#include <pthread.h>

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <unistd.h>

struct arr{

int n;

int a[10];

};

struct file {

struct arr ar;

char\* filename;

};

pthread\_mutex\_t a\_mutex; //dong bo

static int sum =0;

void\* thr1(void\* ar){

struct arr \*ap = (struct arr\*) ar;

ap->n=10; int i=0;

for(i=0;i<ap->n;i++)

ap->a[i] = i+1;

}

void\* thr2(void\* ar){

pthread\_mutex\_lock (&a\_mutex);

struct arr \*ap = (struct arr\*) ar;

int i, s=0;

for(i=0;i<ap->n;i++)

s=s + ap->a[i];

sum=s;

printf("%d \n",sum);

pthread\_mutex\_unlock (&a\_mutex);

}

void\* thr3 (void\* ar){

pthread\_mutex\_lock (&a\_mutex);

struct file \*fi = (struct file\*) ar;

FILE \*out; int count;

out= fopen(fi->filename,"wb");

fprintf(out,"number element or array: %d\n", fi->ar.n);

for(count=0; count<fi->ar.n; count++){

fprintf(out,"%d\t",fi->ar.a[count]);

}

fprintf(out,"\n");

fprintf(out,"sum=%d\n",sum);

fclose(out);

pthread\_mutex\_unlock (&a\_mutex);

}

int main (int argc,char \* argv[]){

int res=pthread\_mutex\_init (&a\_mutex, NULL);

//a\_mutex = PTHREAD\_MUTEX\_INITIALIZER;

int i;

pthread\_t tid[3];

struct arr ar;

int status, \*pstatus= &status;

pthread\_create(&tid[0],NULL,thr1,(void\*) &ar);

if(pthread\_join(tid[0],(void\*\*) pstatus)==0)

{

pthread\_create(&tid[1],NULL,thr2,(void\*) &ar);

struct file arf;

arf.ar=ar;

arf.filename=argv[1];

pthread\_create(&tid[2],NULL,thr3,(void\*) &arf);

}

sleep(2);

return 0;

}