



Foundation Framework

Quản lý bộ nhớ





Nội dung

- Autorelease Pool
- Reference Counting
- Tóm tắt những luật trong việc quản lý bộ nhớ
- Vòng lặp sự kiện và cấp phát bộ nhớ
- Rò rỉ bộ nhớ
- Garbage Collection



Autorelease Pool

- Đây là pool nơi mà hệ thống theo dõi các đối tượng cho việc release sau này.

`NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];`

- Khi pool được thiết lập, Foundation sẽ tự động thêm các đối tượng mảng, chuỗi và các đối tượng khác vào pool này. Khi sử dụng xong, có thể giải phóng bộ nhớ mà nó sử dụng bằng cách gửi message như sau:

`[pool drain];`

- Chúng ta có thể thêm một đối tượng vào autorelease pool hiện tại để cho việc release sau này bằng cách gửi nó một message autorelease:

`[myFraction autorelease];`



Reference Counting

- Foundation Framework cung cấp một giải pháp toàn diện cho việc theo dõi số lượng tham chiếu đến một đối tượng --> Reference Counting.
- Khi một đối tượng được tạo, nó sẽ thiết lập số tham chiếu đến đối tượng đó là 1.
- Để tăng số lượng tham chiếu đến một đối tượng lên 1 thì đối tượng đó gửi message đến retain
[myFraction retain];
- Chú ý rằng: Một số phương thức trong Foundation Framework cũng tăng số tham chiếu đến một đối tượng (như một đối tượng thêm vào một array)



Reference Counting (tt)

- Khi một đối tượng không còn sử dụng nữa, bạn có thể giảm số tham chiếu đến đối tượng này đi 1 bằng cách gửi message đến release
- Khi một đối tượng có số tham chiếu đến là 0 thì hệ thống sẽ biết rằng đối tượng này không còn cần thiết nữa nên nó sẽ thu hồi memory đã cấp phát cho đối tượng này. Điều này có thể làm bằng cách gửi message đến dealloc

Tóm tắt những luật trong việc quản lý bộ nhớ

- Gửi message đến release không thật sự huỷ một đối tượng. Khi số tham chiếu đến đối tượng bằng 0 thì đối tượng đó mới bị huỷ, hệ thống sẽ gửi message đến dealloc để thu hồi memory đã cấp cho đối tượng này.
- Nếu cần phải giữ một đối tượng mà bạn không muốn nó bị huỷ bởi một đối tượng nào đó, bạn sẽ gửi message đến retain.
- Release bất kỳ đối tượng nào đó mà bạn đã gửi message đến retain hoặc đã sử dụng gửi message đến copy, mutable-copy, alloc, new
- Autorelease pool cung cấp một cơ chế release một cách tự động cho các đối tượng

Tóm tắt những luật trong việc quản lý bộ nhớ

- Nếu bạn không cần một đối tượng từ bên trong một phương thức nhưng cần trả về “giá trị” của nó thì gửi nó một autorelease message
- Khi tắt ứng dụng, tất cả memory đã cấp phát cho các đối tượng cần phải được thu hồi cho dù nó có nằm trong Autorelease pool hay không
- Đối với các ứng dụng iOS, autorelease pool sẽ được tạo và thu hồi memory của các đối tượng trong quá trình ứng dụng đang thực thi

Vòng lặp sự kiện và cấp phát bộ nhớ

```
@interface myView : UIView {  
    NSMutableArray *data;  
}  
@property (nonatomic, retain) NSMutableArray *data;  
@end
```

```
- (void) loadView { // or viewDidLoad  
    //...  
    data = [NSMutableArray array];  
}
```

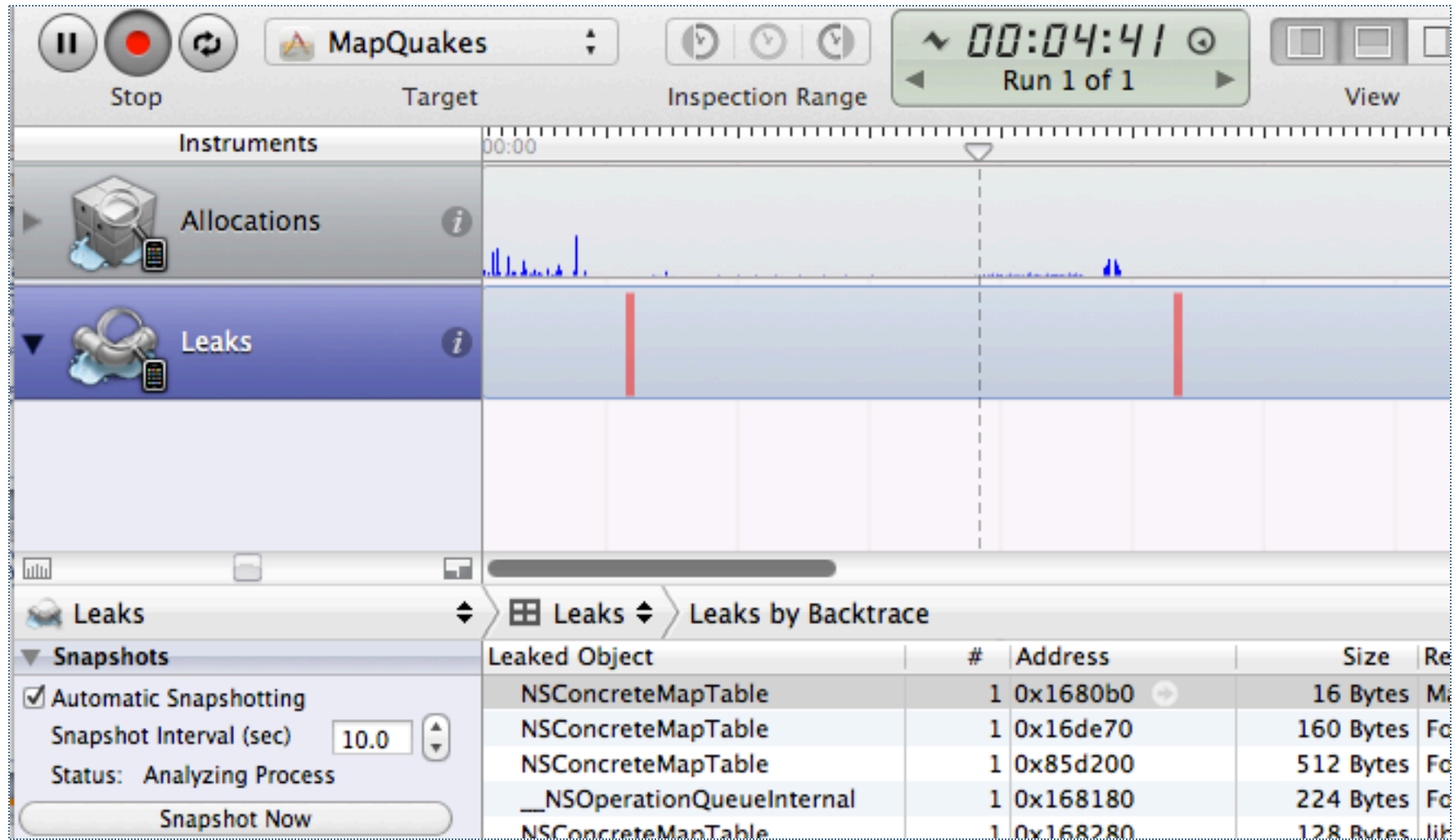
```
- (void) loadView { // or viewDidLoad  
    //...  
    data = [[NSMutableArray array] retain];  
}
```

```
- (void) loadView { // or viewDidLoad  
    //...  
    self.data = [NSMutableArray array];  
}
```

```
- (void) loadView { // or viewDidLoad  
    //...  
    data = [[NSMutableArray alloc] init];  
}
```

```
-(void) dealloc {  
    [data release];  
    [super dealloc];  
}
```


Rò rỉ bộ nhớ



Garbage Collection

- iOS không hỗ trợ garbage collection

