https://pixabay.com/vectors/magnifying-glass-magnifier-glass-189254/
https://zh.wikipedia.org/wiki/Linux

http://wap.yesky.com/gameonline/405/11790405.shtml

作業九：用C11實現spinlock並量測公平性

中正大學 作業系統實驗室
指導大叔： 羅習五

作業目標及負責助教

- 作業目標：
  - 查看pthread的spinlock實作方式
  - 使用C11將重新實現spinlock
  - 使用自行設計的spinlock，試試看在不保障bounded waiting的情況下，公平性怎樣?

查閱pthread的spinlock實作方式

- 可以使用google的關鍵字搜尋「gnu pthread spinlock source code」
- 利用上述關鍵字可以查到下列網頁
  - https://elixir.bootlin.com/glibc/latest/source/nptl/pthread_spin_lock.c

GNU的程式碼

```
int __pthread_spin_lock (pthread_spinlock_t *lock) {
 int val = 0;
 if (__glibc_likely (atomic_exchange_acquire (lock, 1) == 0))
  return 0;
 do {
   do {
   /* TODO Back-off.  */
   atomic_spin_nop ();
   val = atomic_load_relaxed (lock);
  } while (val != 0);
 } while (!atomic_compare_exchange_weak_acquire (lock, &val, 1));
 return 0;
}
int __pthread_spin_unlock (pthread_spinlock_t *lock) {
 atomic_store_release (lock, 0);
 return 0;
}
```

使用C 11實現的程式碼範例

```
static inline int my_spin_lock (atomic_int *lock) {
    int val=0;
    if (likely(atomic_exchange_explicit(lock, 1, memory_order_acq_rel) == 0))
        return 0;
    do {
        do {
            asm("pause");
        } while (*lock != 0);
        val = 0;
    } while (!atomic_compare_exchange_weak_explicit(lock, &val, 1, memory_order_acq_rel, memory_order_relaxed));
    return 0;
}
static inline int my_spin_unlock(atomic_int *lock) {
    atomic_store_explicit(lock, 0, memory_order_release);
    return 0;
}
```

量測每個core進入CS的次數是否公平

```
void thread(void *givenName) {
  int givenID = (intptr_t)givenName;
  srand((unsigned)time(NULL));
  unsigned int rand_seq;
  cpu_set_t set; CPU_ZERO(&set); CPU_SET(givenID, &set);
  sched_setaffinity(gettid(), sizeof(set), &set);
  while(atomic_load_explicit(&wait, memory_order_acquire));
  while(1) {
    my_spin_lock(&a_lock);
    atomic_fetch_add(&in_cs, 1);
    atomic_fetch_add_explicit(&count_array[givenID], 1, memory_order_relaxed);
    if (in_cs != 1) {
      printf("violation: mutual exclusion\n"); exit(0);
    }
    atomic_fetch_add(&in_cs, -1);
    my_spin_unlock(&a_lock);
    int delay_size = rand_r(&rand_seq)%73;
    for (int i=0; i<delay_size; i++);
  }
}
```

作業繳交

1. （15pt）請對GNU的spinlock撰寫簡短的註解（大致上每一行都要）
2. （15pt）請對C11的spinlock撰寫簡短的註解（大致上每一行都要）
3. （20pt）對如何量測各個核心進入CS的次數是否公平的程式碼撰寫註解（大致上每一行都要）
4. （50pt）說明你的硬體的設定，列出實驗數據，並解釋實驗數據
5. 上述1~3繳交程式碼，4繳交pdf