Operating Systems – HW3

Part 1:

The design philosophy of Go: “Don’t communicate by sharing memory; share memory by communicating” means that instead of multiple threads or goroutines accessing the shared variables and coordinating with locks, Go will encourage you to pass the data using channels. The data is sent from one goroutine to another, then the receiving goroutine becomes responsible for that data. Communication will happen clearly through messages passing instead of through a shared state.

Advantages: prevent race conditions, easier reasoning concurrency, guarantees synchronization, and encourage a clean structure.

Disadvantages: Performance overhead can cause deadlocks to channels, cause debugging challenges, and may increase the involvement of the kernel.

https://blogtitle.github.io/go-advanced-concurrency-patterns-part-3-channels/
I believe that this website is a credible source because it references details from the official Go website, and it is detailed and technical. However, this is not official documentation since it is a blog.

https://medium.com/@relieved_gold_mole_613/golang-explanation-share-memory-by-communicating-8be944cbf8f8
This website may or may not be a credible source because not all medium sources are credible. However, I believe this source may be credible because It explains Go concepts kind of accurate and like the actual Go website and the author lists relevant points about the topic.

https://go.dev/blog/codelab-share

This website is a credible source because it is a blog from the Go language website. In the Go programming language website, it teaches users how to code in Go.

Part 2: https://github.com/Katw-1/OS-HW3

Instructions: To run the program, you compile the code. The output is printed on the debug console.

Design: This program compares two types of linked lists. The first list was implemented using Figure 29.8 in the textbook, Concurrent Linked List for safety for multiple threads running at the same time. The second type is Hand Over Hand locking, where each node has its own lock which allows threads to traverse different parts of the list at the same exact time.