Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №3 по курсу**

**«Операционные системы»**

Студент: Литовченко Анна Александровна

Группа: М8О-207Б-21

Вариант: 3

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

**Репозиторий**

https://github.com/Annalitov/OS/lab3

**Постановка задачи**

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработки использовать стандартные средства создания потоков операционной системы (Windows/Unix). Ограничение потоков должно быть задано ключом запуска вашей программы. Так же необходимо уметь продемонстрировать количество потоков, используемое вашей программой с помощью стандартных средств операционной системы.

**Вариант 3**:Отсортировать массив целых чисел при помощи параллельной сортировки слиянием

**Общие сведения о программе**

Программа представлена одним файлом: main.cpp

Операционная система: MacOs

**Общий метод и алгоритм решения**

pthread\_t threads[THREAD\_MAX] - создаем массив идентификаторов потока

pthread\_create(&threads[i], NULL, merge\_sort\_tread, NULL) - создаем потоки

Используемые системные вызовы

pthread\_join - дожидается завершения переданного потока, после чего получает его выходное значение и позволяет программе продолжить работу

pthread\_create - создает новый поток выполнения в программе

**Исходный код**

**main.cpp**

#include <iostream>

#include <cstdlib>

#include <pthread.h>

#include <chrono>

#include <ctime>

using namespace std;

int array\_size;

int n\_threads;

int part=0;

int\* a ;

void merge(int low, int mid, int high)

{

int n1 = mid - low + 1, nr = high - mid, i, j;

int\* left = (int\*)calloc(n1,sizeof(int));

int\* right = (int\*)calloc(nr,sizeof(int));

for(i = 0; i < n1; i++)

left[i] = a[i + low];

for(i = 0; i < nr; i++)

right[i] = a[i + mid + 1];

int k = low;

i = j = 0;

while(i < n1 && j < nr)

{

if(left[i] <= right[j])

a[k++] = left[i++];

else

a[k++] = right[j++];

}

while(i < n1) {

a[k++] = left[i++];

}

while(j < nr) {

a[k++] = right[j++];

}

}

void merge\_sort(int low, int high)

{

int mid = low + ((high - low) / 2);

if(low < high) {

merge\_sort(low, mid);

merge\_sort(mid + 1, high);

merge(low, mid, high);

}

}

void\* merge\_sort\_tread(void \*args)

{

int thread\_part = part;

part+=1;

int low = thread\_part \* (array\_size / n\_threads);

int high = (thread\_part + 1) \* (array\_size / n\_threads) - 1;

int mid = low + (high - low) / 2;

if(low < high)

{

merge\_sort(low, mid);

merge\_sort(mid + 1, high);

merge(low, mid, high);

}

return 0;

}

void merge\_rec(int tread\_m)

{

if(tread\_m>n\_threads)

return;

merge\_rec(tread\_m\*2);

int minsize = array\_size/tread\_m;

for(int i=0;i<tread\_m;i++)

{

int low = i \* minsize;

int high = (i + 1) \* (minsize) - 1;

int mid = low + (high - low) / 2;

merge(low, mid, high);

}

}

int main()

{

cout<<"Enter the number of threads: ";

cin>>n\_threads;

cout<<endl;

cout<<"Enter the array size: ";

cin>>array\_size;

a = (int\*)calloc(array\_size,sizeof(int));

cout<<"The input array: ";

int i;

srand(time(NULL));

for( i=0; i < array\_size; i++)

{

a[i] = rand() % array\_size;

cout<<a[i]<<" ";

}

cout<<"\n";

pthread\_t threads[n\_threads];

int status;

chrono::steady\_clock::time\_point begin = chrono::steady\_clock::now();

for(int i = 0; i < n\_threads; i++)

{

cout<<"part- "<< part << endl;

status = pthread\_create(&threads[i], NULL, merge\_sort\_tread, NULL);

if (status != 0)

cout<<"main error: can't create thread, status = "<< status <<endl;

}

for(int i = 0; i < n\_threads; i++)

pthread\_join(threads[i], NULL);

merge\_rec(2);

merge(0, (array\_size - 1)/2, array\_size - 1);

chrono::steady\_clock::time\_point end = chrono::steady\_clock::now();

cout<<"The output array: "<<endl;

for(int i = 0; i < array\_size; i++)

cout<<a[i]<<endl;

cout<<"The number of threads:"<<n\_threads<<endl;

cout<<chrono::duration\_cast<chrono::microseconds>(end-begin).count()<<endl;

return 0;

}

**Демонстрация работы программы**

(base) litann@Annalit lab3 % ./main

Enter the number of threads: 2

Enter the array size: 100

The input array: 62 12 61 3 78 10 46 83 86 39 14 41 47 0 33 96 45 9 38 13 67 6 43 88 64 8 17 35 19 28 15 70 44 53 72 83 33 78 21 16 4 68 62 64 6 42 32 72 81 87 10 80 74 59 3 57 68 48 48 9 17 76 5 90 27 68 15 39 83 86 5 13 30 92 25 97 10 4 18 11 67 28 22 89 59 2 37 61 66 99 17 5 87 5 78 4 1 14 6 49

part- 0

part- 0

The output array:

0

1

2

3

3

4

4

4

5

5

5

5

6

6

6

8

9

9

10

10

10

11

12

13

13

14

14

15

15

16

17

17

17

18

19

21

22

25

27

28

28

30

32

33

33

35

37

38

39

39

41

42

43

44

45

46

47

48

48

49

53

57

59

59

61

61

62

62

64

64

66

67

67

68

68

68

70

72

72

74

76

78

78

78

80

81

83

83

83

86

86

87

87

88

89

90

92

96

97

99

The number of threads:2

182

(base) litann@Annalit lab3 % ./main

Enter the number of threads: 4

Enter the array size: 100

The input array: 25 6 89 37 13 64 95 3 64 44 33 91 46 31 69 88 6 70 57 21 6 45 42 25 28 70 61 55 22 61 12 75 58 56 80 39 9 90 9 64 13 93 17 59 36 0 2 15 86 12 90 22 52 61 74 20 96 36 42 89 39 35 11 12 60 57 14 64 59 36 99 1 51 33 22 88 1 81 14 3 73 27 83 69 36 75 0 68 2 5 15 71 91 8 64 69 57 71 7 57

part- 0

part- 0

part- 1

part- 2

The output array:

0

0

1

1

2

2

3

3

5

6

6

6

7

8

9

9

11

12

12

12

13

13

14

14

15

15

17

20

21

22

22

22

25

25

27

28

31

33

33

35

36

36

36

36

37

39

39

42

42

44

45

46

51

52

55

56

57

57

57

57

58

59

59

60

61

61

61

64

64

64

64

64

68

69

69

69

70

70

71

71

73

74

75

75

80

81

83

86

88

88

89

89

90

90

91

91

93

95

96

99

The number of threads:4

198

(base) litann@Annalit lab3 %

(base) litann@Annalit lab3 % sudo dtruss -f ./main

PID/THRD SYSCALL(args) = return

Enter the number of threads: 1104/0x28f1: fork() = 0 0

1104/0x28f1: munmap(0x1049B4000, 0x8C000) = 0 0

1104/0x28f1: munmap(0x104A40000, 0x8000) = 0 0

1104/0x28f1: munmap(0x104A48000, 0x4000) = 0 0

1104/0x28f1: munmap(0x104A4C000, 0x4000) = 0 0

1104/0x28f1: munmap(0x104A50000, 0x54000) = 0 0

1104/0x28f1: open(".\0", 0x100000, 0x0) = 3 0

1104/0x28f1: fcntl(0x3, 0x32, 0x16B73F338) = 0 0

1104/0x28f1: close(0x3) = 0 0

1104/0x28f1: fsgetpath(0x16B73F348, 0x400, 0x16B73F328) = 32 0

1104/0x28f1: fsgetpath(0x16B73F358, 0x400, 0x16B73F338) = 14 0

1104/0x28f1: csrctl(0x0, 0x16B73F75C, 0x4) = -1 Err#1

1104/0x28f1: \_\_mac\_syscall(0x18C180143, 0x2, 0x16B73F6B0) = 0 0

1104/0x28f1: csrctl(0x0, 0x16B73F77C, 0x4) = -1 Err#1

1104/0x28f1: \_\_mac\_syscall(0x18C17D094, 0x5A, 0x16B73F710) = 0 0

1104/0x28f1: sysctl([unknown, 3, 0, 0, 0, 0] (2), 0x16B73EC80, 0x16B73EC70, 0x18C17ECA1, 0xD) = 0 0

1104/0x28f1: sysctl([CTL\_KERN, 136, 0, 0, 0, 0] (2), 0x16B73ED28, 0x16B73ED20, 0x0, 0x0) = 0 0

1104/0x28f1: open("/\0", 0x20100000, 0x0) = 3 0

1104/0x28f1: openat(0x3, "System/Cryptexes/OS\0", 0x100000, 0x0) = 4 0

1104/0x28f1: dup(0x4, 0x0, 0x0) = 5 0

1104/0x28f1: fstatat64(0x4, 0x16B73E801, 0x16B73E770) = 0 0

1104/0x28f1: openat(0x4, "System/Library/dyld/\0", 0x100000, 0x0) = 6 0

1104/0x28f1: fcntl(0x6, 0x32, 0x16B73E800) = 0 0

1104/0x28f1: dup(0x6, 0x0, 0x0) = 7 0

1104/0x28f1: dup(0x5, 0x0, 0x0) = 8 0

1104/0x28f1: close(0x3) = 0 0

1104/0x28f1: close(0x5) = 0 0

1104/0x28f1: close(0x4) = 0 0

1104/0x28f1: close(0x6) = 0 0

1104/0x28f1: shared\_region\_check\_np(0x16B73EE30, 0x0, 0x0) = 0 0

1104/0x28f1: fsgetpath(0x16B73F368, 0x400, 0x16B73F2B8) = 82 0

1104/0x28f1: fcntl(0x8, 0x32, 0x16B73F368) = 0 0

1104/0x28f1: close(0x8) = 0 0

1104/0x28f1: close(0x7) = 0 0

1104/0x28f1: getfsstat64(0x0, 0x0, 0x2) = 10 0

1104/0x28f1: getfsstat64(0x1046D2090, 0x54B0, 0x2) = 10 0

1104/0x28f1: getattrlist("/\0", 0x16B73F6A8, 0x16B73F668) = 0 0

1104/0x28f1: fsgetpath(0x16B73F2E8, 0x400, 0x16B73F2C8) = 82 0

1104/0x28f1: stat64("/System/Volumes/Preboot/Cryptexes/OS/System/Library/dyld/dyld\_shared\_cache\_arm64e\0", 0x16B73F750, 0x0) = 0 0

1104/0x28f1: stat64("/Users/litann/Desktop/lab3/main\0", 0x16B73EA90, 0x0) = 0 0

1104/0x28f1: open("/Users/litann/Desktop/lab3/main\0", 0x0, 0x0) = 3 0

1104/0x28f1: mmap(0x0, 0xAA51, 0x1, 0x40002, 0x3, 0x0) = 0x104750000 0

1104/0x28f1: fcntl(0x3, 0x32, 0x16B73EBA8) = 0 0

1104/0x28f1: close(0x3) = 0 0

1104/0x28f1: munmap(0x104750000, 0xAA51) = 0 0

1104/0x28f1: stat64("/Users/litann/Desktop/lab3/main\0", 0x16B73F000, 0x0) = 0 0

1104/0x28f1: stat64("/usr/lib/libc++.1.dylib\0", 0x16B73DFD0, 0x0) = -1 Err#2

1104/0x28f1: stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/libc++.1.dylib\0", 0x16B73DF80, 0x0) = -1 Err#2

1104/0x28f1: stat64("/usr/lib/system/libdispatch.dylib\0", 0x16B73BBB0, 0x0) = -1 Err#2

1104/0x28f1: stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/system/libdispatch.dylib\0", 0x16B73BB60, 0x0) = -1 Err#2

1104/0x28f1: stat64("/usr/lib/system/libdispatch.dylib\0", 0x16B73BBB0, 0x0) = -1 Err#2

1104/0x28f1: stat64("/usr/lib/libSystem.B.dylib\0", 0x16B73DFD0, 0x0) = -1 Err#2

1104/0x28f1: stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/libSystem.B.dylib\0", 0x16B73DF80, 0x0) = -1 Err#2

1104/0x28f1: open("/dev/dtracehelper\0", 0x2, 0x0) = 3 0

1104/0x28f1: ioctl(0x3, 0x80086804, 0x16B73DCC8) = 0 0

1104/0x28f1: close(0x3) = 0 0

1104/0x28f1: open("/Users/litann/Desktop/lab3/main\0", 0x0, 0x0) = 3 0

1104/0x28f1: \_\_mac\_syscall(0x18C180143, 0x2, 0x16B73D2C0) = 0 0

1104/0x28f1: map\_with\_linking\_np(0x16B73CFF0, 0x1, 0x16B73D020) = 0 0

1104/0x28f1: close(0x3) = 0 0

1104/0x28f1: mprotect(0x1046C4000, 0x4000, 0x1) = 0 0

1104/0x28f1: shared\_region\_check\_np(0xFFFFFFFFFFFFFFFF, 0x0, 0x0) = 0 0

1104/0x28f1: access("/AppleInternal/XBS/.isChrooted\0", 0x0, 0x0) = -1 Err#2

1104/0x28f1: bsdthread\_register(0x18C41FE24, 0x18C41FE18, 0x4000) = 1073742303 0

1104/0x28f1: shm\_open(0x18C2E4F52, 0x0, 0x6B73EAE0) = 3 0

1104/0x28f1: fstat64(0x3, 0x16B73DE90, 0x0) = 0 0

1104/0x28f1: mmap(0x0, 0x4000, 0x1, 0x40001, 0x3, 0x0) = 0x104758000 0

1104/0x28f1: close(0x3) = 0 0

1104/0x28f1: ioctl(0x2, 0x4004667A, 0x16B73DF3C) = 0 0

1104/0x28f1: mprotect(0x104764000, 0x4000, 0x0) = 0 0

1104/0x28f1: mprotect(0x104770000, 0x4000, 0x0) = 0 0

1104/0x28f1: mprotect(0x104774000, 0x4000, 0x0) = 0 0

1104/0x28f1: mprotect(0x104780000, 0x4000, 0x0) = 0 0

1104/0x28f1: mprotect(0x104784000, 0x4000, 0x0) = 0 0

1104/0x28f1: mprotect(0x104790000, 0x4000, 0x0) = 0 0

1104/0x28f1: mprotect(0x10475C000, 0x98, 0x1) = 0 0

1104/0x28f1: mprotect(0x10475C000, 0x98, 0x3) = 0 0

1104/0x28f1: mprotect(0x10475C000, 0x98, 0x1) = 0 0

1104/0x28f1: mprotect(0x104794000, 0x4000, 0x1) = 0 0

1104/0x28f1: mprotect(0x104798000, 0x98, 0x1) = 0 0

1104/0x28f1: mprotect(0x104798000, 0x98, 0x3) = 0 0

1104/0x28f1: mprotect(0x104798000, 0x98, 0x1) = 0 0

1104/0x28f1: mprotect(0x10475C000, 0x98, 0x3) = 0 0

1104/0x28f1: mprotect(0x10475C000, 0x98, 0x1) = 0 0

1104/0x28f1: mprotect(0x104794000, 0x4000, 0x3) = 0 0

1104/0x28f1: mprotect(0x104794000, 0x4000, 0x1) = 0 0

1104/0x28f1: objc\_bp\_assist\_cfg\_np(0x18C0B9800, 0x80000018001C1048, 0x0) = -1 Err#5

1104/0x28f1: issetugid(0x0, 0x0, 0x0) = 0 0

1104/0x28f1: getentropy(0x16B73D988, 0x20, 0x0) = 0 0

1104/0x28f1: getpid(0x0, 0x0, 0x0) = 1104 0

1104/0x28f1: csops(0x450, 0x10, 0x16B73DFA0) = 0 0

1104/0x28f1: csops\_audittoken(0x450, 0x10, 0x16B73E000) = 0 0

1104/0x28f1: proc\_info(0x2, 0x450, 0xD) = 64 0

1104/0x28f1: csops\_audittoken(0x450, 0x10, 0x16B73E090) = 0 0

1104/0x28f1: sysctl([unknown, 3, 0, 0, 0, 0] (2), 0x16B73E3D0, 0x16B73E3C0, 0x18F288D3D, 0x15) = 0 0

1104/0x28f1: sysctl([CTL\_KERN, 134, 0, 0, 0, 0] (2), 0x16B73E478, 0x16B73E460, 0x0, 0x0) = 0 0

1104/0x28f1: csops(0x450, 0x0, 0x16B73E52C) = 0 0

1104/0x28f1: mprotect(0x1046D0000, 0x40000, 0x1) = 0 0

1104/0x28f1: getrlimit(0x1008, 0x16B73F3A8, 0x0) = 0 0

1104/0x28f1: fstat64(0x1, 0x16B73F3A0, 0x0) = 0 0

1104/0x28f1: ioctl(0x1, 0x4004667A, 0x16B73F3EC) = 0 0

1104/0x28f1: write\_nocancel(0x1, "Enter the number of threads: \0", 0x1D) = 29 0

1104/0x28f1: fstat64(0x0, 0x16B73F480, 0x0) = 0 0

1104/0x28f1: ioctl(0x0, 0x4004667A, 0x16B73F4CC) = 0 0

4

Enter the array size: 1104/0x28f1: read\_nocancel(0x0, "4\n\0", 0x1000) = 2 0

1104/0x28f1: write\_nocancel(0x1, "\n\0", 0x1) = 1 0

1104/0x28f1: write\_nocancel(0x1, "Enter the array size: \0", 0x16) = 22 0

100

The input array: 69 48 96 89 68 25 27 85 86 34 44 42 86 55 16 6 31 3 68 45 73 11 14 72 1 42 31 42 85 76 3 70 77 57 8 10 52 42 88 97 15 10 57 9 14 19 0 59 67 30 44 44 8 48 19 47 21 11 16 73 18 1 37 37 57 9 72 62 68 51 3 84 63 2 44 51 18 83 64 19 23 97 28 32 42 84 68 73 17 99 25 80 75 11 98 94 61 69 30 90

part- 0

part- 0

part- 1

part- 2

The output array:

0

1

1

2

3

3

3

6

8

8

9

9

10

10

11

11

11

14

14

15

16

16

17

18

18

19

19

19

21

23

25

25

27

28

30

30

31

31

32

34

37

37

42

42

42

42

42

44

44

44

44

45

47

48

48

51

51

52

55

57

57

57

59

61

62

63

64

67

68

68

68

68

69

69

70

72

72

73

73

73

75

76

77

80

83

84

84

85

85

86

86

88

89

90

94

96

97

97

98

99

The number of threads:4

282

1104/0x28f1: read\_nocancel(0x0, "100\n\0", 0x1000) = 4 0

1104/0x28f1: write\_nocancel(0x1, "The input array: 69 48 96 89 68 25 27 85 86 34 44 42 86 55 16 6 31 3 68 45 73 11 14 72 1 42 31 42 85 76 3 70 77 57 8 10 52 42 88 97 15 10 57 9 14 19 0 59 67 30 44 44 8 48 19 47 21 11 16 73 18 1 37 37 57 9 72 62 68 51 3 84 63 2 44 51 18 83 64 19 23 97 28 32", 0x132) = 306 0

1104/0x28f1: write\_nocancel(0x1, "part- 0\n)\325\260\234P\004\0", 0x8) = 8 0

1104/0x28f1: bsdthread\_create(0x1046C26B4, 0x0, 0x16B7C7000) = 1803317248 0

1104/0x28f1: write\_nocancel(0x1, "part- 0\n\0", 0x8) = 8 0

1104/0x295c: fork() = 0 0

1104/0x295c: thread\_selfid(0x0, 0x0, 0x0) = 10588 0

1104/0x28f1: bsdthread\_create(0x1046C26B4, 0x0, 0x16B853000) = 1803890688 0

1104/0x295d: fork() = 0 0

1104/0x28f1: write\_nocancel(0x1, "part- 1\n\0", 0x8) = 8 0

1104/0x295d: thread\_selfid(0x0, 0x0, 0x0) = 10589 0

1104/0x295c: \_\_disable\_threadsignal(0x1, 0x0, 0x0) = 0 0

1104/0x28f1: bsdthread\_create(0x1046C26B4, 0x0, 0x16B8DF000) = 1804464128 0

1104/0x28f1: write\_nocancel(0x1, "part- 2\n\0", 0x8) = 8 0

1104/0x295d: \_\_disable\_threadsignal(0x1, 0x0, 0x0) = 0 0

1104/0x295e: fork() = 0 0

1104/0x28f1: bsdthread\_create(0x1046C26B4, 0x0, 0x16B96B000) = 1805037568 0

1104/0x295e: thread\_selfid(0x0, 0x0, 0x0) = 10590 0

1104/0x295f: fork() = 0 0

1104/0x295f: thread\_selfid(0x0, 0x0, 0x0) = 10591 0

1104/0x295e: \_\_disable\_threadsignal(0x1, 0x0, 0x0) = 0 0

1104/0x295f: \_\_disable\_threadsignal(0x1, 0x0, 0x0) = 0 0

1104/0x295e: ulock\_wake(0x1000002, 0x16B8DF034, 0x0) = 0 0

1104/0x28f1: ulock\_wait(0x1020002, 0x16B8DF034, 0x1C03) = 0 0

1104/0x28f1: write\_nocancel(0x1, "The output array: \n\0", 0x13) = 19 0

1104/0x28f1: write\_nocancel(0x1, "0\n\0", 0x2) = 2 0

1104/0x28f1: write\_nocancel(0x1, "1\n\0", 0x2) = 2 0

1104/0x28f1: write\_nocancel(0x1, "1\n\0", 0x2) = 2 0

1104/0x28f1: write\_nocancel(0x1, "2\n\0", 0x2) = 2 0

1104/0x28f1: write\_nocancel(0x1, "3\n\0", 0x2) = 2 0

1104/0x28f1: write\_nocancel(0x1, "3\n\0", 0x2) = 2 0

1104/0x28f1: write\_nocancel(0x1, "3\n\0", 0x2) = 2 0

1104/0x28f1: write\_nocancel(0x1, "6\n\0", 0x2) = 2 0

1104/0x28f1: write\_nocancel(0x1, "8\n\0", 0x2) = 2 0

1104/0x28f1: write\_nocancel(0x1, "8\n\0", 0x2) = 2 0

1104/0x28f1: write\_nocancel(0x1, "9\n\0", 0x2) = 2 0

1104/0x28f1: write\_nocancel(0x1, "9\n\0", 0x2) = 2 0

1104/0x28f1: write\_nocancel(0x1, "10\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "10\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "11\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "11\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "11\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "14\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "14\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "15\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "16\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "16\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "17\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "18\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "18\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "19\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "19\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "19\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "21\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "23\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "25\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "25\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "27\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "28\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "30\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "30\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "31\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "31\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "32\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "34\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "37\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "37\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "42\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "42\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "42\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "42\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "42\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "44\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "44\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "44\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "44\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "45\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "47\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "48\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "48\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "51\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "51\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "52\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "55\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "57\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "57\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "57\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "59\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "61\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "62\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "63\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "64\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "67\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "68\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "68\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "68\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "68\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "69\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "69\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "70\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "72\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "72\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "73\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "73\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "73\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "75\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "76\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "77\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "80\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "83\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "84\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "84\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "85\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "85\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "86\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "86\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "88\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "89\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "90\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "94\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "96\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "97\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "97\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "98\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "99\n\0", 0x3) = 3 0

1104/0x28f1: write\_nocancel(0x1, "The number of threads:4\n\0", 0x18) = 24 0

1104/0x28f1: write\_nocancel(0x1, "282\n\0", 0x4) = 4 0

1104/0x28f1: lseek(0x0, 0xFFFFFFFFFFFFFFFF, 0x1) = 51537 0

**Выводы**

На сегодняшний день практически все программы используют потоки. Проделав лабораторную работу, я приобрела практические навыки в управлении ими в ОС и обеспечила синхронизацию между ними.