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
4. Install GCC Software Colletion

>> sudo apt-get install build-essential

1. How to use GCC

* [gcc and make](https://www3.ntu.edu.sg/home/ehchua/programming/cpp/gcc_make.html)

1. posix thread

#include <pthread.h>

pthread\_create()

1. 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;

}



结果 截图：

