Analysis of Scheduling and Synchronization in XV6

GWU CSCI 3411 - Fall 2019 - Lab 6
Prepared by James Taylor (2017)

September 22, 2019

Topics

1 Introduction

In Lab 3, you were introduced to XV6. Since then, you have done a number of homework exercises in XV6, so you should be generally familiar with the XV6 framework. In this lab, we will continue to work with XV6 with a focus on how XV6 handles Scheduling and Synchronization.

This lab is organized around a number of questions on Scheduling and Synchronization. We will work in groups of 2 to 3 students and you are encouraged to discuss the questions amongst your group. You are given this time to examine and analyze the XV6 source code with the goal of answering the questions posed in this document. Your performance on future assignments will depend on your comprehension of the XV6 source and in particular your understanding of how XV6 addresses the topics raised in this lab.

Remember that you have linux tools available to examine the XV6 source code. In particular, you should use `grep` to expedite your review of the XV6 source.

2 Scheduling

1. Where is the dispatch function for context switching threads?
2. Where is the logic for the scheduler?
3. XV6 tracks its "runqueue" in a unique way. How is it implemented?
4. What scheduling policy does it use?

3 Synchronization

1. What do `pushcli` and `popcli` do? Any guesses? Can anyone explain parts of it?
2. What do `acquire` and `release` do?
3. Any guesses what `xchg` does? Its main use is in `acquire`, so that may give some hints.
4. How does the implementation of `acquire` and `release` use enabling and disabling interrupts? How does it use atomic instructions?

5. If it didn't use interrupt disabling how would the behavior of the system change for the worse?
6. If the implementation didn't use atomic instructions and the "locked" variable, how would the behavior change for the worse?
7. What is a `sleeplock`, and how does it differ from these spinlocks?
8. How does a `sleeplock` work, and in what situations do you think it is used in xv6?
9. How does the `sleeplock` synchronize access to the words and memory of its own implementation?