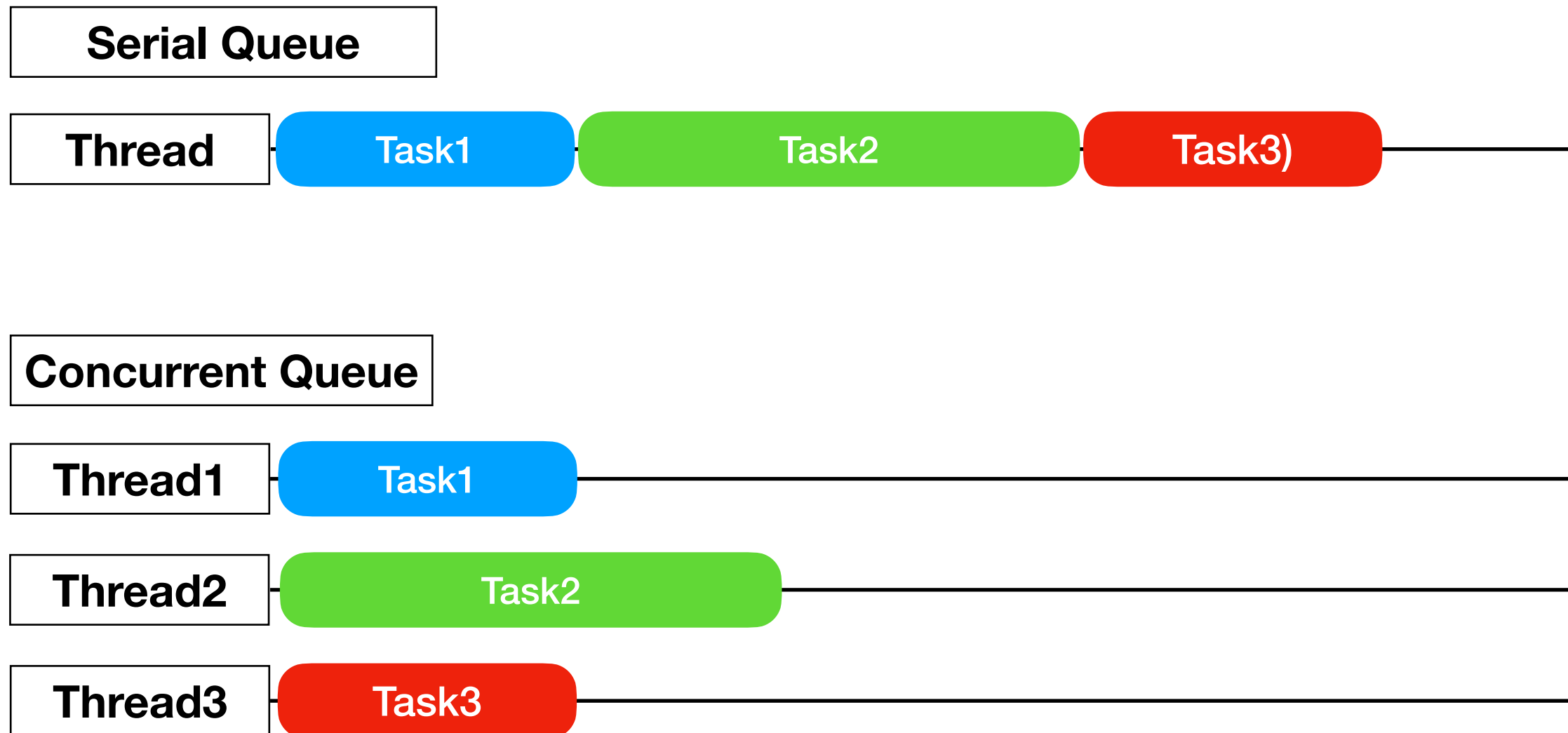
sync, async

```
DispatchQueue.main.sync {  
    self.tableView.reloadData()  
}
```

```
DispatchQueue.main.async {  
    self.tableView.reloadData()  
}
```

Serial Queue

Concurrent Queue



DispatchQueue의 종류

- MainQueue
 - 유일함
 - Main Thread 에서 처리함
 - Serial Queue
- GlobalQueue
 - Quality of Service(QoS)에 따라 6개로 구분
 - Concurrent Queue
- CustomQueue

Quality of Service

- `DispatchQoS.userInteractive`
 - 애니메이션 또는 UI 업데이트에 사용
- `DispatchQoS.userInitiated`
 - API에서 데이터를 로드하는 것과 같은 작업에 사용
- `DispatchQoS.default`
 - 기본적인 작업에 사용
- `DispatchQoS.utility`
 - 사용자가 추적할 필요가 없는 작업에 사용
- `DispatchQoS.background`
 - 로컬 데이터베이스에 데이터를 저장하거나 우선 순위가 높지 않은 유지 관리 코드와 같은 작업에 사용
- `DispatchQoS.unspecified`

Global Queue

```
DispatchQueue.global(qos: .background).async {  
    <#code#>  
}
```

```
DispatchQueue.global().async {  
    <#code#>  
}
```

```
DispatchQueue.global(qos: .background).async(qos:.utility) {  
    <#code#>  
}
```

Global Queue

시간이 오래걸리는 작업들은 Main Thread에서 작업해도
자동으로 global Queue로 Dispatch됨

```
let session = URLSession.shared  
  
let task = session.dataTask(with: request,  
                             completionHandler:dataTaskHandler)  
  
task.resume()
```