**let logger = Logger(subsystem: "com.yourapp.identifier", category: "networking")**

let predicate = NSPredicate(format: "subsystem == 'com.yourapp.network' AND category == 'Network'")

os\_log("Filtered log message", log: networkLog, type: .debug)

func assertMainThread(file: StaticString = #file, line: UInt = #line) {

assert(Thread.isMainThread, "This code must be executed on the main thread", file: file, line: line)

}

// Usage:

assertMainThread()

class DebugOverlayView: UIView {

override func draw(\_ rect: CGRect) {

super.draw(rect)

guard let context = UIGraphicsGetCurrentContext() else { return }

context.saveGState()

context.setStrokeColor(UIColor.red.cgColor)

context.setLineWidth(2.0)

context.stroke(rect)

context.restoreGState()

}

}

// Usage

let debugView = DebugOverlayView(frame: view.bounds)

view.addSubview(debugView)

if lock1.lock(before: Date().addingTimeInterval(5)) {

if lock2.lock(before: Date().addingTimeInterval(5)) {

// Access shared resource

lock2.unlock()

}

lock1.unlock()

} else {

print("Failed to acquire locks, potential deadlock")

}

struct Person: CustomStringConvertible {

var name: String

var age: Int

var description: String {

return "\(name) is \(age) years old"

}

}

let person = Person(name: "Alice", age: 30)

print(person)

debugPrint(person)

let index = 5

precondition(index >= 0 && index < 10, "Index out of bounds")