Lab2 Report

陈昱衡 521021910939

2023 年 4 月 11 日

Stooge Farmers Problem

Solution

1. No doubt, we are supposed to use semaphore to realize correct synchronization between threads.I use four semaphores.Semaphore shovel make sure that only one of Larry and Curly will get to use the only one shovel.Semaphore dig stands for whether Larry can dig more holes.It will be ”wait” by Larry and ”post” by Curly.Semaphore ”fill” describes that there exist holes planted by Meo but not filled by Curly yet which will be ”wait” by Curly and ”post” by Meo.Semaphore seed, ”wait” by Meo and ”post” by Larry, says that there are un-planted hole which Moe can plant a seed.
2. And I declare three functions to run in three threads with infinite loop.
3. After I have finished the job, I run the executable file and find that the output was too neat.It is something like that N Larry's work followed by N Meo's work and followed by N Curly's work.It seems that it lacks in thread switching randomness.So I tried various methods and finally determined to use **usleep()** to invoke thread switching.
4. Instead of employ **rand()** to generate pseudo random number,I use inline assembly making use of x86-64 **rdrand** directive to generate real random number.
5. In order to avoid wasting system resources, I use **___attribute___((destructor)) void cleanup()** to release resources even if the program exits abnormally.

The Faneuil Hall problem

Solution

1. I use four semaphores in solving this task.
   - Semaphore ”enter” is applied to indicate whether Immigrants and Spectators can enter the hall.In thread **Immigrants** and **Spectators** , it will be ”wait” and ”post” immediately.But in thread **Judge**, it will be ”wait” at the begin of the function and ”post” at the end.
   - Semaphore ”to_confirm” is applied to inform the judge whether all the Immigrants entering in had already sit down so the judge can confirm.
   - Semaphore ”leave” is applied to stop Immigrants from leaving the hall while the judge is present at the hall.It will be ”wait” at the entry of the function **Judge** and ”post” at the end.Also , like semaphore ”enter”, in threads **Immigrants**,it will be ”wait” and ”post” immediately.
   - Semaphore ”confirmed” is ”post” in thread **Judge** in a N-time loop where N is equal to the number of the immigrants the judge has confirmed.And it will be ”wait” by each of the confirmed immigrants.
2. Like task 1, **usleep** and **rdrand** is used.

3. In order to simulate the real situation as much as possible, I create 5 **Immigrants** threads , 1 **Judge** thread and 2 **Spectators** threads.

4. In order to avoid wasting system resources, I use **___attribute___((destructor)) void cleanup()** to release resources even if the program exits abnormally.