

Module 3: Threads in Go

Topic 1.2: Exiting Goroutines

Early Exit

```
func main() {  
    go fmt.Printf("New routine")  
    fmt.Printf("Main routine")  
}
```

- Only “**Main routine**” is printed
- Main finished before the new goroutine started

Delayed Exit

```
func main() {  
    go fmt.Printf("New routine")  
    time.Sleep(100 * time.Millisecond)  
    fmt.Printf("Main routine")  
}
```

- Add a delay in the main routine to give the new routine a chance to complete
- “**New RoutineMain Routine**” is now printed

Timing with Goroutines

- Adding a delay to wait for a goroutine is **bad!**
- Timing assumptions may be wrong
 - Assumption: delay of 100 ms will ensure that goroutine has time to execute
 - Maybe the OS schedules another thread
 - Maybe the Go Runtime schedules another goroutine
- Timing is nondeterministic
- Need formal **synchronization** constructs