OS Project2 Report

521030910229 Hao Chiyu

April 13, 2023

Contents

1 Introduction

In the second project, I finished two different labs. Both of them are about inter-process communication and synchronization.

2 Stooge Farmer Problem

2.1 Implementation

In this lab, I declared three semaphores: shovel, empty_hole and seed_hole.
Larry: wait shovel, then digs a hole and signals empty_hole, then releases shovel.
Curly: wait Larry signaling empty_hole, puts seed in and signals seed_hole.
Moe: waits Curly signaling seed_hole then waits shovel, fill it then releases shovel.

2.2 Starvation

To avoid starvation, we add constraints that the number of unfilled holes can not be bigger than a constant. This will avoid Moe cannot get shovel for too long time. And because Moe must wait for seed_hole, so Larry will not starve.

3 Faneuil Hall Problem

3.1 Implementation

In this lab, I declared two semaphores: immigrant_sit_down, and immigrant_confirmed. Then I also declare a bool: judge_in_hall.

immigrant: wait no judge in hall, come in, check in and sit down, then signal immigrant_sit_down and wait immigrant_confirmed, get Certificate and swear. At last, they wait judge leaving and leave.

judge: wait immigrant_sit_down, then come in and sets judge_in_hall to true, confirm immigrant and leave. Don't forget to set judge_in_hall to false.

spectator: wait no judge in hall, come in and spectate. At last, they wait judge leaving and leave.

3.2 Starvation

To avoid starvation, we add constraints that judges only can come in when there has been a immigrant sat down, which can avoid judges always come in and leave and immigrants and spectators cannot come in.