

# Priority Scheduling

---

## Intruduction

When a thread is added to the ready list that has a higher priority than the currently running thread, the current thread should immediately yield the processor to the new thread. Similarly, when threads are waiting for a lock, semaphore, or condition variable, the highest priority waiting thread should be awakened first. A thread may raise or lower its own priority at any time, but lowering its priority such that it no longer has the highest priority must cause it to immediately yield the CPU.

Thread priorities range from *PRIMIN* (0) to *PRMAX* (63). Lower numbers correspond to lower priorities, so that priority 0 is the lowest priority and priority 63 is the highest. The initial thread priority is passed as an argument to `threadcreate()`. *If there's no reason to choose another priority, use `PRDEFAULT` (31).* The `PRI_` macros are defined in 'threads/thread.h', and you should not change their values.

## Assignment

### Description

In this assignment, you are going to implement priority scheduling in Pintos. Implement the following functions that allow a thread to examine and modify its own priority. Skeletons for these functions are provided in 'threads/thread.c'.

**`void threadsetpriority (int new_priority)`** [Function]

Sets the current thread's priority to new priority. If the current thread no longer has the highest priority, yields.

You need not provide any interface to allow a thread to directly modify other threads' priorities.

## Requirement

You should pass all following tests:

1. alarm-single
2. alarm-multiple
3. alarm-simultaneous
4. alarm-priority
5. alarm-zero
6. alarm-negative
7. priority-change

8. priority-fifo
9. priority-preempt

Run `make check` in ***src/threads/*** to make sure your design can pass all these tests.

You should download and fill the [DESIGNDOC](#) and archive your code(**the whole pintos folder**) and DESIGNDOC into a **zip** file named ***sidnameInPinYin*.id.zip**(e.g ***12330441zhangsanv0.zip***) and upload to the ftp.