

PROJECT DESIGN DOCUMENT

Team Name : ADS

1. Avvaru Adithya 20162116
2. Dishant Chauhan 20162109
3. Sunny Ahlawat 20162068

Part 1: Understanding Pintos

Basically understood the source code and flow of Pintos and the procedure run pintos

Part 2: Adding file

We understood the procedure to add files to Pintos Kernel and run the program/function

We added hello.c and the corresponding hello.h file to Pintos and ran it successfully.

The command used to execute this program:

pintos -q run hello

Part 3: Implementing thread sleep mechanism

Problem: timer_sleep(), defined in timer.c, is a function which suspends the thread for say x ticks but spins into a loop checking the current time and calling thread_yield(). This actually leads to busy wating.

Approach: Instead of allowing the thread to call CPU using thread_yield() and checking time using timer_elapsed(), we calculate the wakeUp time of the thread using the current ticks and no of ticks to sleep and wake up the thread after this wakeUp time. This enables us to ensure that CPU is idle for this many ticks instead of being wasted and thus can perform tasks of another thread

Datastructures Used:

1. wait_time -- It is added to thread structure. It tell at what time a thread is to be woken up
2. struct list waiting_list -- It is used to contain a list of elems of blocked threads which calls timer_sleep(). It is declared in thread.h

Part 4: Implementation of priority Scheduling

Approach :

In priority Scheduling, we unblock the thread based on te priority of the current thread compared with the priority of the passed thread thread_unblock().

Based on the priority passed in thread_set_priority(), we are updating the priority of the current thread by comparing the priorities of the threads in ready_list using mySort function.

Selection of next thread to be assigned CPU is based on priority where we sorted the ready_list based on priority using mySort function