**西安电子科技大学 网络与信息安全学院**

**操 作 系 统 实验报告**

**班 级： 1618011**

**学 号： 16180110065**

**姓 名： 王宇卿**

**Github： https://github.com/pitifulnoble**

**电子邮箱： pitifulnoble@outlook.com**

**2018年4月20日**

|  |
| --- |
| **实验题目：Multithreaded Sorting Application** |
| **一、实验题目**  写一个c程序实现以下特性：  该程序有多个线程，将一个序列分成两组，然后对每组新序列执行任意排序算法，这个过程通过  创建新的线程实现。 |
| **二、相关原理与知识**  **了解C程序库pthread.h**  **对Linux基本知识的了解**  **熟悉线程相关知识** |
| **三、实验过程**  **对下列序列实现从小到大的排序：**  **捕获**  **通过创建线程实现：**  **捕获**  **结果展示：**  **捕获** |
| **四、实验结果与分析**  完成了实验要求，通过创建新的线程，对序列做了一系列操作。 |
| **五、问题总结**  **在编译源代码时遇到问题。出现如下错误：**  **捕获**  **通过查阅相关资料，发现带有ptread.h头文件的C程序源代码需要在**  **编译时加上-pthread参数。** |
| **源代码**  **#include<stdio.h>**  **#include<stdlib.h>**  **#include<pthread.h>**  **#include<string.h>**  **struct thread{**  **int length;**  **int\* list;**  **};**  **void \*sort(void \* data)**  **{**  **Data D=\*(Data\*)data;**  **int len=D.length;**  **int\* a=D.list;**  **int i;**  **int temp= 0;**  **for(i=0;i<len;i++)**  **{**  **for(int j=i+1;j<len;j++)**  **{**  **if(a[i] > a[j])**  **{**  **temp = a[i];**  **a[i] = a[j];**  **a[j] = temp;**  **}**  **}**  **}**  **pthread\_exit(NULL);**  **}**  **void sort1(int\* a,int len){**  **int temp= 0,i;**  **for(i=0;i<len;i++)**  **{**  **for(int j=i+1;j<len;j++)**  **{**  **if(a[i] > a[j])**  **{**  **temp = a[i];**  **a[i] = a[j];**  **a[j] = temp;**  **}**  **}**  **printf("line[%d]:%d\n",i,a[i]);**  **}**  **}**  **void \*merge(void \* data){**  **Data \*D=(Data\*)data;**  **int i,j;**  **memcpy(fin,D[0].list,D[0].length\*sizeof(int));**  **memcpy(fin+D[0].length,D[1].list,D[1].length\*sizeof(int));**  **sort1(fin,D[0].length+D[1].length);**  **pthread\_exit(NULL);**  **}**  **int main(){**  **int data[10]={5,9,66,233,53,9,23,38,47,203};**  **fin=(int\*)malloc(sizeof(int)\*(sizeof(data)/sizeof(int)));**  **pthread\_t threads[3];**  **int rc[3];**  **int i,j;**  **int len=sizeof(data)/sizeof(int);**  **Data seq[2];**  **for(i=0;i<2;i++){**  **if(i==0) seq[i].length=len/2;**  **else seq[i].length=len-len/2;**  **seq[i].list=(int\*)malloc(sizeof(int)\*seq[i].length);**  **for(j=0;j<seq[i].length;j++){**  **if(i==0) seq[i].list[j]=data[j];**  **else seq[i].list[j]=data[j+len/2];**  **printf("sed%d[%d] former:%d\n",i+1,j,seq[i].list[j]);**  **}**  **rc[i]=pthread\_create(&(threads[i]),NULL,sort,(void\*)&(seq[i]));**  **}**  **pthread\_join(threads[0],NULL);**  **pthread\_join(threads[1],NULL);**  **rc[2]=pthread\_create(&threads[2],NULL,merge,(void\*)seq);**  **pthread\_join(threads[2],NULL);**  **}** |