Experiment 6

Experiment 6: Thread Synchronization

Aim:

Master the mutexs mechanism

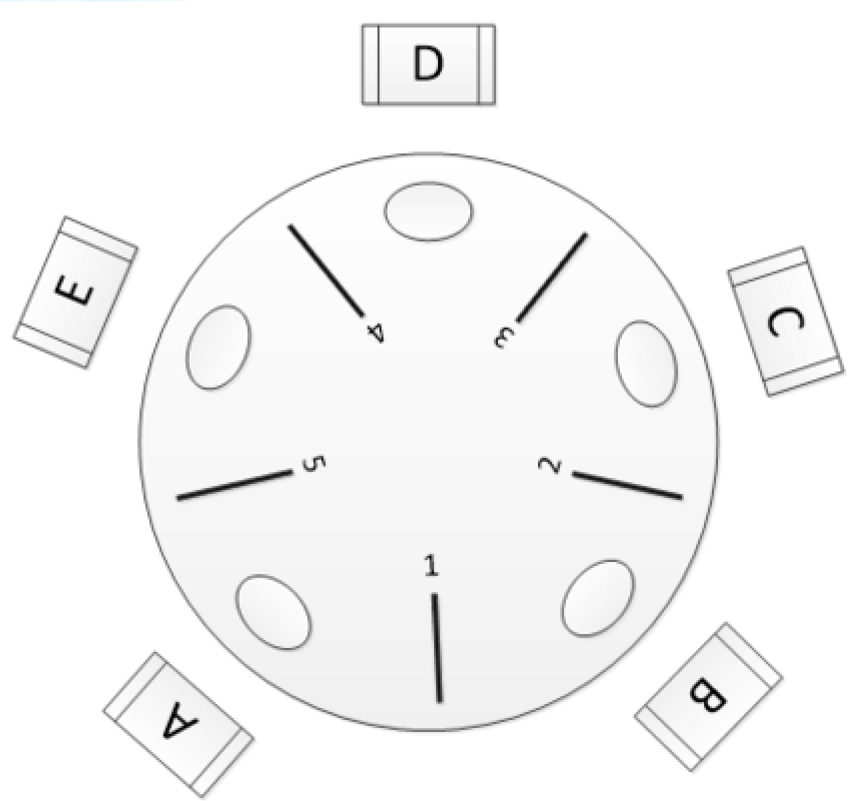
Master the condition mechanism

How to design the program to implement the multi-thread cooperative work.

Content:

Implementing the philosopher eating problem by adopting the mutexs and condition mechanism

Experiment 6: Thread Synchronization



function

```
int pthread_cond_wait( pthread_cond_t *cond,
                       pthread_mutex_t *mutex);
```

Cond: the condition argument.

block-->unlock-->wait()return-->lock

条件变量是利用线程间共享的全局变量进行同步的一种机制，主要包括两个动作：一个线程等待"条件变量的条件成立"而挂起；另一个线程使"条件成立"（给出条件成立信号）。为了防止竞争，条件变量的使用总是和一个互斥锁结合在一起。

```
int pthread_mutex_lock(pthread_mutex_t *mutex);
```

Lock the mutex lock;

```
int pthread_mutex_unlock(pthread_mutex_t *mutex);
```

Unlock the mutex lock;

```
int pthread_cond_broadcast(pthread_cond_t *cond)
```

Wake up the thread blocked by the condition argument;

```
pthread_attr_setdetachstate(&attr, PTHREAD_CREATE_DETACHED);
```

Detach the thread and the system will release the resource