要取得 [a,b] 的随机整数，使用 (rand() % (b-a+1))+ a;

lab06P Dining Philosophers Problem

软件工程 2018 级 1813075 刘茵

Target

1. Write a c/c++ program
2. To implement the dining philosophers problem
3. Gcc

---

1) Install GCC Software Colletion

```
>> sudo apt-get install build-essential
```

2) How to use GCC

● gcc and make

3) posix thread

```
#include <pthread.h>
pthread_create()
```

4) write a c program to implement the dining philosophers problem:

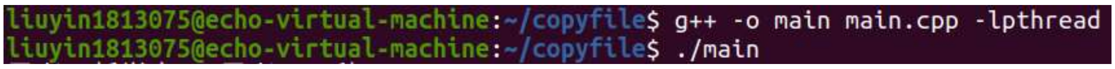
- 5 philosophers
- 思考随机时间 3~8s
- 就餐随机时间 2~10s
- 就餐 10 次后结束

>>>>>>>
仅当哲学家的左右手筷子都拿起时才允许进餐，利用信号量的保护机制实现。
原理：通过互斥信号量 mutex 对 eat() 之前取左侧和右侧筷子的操作进行保护，可以防止死锁的出现。

● main.cpp:

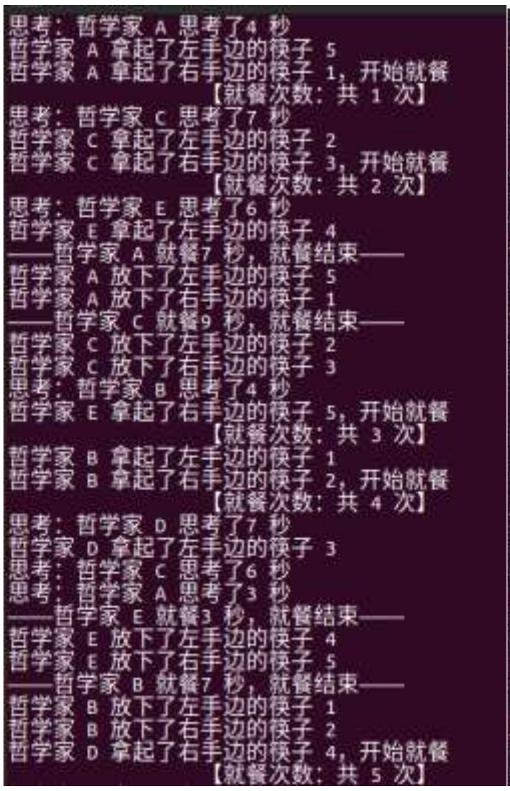
```
//       要取得 [a,b] 的随机整数，使用 (rand() % (b-a+1))+ a;

#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>
#include <stdio.h>
#include <semaphore.h> // 信号量 sem_init、sem_wait、sem_post、sem_destroy
#include <sys/time.h>
#include <time.h>
#include <memory.h>
#include <errno.h>
#include <math.h>
//筷子作为 mutex
pthread_mutex_t chopstick[6] ;
sem_t NUM;
sem_t Left,Right;
int num=0;
void *eat_think(void *arg)
{
        char phi = *(char *)arg;
        int left,right; //左右筷子的编号
        switch (phi){
                case 'A':
                        left = 5;
                        right = 1;
                        break;
                case 'B':
                        left = 1;
                        right = 2;
                        break;
                case 'C':
                        left = 2;
                        right = 3;
                        break;
                case 'D':
                        left = 3;
                        right = 4;
                        break;
                case 'E':
                        left = 4;
                        right = 5;
                        break;
        }
        int i;
        for(;;){
                if(num>=10)
                        exit(0);
                int time1=(rand() % (8-3+1))+ 3;
                usleep(time1); //思考
                printf("思考：哲学家 %c 思考了%d 秒\n",phi,time1);
                sem_wait(&Left);
                pthread_mutex_lock(&chopstick[left]); //拿起左手的筷子
                printf("哲学家 %c 拿起了左手边的筷子 %d\n", phi, left);
                pthread_mutex_lock(&chopstick[right]);
                printf("哲学家 %c 拿起了右手边的筷子 %d，开始就餐\n", phi, right);
                num++;
                printf("                              【就餐次数：共 %d 次】\n",num);
                sem_post(&Left);
                int time2=(rand() % (10-2+1))+ 2;
                usleep(time2); //吃饭
                printf("————————哲学家 %c 就餐%d 秒，就餐结束————————\n",phi,time2);
                pthread_mutex_unlock(&chopstick[left]); //放下左手的筷子
                printf("哲学家 %c 放下了左手边的筷子 %d\n", phi, left);
                pthread_mutex_unlock(&chopstick[right]); //放下右手的筷子
                printf("哲学家 %c 放下了右手边的筷子 %d\n", phi, right);
        }
        return 0;
}
int main(){
        pthread_t A,B,C,D,E; //5 个哲学家
        char a='A',b='B',c='C',d='D',e='E';
        sem_init(&NUM,0,1);
        sem_init(&Left,0,1);
        sem_init(&Right,0,1);
        int i;
        for (i = 0; i < 5; i++)
                pthread_mutex_init(&chopstick[i],NULL);
        pthread_create(&A,NULL, eat_think, &a);
        pthread_create(&B,NULL, eat_think, &b);
        pthread_create(&C,NULL, eat_think, &c);
        pthread_create(&D,NULL, eat_think, &d);
        pthread_create(&E,NULL, eat_think, &e);

        pthread_join(A,NULL);
        pthread_join(B,NULL);
        pthread_join(C,NULL);
        pthread_join(D,NULL);
```

```
    pthread_join(E,NULL);
    return 0;
}
```

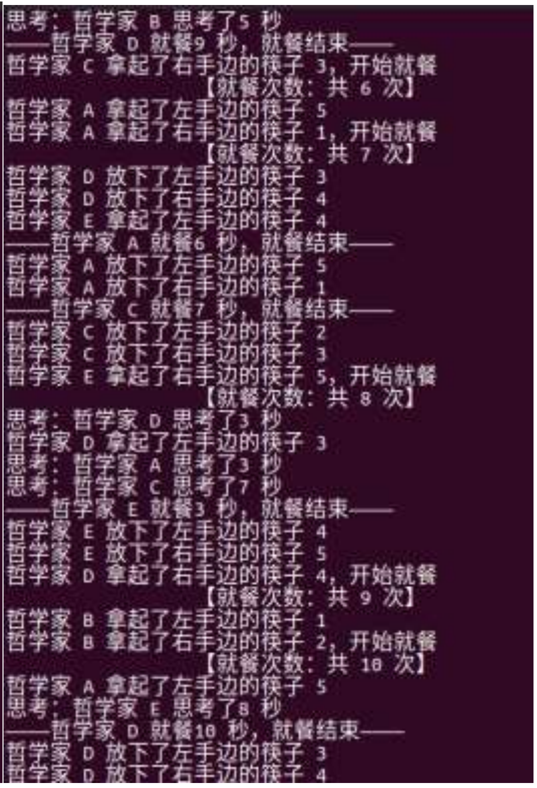


```
liuyin1813075@echo-virtual-machine:~/copyfile$ g++ -o main main.cpp -lpthread
liuyin1813075@echo-virtual-machine:~/copyfile$ ./main
```

结果 截图:



```
思考: 哲学家 A 思考了4 秒
哲学家 A 拿起了左手边的筷子 5
哲学家 A 拿起了右手边的筷子 1, 开始就餐
                    【就餐次数: 共 1 次】
思考: 哲学家 C 思考了7 秒
哲学家 C 拿起了左手边的筷子 2
哲学家 C 拿起了右手边的筷子 3, 开始就餐
                    【就餐次数: 共 2 次】
思考: 哲学家 E 思考了6 秒
哲学家 E 拿起了左手边的筷子 4
——哲学家 A 就餐7 秒, 就餐结束——
哲学家 A 放下了左手边的筷子 5
哲学家 A 放下了右手边的筷子 1
——哲学家 C 就餐9 秒, 就餐结束——
哲学家 C 放下了左手边的筷子 2
哲学家 C 放下了右手边的筷子 3
思考: 哲学家 B 思考了4 秒
哲学家 E 拿起了右手边的筷子 5, 开始就餐
                    【就餐次数: 共 3 次】
哲学家 B 拿起了左手边的筷子 1
哲学家 B 拿起了右手边的筷子 2, 开始就餐
                    【就餐次数: 共 4 次】
思考: 哲学家 D 思考了7 秒
哲学家 D 拿起了左手边的筷子 3
思考: 哲学家 C 思考了6 秒
思考: 哲学家 A 思考了3 秒
——哲学家 E 就餐3 秒, 就餐结束——
哲学家 E 放下了左手边的筷子 4
哲学家 E 放下了右手边的筷子 5
——哲学家 B 就餐7 秒, 就餐结束——
哲学家 B 放下了左手边的筷子 1
哲学家 B 放下了右手边的筷子 2
哲学家 D 拿起了右手边的筷子 4, 开始就餐
                    【就餐次数: 共 5 次】
```



```
思考: 哲学家 B 思考了5 秒
——哲学家 D 就餐9 秒, 就餐结束——
哲学家 C 拿起了右手边的筷子 3, 开始就餐
                    【就餐次数: 共 6 次】
哲学家 A 拿起了左手边的筷子 5
哲学家 A 拿起了右手边的筷子 1, 开始就餐
                    【就餐次数: 共 7 次】
哲学家 D 放下了左手边的筷子 3
哲学家 D 放下了右手边的筷子 4
哲学家 E 拿起了左手边的筷子 4
——哲学家 A 就餐6 秒, 就餐结束——
哲学家 A 放下了左手边的筷子 5
哲学家 A 放下了右手边的筷子 1
——哲学家 C 就餐7 秒, 就餐结束——
哲学家 C 放下了左手边的筷子 2
哲学家 C 放下了右手边的筷子 3
哲学家 E 拿起了右手边的筷子 5, 开始就餐
                    【就餐次数: 共 8 次】
思考: 哲学家 D 思考了3 秒
哲学家 D 拿起了左手边的筷子 3
思考: 哲学家 A 思考了3 秒
思考: 哲学家 C 思考了7 秒
——哲学家 E 就餐3 秒, 就餐结束——
哲学家 E 放下了左手边的筷子 4
哲学家 E 放下了右手边的筷子 5
哲学家 D 拿起了右手边的筷子 4, 开始就餐
                    【就餐次数: 共 9 次】
哲学家 B 拿起了左手边的筷子 1
哲学家 B 拿起了右手边的筷子 2, 开始就餐
                    【就餐次数: 共 10 次】
哲学家 A 拿起了左手边的筷子 5
思考: 哲学家 E 思考了8 秒
——哲学家 D 就餐10 秒, 就餐结束——
哲学家 D 放下了左手边的筷子 3
哲学家 D 放下了右手边的筷子 4
```