

...

OS Term Project

OS Term Project

OS term project summary

- In this project, you will implement several scheduling algorithms to gain a better understanding of scheduler policies
- You will learn about "busy waiting", "priority scheduling", "preemption" and "priority inversion" and "thread scheduling" throughout this project
- You will start with basic Pintos kernel and, you figure out what you need to change to make it an advanced one.

OS Term Project

This project will help you study following skills and knowledge

- (1) The internals of the thread scheduling algorithm
- (2) How the kernel manages scheduling policy
- (3) How to modify the kernel to improve the performance of the scheduler
- (4) Important data structures for thread scheduling

What is Pintos?

Pintos

- Pintos is an educational operating system developed at Stanford University
- Pintos is designed to provide an experience to develop operating system without being excessively complex
- Pintos is developed with several limitations in terms of file system, thread scheduler, virtualization etc.
- Our purpose is to improve Pintos with advanced ideas



In this course, we focus on “thread scheduler”

Project schedule

Project #0: Get familiar with Pintos

- **Overview:** Install Pintos and analyze core functions in Pintos
- **Due date:** Monday, April 8, 2024, 11:59 PM

Project #1: Avoid busy-waiting algorithm

- **Overview:** Change the busy-waiting algorithm to an advanced one by adding "block status" in thread's lifecycle
- **Due date:** Monday, May 13, 2024, 11:59 PM

Project #2: Implement priority scheduling

- **Overview:** Develop priority scheduling based on the priority of each thread
- **Due date:** Monday, June 10, 2024, 11:59 PM

Project #0: Get familiar with Pintos

Part 1) Install Pintos on your PC/laptop

- We will provide an install guide for students
- Refer to "Part 0_Introduction (version 1.0).pdf" file

Part 2) Analyze core functions in Pintos line-by-line

- timer_sleep() / thread_yield() / schedule() / thread_block() / init_thread()
- These functions are responsible for thread scheduling in Pintos
- Additionally, an understanding of these functions is required to successfully carry out Projects #1 and #2

Project #1: Avoid busy-waiting algorithm

Limitation of a busy-waiting

- Busy-waiting is a technique where thread waits and constantly checks for a condition to be satisfied before proceeding with its execution
- In other words, a sleeping thread is occupying the CPU, inducing performance degradation
- This results in wasting CPU resource

Goal: Implement an efficient scheduling algorithm without using busy-waiting

- We can achieve this by adding "block" status in the thread's lifecycle
- We can manage sleeping threads separately in a list (i.e., block list) other than the ready list

Project #2: Implement priority scheduling

Limitation of a FIFO scheduling

- FIFO is simple and easy to understand but it has several disadvantages
- It doesn't release the CPU until it finishes executing as it is a Non-Preemptive CPU scheduling algorithm
- Thus, it has a high average waiting time
- It may cause starvation if the first job has the longest execution time

Goal: Implement an efficient priority scheduling

- We can achieve this by giving priority for each thread in Pintos
- We should implement preemption in the scheduler

Thank you

: -)