Solutions Manual

AUTHOR NAME

November 5, 2016

CONTENTS

I Lecture Slides **5**

9 Monitors and Condition Variables **7**
9.1 What's wrong with *semaphors*? 7
9.2 Monitors 7

II Notes From Text **9**

Index **11**

Part I

Lecture Slides

CHAPTER 9

MONITORS AND CONDITION VARIABLES

WHAT'S WRONG WITH *semaphors*?

- are shared global variables
- no linguistic connection between semaphores and data they control
- can be accessed from anywhere
- dual purposed (mutex and sched constraints)
- no guarantee of proper usage

Solution: use a higher level construct

MONITORS

A monitor is similar to a class that ties data/operations and synchornization together.

They differ from classes by guaranteeing mutual exlusion and requiring all data to be private.

Definition. 1. (From Wikipedia) A *monitor* is a synchronization construct that allows threads to have both mutual exclusion and the ability to wait (block) for a certain condition to become true.

2. (From slides) A *monitor* is a defines a *lock* and zero or more *condition variables* for managing concurrent access to shared data.

Part II

Notes From Text

INDEX

monitor, 7

semaphors, 7