Харківський національний університет імені В.Н. Каразіна
Факультет комп’ютерних наук

Звіт
з лабораторної роботи №6
дисципліна: «Операційні системи»

Виконала: студентка групи КС34
Євстігнєєва Марія Олексіївна
Перевірив: Паршинцев Богдан Володимирович

Харків 2022

Задание №1 (Атрибуты потока)

Напишите программу, которая с помощью командной строки получает параметр — количество создаваемых потоков. Из основной функции вызывается функция, куда передается рабочий массив, в которой создается заданное количество расчетных потоков – потомков и ожидается завершение их работы. (В упрощенном варианте можно все это сделать в функции main.) Каждый расчетный поток получает в качестве аргумента номер потока выполнения (отсчет с 0), засыпает на это количество секунд и генерирует псевдослучайное число в диапазоне $[1,10]$. Затем, полученное псевдослучайное число заносится в глобальный массив целых чисел в ячейку, соответствующую номеру потока, а поток засыпает на это сгенерированное количество секунд. Параллельно с расчетными потоками запускается отсоединенный поток, который постоянно с периодом 1 секунда выводит на экран сообщение, отображающее значения, хранящиеся в массиве. Когда массив станет полностью заполненным, отсоединенный поток завершает свою работу. Предусмотреть информационные сообщения, поясняющие работу программы.

Задание №2 (Асинхронно отменяемый поток)

Напишите программу, которая с помощью командной строки получает параметр — время задержки. В основной программе создается поток – потомок и ожидается завершение его работы. Поток–потомок устанавливается в режим асинхронно отменяемого потока. Поток выводит строку с текстом (включающим номер итерации) бесконечное количество раз и «засыпает» на секунду после каждого вывода. После создания потока основной поток засыпает на заданное количество секунд (время задержки) и пытается отменить работающий поток. Затем анализируется статус завершения потока и выводится сообщение о том, как же завершился поток — в результате отмены или обычным образом.

Задание №3 (Не отменяемый поток)

Напишите программу, которая с помощью командной строки получает параметр — время задержки. В основной программе создается поток – потомок и ожидается завершение его работы. Поток–потомок устанавливается в не отменяемое состояние. Поток выводит строку с текстом (включающим номер итерации) заданное количество раз (оно равно удвоенному времени задержки) и «засыпает» на секунду после каждого вывода. После создания потока основной поток засыпает на заданное количество секунд (время задержки) и пытается отменить работающий поток. Затем анализируется статус завершения потока и выводится сообщение о том, как же завершился поток — в результате отмены или обычным образом.

Задание №4 (Синхронно отменяемый поток)

Напишите программу, которая с помощью командной строки получает параметр — время задержки. В основной программе создается поток – потомок и ожидается завершение его работы. Данный поток – потомок устанавливается в режим синхронно отменяемого потока. В нем вычисляется значение числа π по формуле Лейбница:

$$\frac{\pi}{4}=1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\frac{1}{11}-\dots$$

Вычисления продолжаются до $n=100000$. Текущее значение числа π выводится в стандартный поток вывода. Вычисление очередного слагаемого и добавление его в сумму происходит в не отменяемой секции. Сразу после этого формируется возможная точка выхода.

Основной поток ожидает заданное при запуске программы количество секунд и посылает сигнал на отмену потока. Затем анализируется: в результате чего завершился поток. Если в результате отмены — выводится сообщение. Если в поток досчитал до конца, то выводится результат.

Задание №5 (Потоковые данные)

Напишите программу, которая с помощью командной строки получает параметр — количество создаваемых потоков. В основной программе (для этого можно сделать отдельную функцию) создается заданное количество потоков – потомков и ожидается завершение их работы. Каждый поток выводит в стандартный поток вывода случайное количество строк, со случайными числами. Случайные числа генерируются из заданного диапазона. Каждая строка включает в себя идентификатор потока, текстовое сообщение и сгенерированное псевдослучайное число. Неизменяемая часть строки хранится в области потоковых данных для каждого потока. Используйте функцию sleep, чтобы сделать работу программы более наглядной.

Задание №6 (Обработчики очистки)

Модифицируйте программу из **Задания №2** так, чтобы поток – потомок перед своим завершением выводил в стандартный поток ошибок сообщение об этом с данными, полученными из завершаемого потока. Используйте функции pthread_cleanup_push, pthread_cleanup_pop.

Результати:

```
[thread 0] sleeping for 0 sec...
[generating number 0] 7


[thread 0] sleeping for 7 sec...
[control] Array for now
[control] [0] 7
[control] [1] 0
[control] [2] 0
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 0
[control] [2] 0
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0
```

```
[control] Array for now
[control] [0] 7
[control] [1] 0
[control] [2] 0
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 0
[control] [2] 0
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 0
[control] [2] 0
```

```
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 0
[control] [2] 0
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[thread 1] sleeping for 1 sec...
[control] Array for now
[control] [0] 7
[control] [1] 0
[control] [2] 0
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
```

```
[control] [7] 0
[control] [8] 0
[control] [9] 0

[generating number 1] 3


[thread 1] sleeping for 3 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 0
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 0
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
```

```
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 0
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[thread 2] sleeping for 2 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 0
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
```

```
[control] [0] 7
[control] [1] 3
[control] [2] 0
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[generating number 2] 3


[thread 2] sleeping for 3 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
```

```
[control] [1] 3
[control] [2] 3
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[thread 3] sleeping for 3 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 0
[control] [4] 0
```

```
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 0
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0
```

```
[generating number 3] 2


[thread 3] sleeping for 2 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0
```

```
[thread 4] sleeping for 4 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
```

```
[control] [3] 2
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 0
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[generating number 4] 5



[thread 4] sleeping for 5 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
```

```
[control] [4] 5
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
```

```
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[thread 5] sleeping for 5 sec...
[control] Array for now
[control] [0] 7
```

```
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 0
```

```
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 0
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0
```

```
[generating number 5] 1


[thread 5] sleeping for 1 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[thread 6] sleeping for 6 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0
```

```
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
```

```
[control] [4] 5
[control] [5] 1
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 0
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 0
[control] [7] 0
[control] [8] 0
```

```
[control] [9] 0

[generating number 6] 9


[thread 6] sleeping for 9 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0
```

```
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
```

```
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
```

```
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[thread 7] sleeping for 7 sec...
[control] Array for now
```

```
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
```

```
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0
```

```
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 0
[control] [8] 0
[control] [9] 0

[generating number 7] 6


[thread 7] sleeping for 6 sec...
```

```
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
```

```
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
```

```
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0

[thread 8] sleeping for 8 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
```

```
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
```

```
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0
```

```
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 0
[control] [9] 0

[generating number 8] 2



[thread 8] sleeping for 2 sec...
[control] Array for now
```

```
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 2
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 2
[control] [9] 0

[thread 9] sleeping for 9 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
```

```
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 2
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 2
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 2
```

```
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 2
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 2
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
```

```
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 2
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 2
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
```

```
[control] [7] 6
[control] [8] 2
[control] [9] 0

[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
[control] [8] 2
[control] [9] 0

[generating number 9] 2



[thread 9] sleeping for 2 sec...
[control] Array for now
[control] [0] 7
[control] [1] 3
[control] [2] 3
[control] [3] 2
[control] [4] 5
[control] [5] 1
[control] [6] 9
[control] [7] 6
```

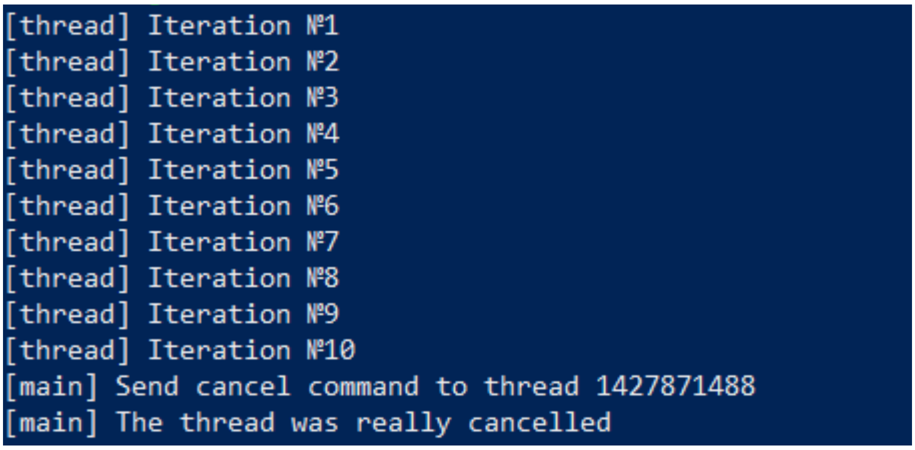
```
[control] [8] 2

[control] [9] 2



Final array

[Main] [0] 7

[Main] [1] 3

[Main] [2] 3

[Main] [3] 2

[Main] [4] 5

[Main] [5] 1

[Main] [6] 9

[Main] [7] 6

[Main] [8] 2

[Main] [9] 2
```

Код:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>

struct thread_arg {
    int num;
    int *arr;
};

static void make_detached(struct thread_arg * arr);
void * threadFunc(void * args);
void * myFunc(void * args);
void * control(void * args);

int main(int argc, char *argv[]) {
    if (argc == 1) {
        fprintf(stderr, "\nYou didn't enter the number\n");
        return 0;
    }

    int number = atoi(argv[1]);

    pthread_t thread;
    struct thread_arg targ;
    targ.num = number;
    targ.arr = malloc(sizeof(int) * targ.num);
    if(!targ.arr) {
        fprintf(stderr, "ERROR! %s\n", "Cannot Allocate Working Array");
        exit(1);
    }

    make_detached(&targ);

    srand(time(NULL));
    int result = pthread_create(&thread, NULL, &threadFunc, &targ);

    if(result != 0) {
        fprintf(stderr, "Error!\n");
        return 1;
    }

    if (pthread_join(thread, NULL) != 0) {
        fprintf(stderr, "Join error\n");
        return 2;
    }

    fprintf(stderr, "\n\nFinal array\n");
    for(int i = 0; i < targ.num; i++) {
        fprintf(stderr, "[Main] [%d] %d\n", i, targ.arr[i]);
    }
}

void * threadFunc(void * args){
    struct thread_arg targ = *(struct thread_arg *) args;

    pthread_t *tid_arr;
    tid_arr = (pthread_t*)calloc(targ.num, sizeof(pthread_t));

    for(int i = 0; i < targ.num; i++) {
        struct thread_arg targs;
        targs.num = i;
        targs.arr = targ.arr;

        int result = pthread_create(tid_arr+i, NULL, &myFunc, &targs);

        if(result != 0) {
            fprintf(stderr, "[thread] Error!\n");
        }

        if (pthread_join(*(tid_arr+i), NULL) != 0) {
            fprintf(stderr, "[thread] Join error\n");
        }
    }
}

void * myFunc(void * args) {
    struct thread_arg targ = *(struct thread_arg *) args;

    fprintf(stderr, "\n[thread %d] sleeping for %d sec...", targ.num, targ.num);
    sleep(targ.num);

    int min = 1, max = 10;
    int number = rand()%(max - min) + 1;
    targ.arr[targ.num] = number;
    fprintf(stderr, "\n[generating number %d] %d\n\n", targ.num,
targ.arr[targ.num]);

    fprintf(stderr, "\n[thread %d] sleeping for %d sec...", targ.num, number);
    sleep(number);
}

void * control(void * args){
    struct thread_arg targ = *(struct thread_arg *) args;
    int flag = 1;

    while(flag) {
        sleep(1);
        flag = 0;
        fprintf(stderr, "\n[control] Array for now\n");
        for(int i = 0; i < targ.num; i++) {
            fprintf(stderr, "[control] [%d] %d\n", i, targ.arr[i]);
            if(targ.arr[i] == 0)
                flag = 1;
        }
    }
}

static void make_detached(struct thread_arg * arr) {
    pthread_attr_t attr;
    pthread_t thread;

    pthread_attr_init(&attr);
    pthread_attr_setdetachstate(&attr, PTHREAD_CREATE_DETACHED);
    if (pthread_create(&thread, &attr, &control, (void*)arr) != 0) {
        fprintf(stderr, "Error while Creation Detached Thread\n");
        exit(EXIT_FAILURE);
    }
    pthread_attr_destroy(&attr);
}
```

Завдання №2:

Результат:



```
[thread] Iteration №1
[thread] Iteration №2
[thread] Iteration №3
[thread] Iteration №4
[thread] Iteration №5
[thread] Iteration №6
[thread] Iteration №7
[thread] Iteration №8
[thread] Iteration №9
[thread] Iteration №10
[main] Send cancel command to thread 1427871488
[main] The thread was really cancelled
```

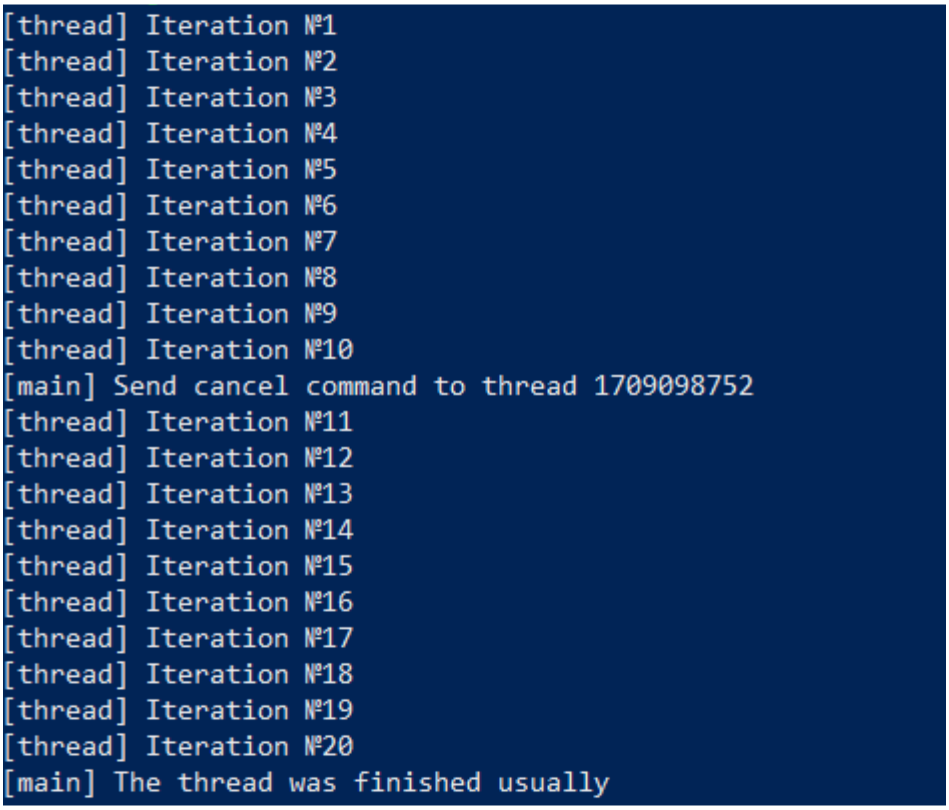
Результати виконання програми

Код:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <unistd.h>

void * threadFunc(void * args);

int main(int argc, char *argv[]) {
    if (argc == 1) {
        fprintf(stderr, "\n[main] You didn't enter the number\n");
        return 0;
    }

    int sec = atoi(argv[1]);

    pthread_t thread;
    int result = pthread_create(&thread, NULL, &threadFunc, NULL);

    if(result != 0) {
        fprintf(stderr, "[main] Error!\n");
        return 1;
    }

    sleep(sec);
    printf("[main] Send cancel command to thread %u\n", (unsigned
int)thread);
    pthread_cancel(thread);

    void* iscanceled;

    if (pthread_join(thread, &iscanceled) != 0) {
        fprintf(stderr, "[main] Join error\n");
        return 2;
    }

    if (iscanceled == PTHREAD_CANCELED)
        fprintf(stderr, "main] The thread was really cancelled\n");
    else
        fprintf(stderr, "main] The thread was finished usually\n");

}

void * threadFunc(void * args){
    int i = 0;

    if(pthread_setcanceltype(PTHREAD_CANCEL_ASYNCHRONOUS, NULL) != 0) {
        fprintf(stderr, "[thread] Cannot change cancel type\n");
        exit(EXIT_FAILURE);
    }

    while(1) {
        fprintf(stderr, "[thread] Iteration №%d\n", ++i);
        sleep(1);
    }
}
```

Завдання №3

Результат:



```
[thread] Iteration №1
[thread] Iteration №2
[thread] Iteration №3
[thread] Iteration №4
[thread] Iteration №5
[thread] Iteration №6
[thread] Iteration №7
[thread] Iteration №8
[thread] Iteration №9
[thread] Iteration №10
[main] Send cancel command to thread 1709098752
[thread] Iteration №11
[thread] Iteration №12
[thread] Iteration №13
[thread] Iteration №14
[thread] Iteration №15
[thread] Iteration №16
[thread] Iteration №17
[thread] Iteration №18
[thread] Iteration №19
[thread] Iteration №20
[main] The thread was finished usually
```

Результати виконання програми

Код:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <unistd.h>

struct thread_arg {
   int num;
};

void * threadFunc(void * args);

int main(int argc, char *argv[]) {
   if (argc == 1) {
      fprintf(stderr, "\n[main] You didn't enter the number\n");
      return 0;
   }

   int sec = atoi(argv[1]);

   pthread_t thread;
   struct thread_arg targ;
   targ.num = sec*2;
   int result = pthread_create(&thread, NULL, &threadFunc, &targ);

   if(result != 0) {
      fprintf(stderr, "[main] Error!\n");
```

```
        return 1;
    }

    sleep(sec);
    printf("[main] Send cancel command to thread %u\n", (unsigned int)thread);
    pthread_cancel(thread);

    void* iscanceled;

    if (pthread_join(thread, &iscanceled) != 0) {
        fprintf(stderr, "[main] Join error\n");
        return 2;
    }

    if (iscanceled == PTHREAD_CANCELED)
        fprintf(stderr, "[main] The thread was really cancelled\n");
    else
        fprintf(stderr, "[main] The thread was finished usually\n");

}

void * threadFunc(void * args){
    struct thread_arg targ = *(struct thread_arg *) args;

    if(pthread_setcancelstate(PTHREAD_CANCEL_DISABLE, NULL) != 0) {
        fprintf(stderr, "[thread] Cannot change cancel state\n");
        exit(EXIT_FAILURE);
    }

    for(int i = 0; i < targ.num; i++) {
        fprintf(stderr, "[thread] Iteration №%d\n", i+1);
        sleep(1);
    }
}
```

Завдання №4:

Результат (скопійовані останні рядки, бо всі не вмістилися):

1 випадок (задали 1 секунду):

```
55157          3.14158358855726937975
55158          3.14158358872161613817
55159          3.14158358888595667935
55160          3.14158358905029144736
55161          3.14158358921462044222
55162          3.14158358937894321983
55163          3.14158358954326022427
```

```
55164           3.14158358970757101147
55165           3.14158358987187602551
[main] Send cancel command to thread 963401472
55166           3.14158359003617482230
55167           3.14158359020046784593
55168           3.14158359036475509640
55169           3.14158359052903612962
55170           3.14158359069331138969
55171           3.14158359085758043250
55172           3.14158359102184370215
55173           3.14158359118610119864
55174           3.14158359135035247789
55175           3.14158359151459798397
55176           3.14158359167883727281
55177           3.14158359184307078849
55178           3.14158359200729853100
55179           3.14158359217152005627
55180           3.14158359233573580838
55181           3.14158359249994534323
55182           3.14158359266414910493
55183           3.14158359282834709347
55184           3.14158359299253886476
55185           3.14158359315672486289
55186           3.14158359332090464378
55187           3.14158359348507865150
55188           3.14158359364924688606
55189           3.14158359381340890337
55190           3.14158359397756514753
55191           3.14158359414171517443
```

55192	3.14158359430585942818
55193	3.14158359446999790876
55194	3.14158359463413017210
55195	3.14158359479825666227
55196	3.14158359496237693520
55197	3.14158359512649143497
55198	3.14158359529060016158
55199	3.14158359545470267093
55200	3.14158359561879940713
55201	3.14158359578288992608
55202	3.14158359594697467188
55203	3.14158359611105364451
55204	3.14158359627512639989
55205	3.14158359643919338211
55206	3.14158359660325414708
55207	3.14158359676730913890
55208	3.14158359693135835755
55209	3.14158359709540135896
55210	3.14158359725943858720
55211	3.14158359742347004229
55212	3.14158359758749528012
55213	3.14158359775151474480
55214	3.14158359791552799223
55215	3.14158359807953546650
55216	3.14158359824353716760
55217	3.14158359840753265146
55218	3.14158359857152236216
55219	3.14158359873550629970
55220	3.14158359889948402000

```
55221           3.14158359906345596713
55222           3.14158359922742169701
55223           3.14158359939138165373
55224           3.14158359955533583729
55225           3.14158359971928380361
55226           3.14158359988322599676
55227           3.14158360004716241676
55228           3.14158360021109261950
55229           3.14158360037501704909
55230           3.14158360053893526143
55231           3.14158360070284770060
55232           3.14158360086675436662
55233           3.14158360103065481539
55234           3.14158360119454949100
55235           3.14158360135843839345
55236           3.14158360152232107865
55237           3.14158360168619799069
55238           3.14158360185006912957
55239           3.14158360201393405120
55240           3.14158360217779319967
[main] The thread was really cancelled
```

2 випадок (задали 3 секунди):

```
99916           3.14158764938628820573
99917           3.14158764943637169864
99918           3.14158764948645430337
99919           3.14158764953653601992
99920           3.14158764958661640421
99921           3.14158764963669590031
99922           3.14158764968677450824
```

99923	3.14158764973685222799
99924	3.14158764978692861547
99925	3.14158764983700411477
99926	3.14158764988707872590
99927	3.14158764993715244884
99928	3.14158764998722483952
99929	3.14158765003729634202
99930	3.14158765008736695634
99931	3.14158765013743668248
99932	3.14158765018750507636
99933	3.14158765023757258206
99934	3.14158765028763919958
99935	3.14158765033770492892
99936	3.14158765038776932599
99937	3.14158765043783283488
99938	3.14158765048789545560
99939	3.14158765053795718813
99940	3.14158765058801758840
99941	3.14158765063807710050
99942	3.14158765068813572441
99943	3.14158765073819346014
99944	3.14158765078824986361
99945	3.14158765083830537890
99946	3.14158765088836000601
99947	3.14158765093841374494
99948	3.14158765098846615160
99949	3.14158765103851767009
99950	3.14158765108856830039
99951	3.14158765113861804252

99952	3.14158765118866645238
99953	3.14158765123871397407
99954	3.14158765128876060757
99955	3.14158765133880635290
99956	3.14158765138885076595
99957	3.14158765143889429083
99958	3.14158765148893692754
99959	3.14158765153897823197
99960	3.14158765158901864822
99961	3.14158765163905817630
99962	3.14158765168909681620
99963	3.14158765173913412383
99964	3.14158765178917054328
99965	3.14158765183920607456
99966	3.14158765188924071765
99967	3.14158765193927402848
99968	3.14158765198930645113
99969	3.14158765203933798560
99970	3.14158765208936863189
99971	3.14158765213939794592
99972	3.14158765218942637176
99973	3.14158765223945390943
99974	3.14158765228948055892
99975	3.14158765233950587614
99976	3.14158765238953030519
99977	3.14158765243955384605
99978	3.14158765248957649874
99979	3.14158765253959781916
99980	3.14158765258961825140

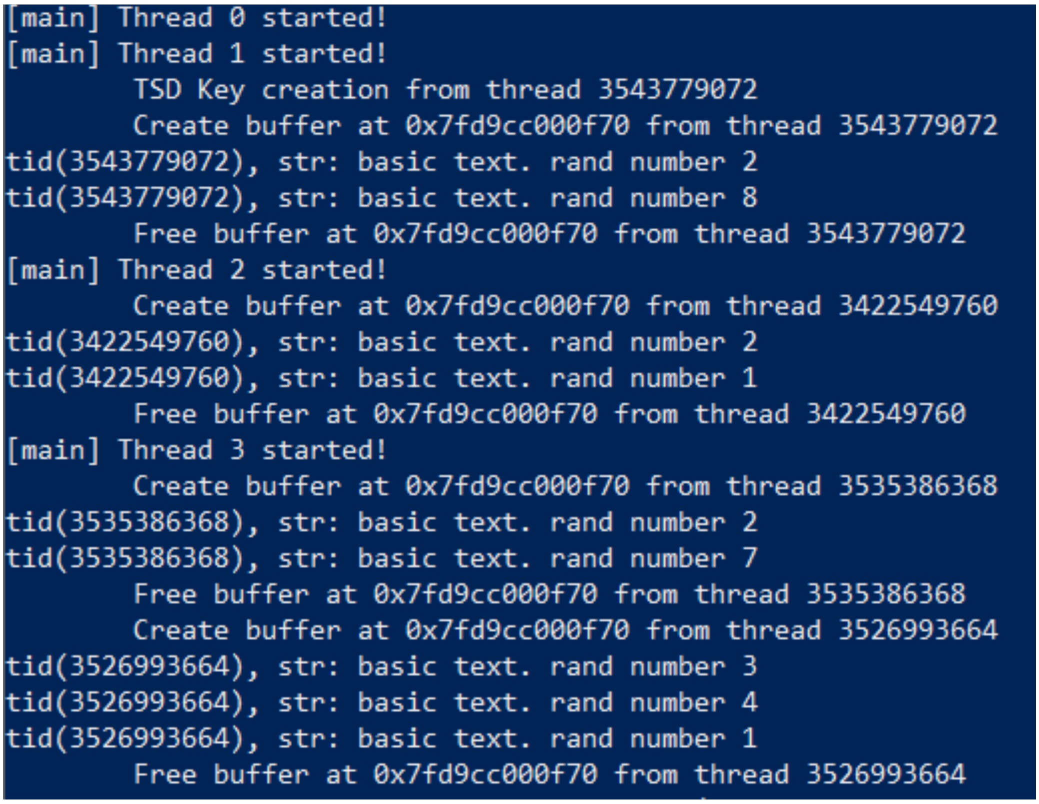
```
99981          3.14158765263963779546
99982          3.14158765268965645134
99983          3.14158765273967377496
99984          3.14158765278969021040
99985          3.14158765283970575766
99986          3.14158765288972041674
99987          3.14158765293973374355
99988          3.14158765298974618219
99989          3.14158765303975773264
99990          3.14158765308976839492
99991          3.14158765313977772493
99992          3.14158765318978616676
99993          3.14158765323979372042
99994          3.14158765328980038589
99995          3.14158765333980571910
99996          3.14158765338981016413
99997          3.14158765343981372098
99998          3.14158765348981638965
99999          3.14158765353981772606
100000          3.14158765358981817428
[main] Send cancel command to thread 1191208704
[main] The thread was finished usually
[main] Final number: 3.14158765358981817428
```

Код:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <unistd.h>

double resultPi;

void * threadFunc(void * args);

int main(int argc, char *argv[]) {
    if (argc == 1) {
        fprintf(stderr, "\n[main] You didn't enter the number\n");
        return 0;
    }

    int sec = atoi(argv[1]);

    pthread_t thread;
    int result = pthread_create(&thread, NULL, &threadFunc, NULL);

    if(result != 0) {
        fprintf(stderr, "[main] Error!\n");
        return 1;
    }

    sleep(sec);
    printf("[main] Send cancel command to thread %u\n", (unsigned
int)thread);
    pthread_cancel(thread);

    void* iscanceled;

    if (pthread_join(thread, &iscanceled) != 0) {
        fprintf(stderr, "[main] Join error\n");
        return 2;
    }

    if (iscanceled == PTHREAD_CANCELED)
        fprintf(stderr, "[main] The thread was really cancelled\n");
    else {
        fprintf(stderr, "[main] The thread was finished usually\n");
        fprintf(stderr, "[main] Final number: %.20f\n", resultPi);
    }

}

void * threadFunc(void * args){
    int i = 0;

    if(pthread_setcanceltype(PTHREAD_CANCEL_DEFERRED, NULL) != 0) {
        fprintf(stderr, "[thread] Cannot change cancel type\n");
        exit(EXIT_FAILURE);
    }

    int count = 100000, count1;
    double pi, a, b;

    fprintf(stderr, "\n%s  %s", "Approximation", "Number PI\n");

    int old_cancelled_state;
    for (count, count1 = 1; count != 0; count--, count1++) {
        pthread_setcancelstate(PTHREAD_CANCEL_DISABLE,
&old_cancelled_state);
        if (count1 == 1) {
            a = 4 - 1;
            b = 1;
            pi = 4 / b - 4 / a;
            fprintf(stderr, "%d              %.20f\n", count1, pi);
        }
        if (count1 > 1) {
            a = a + 4;
            b = b + 4;
            pi = pi + (4 / b - 4 / a);
            fprintf(stderr, "%d             %.20f\n", count1, pi);
        }
        pthread_setcancelstate(old_cancelled_state, NULL);
        pthread_testcancel();
    }
    resultPi = pi;
    return NULL;
}
```

Завдання №5:

Результат:



```
[main] Thread 0 started!
[main] Thread 1 started!
        TSD Key creation from thread 3543779072
        Create buffer at 0x7fd9cc000f70 from thread 3543779072
tid(3543779072), str: basic text. rand number 2
tid(3543779072), str: basic text. rand number 8
        Free buffer at 0x7fd9cc000f70 from thread 3543779072
[main] Thread 2 started!
        Create buffer at 0x7fd9cc000f70 from thread 3422549760
tid(3422549760), str: basic text. rand number 2
tid(3422549760), str: basic text. rand number 1
        Free buffer at 0x7fd9cc000f70 from thread 3422549760
[main] Thread 3 started!
        Create buffer at 0x7fd9cc000f70 from thread 3535386368
tid(3535386368), str: basic text. rand number 2
tid(3535386368), str: basic text. rand number 7
        Free buffer at 0x7fd9cc000f70 from thread 3535386368
        Create buffer at 0x7fd9cc000f70 from thread 3526993664
tid(3526993664), str: basic text. rand number 3
tid(3526993664), str: basic text. rand number 4
tid(3526993664), str: basic text. rand number 1
        Free buffer at 0x7fd9cc000f70 from thread 3526993664
```

Результати виконання програми

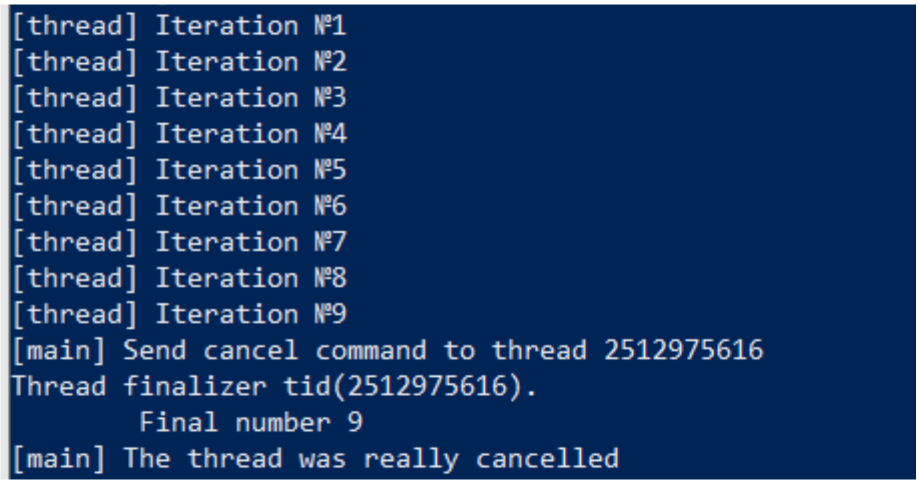
Код:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <string.h>

#define STR_MAX_LEN 256

static pthread_key_t str_key;

void * myFunc(void * args);
```

```
char *strConstr(char *string);

int main(int argc, char *argv[]){
    if (argc == 1) {
        fprintf(stderr, "\n[main] You didn't enter the number\n");
        return 0;
    }

    srand(time(NULL));

    int amount = atoi(argv[1]);
    pthread_t tid_arr [amount];

    static pthread_once_t once = PTHREAD_ONCE_INIT;


    for(int i = 0; i < amount; i++) {
        fprintf(stderr, "[main] Thread %d started!\n", i);
        if(pthread_create(tid_arr+i, NULL, &myFunc, "basic text. rand
number") != 0) {
            fprintf(stderr, "[main] Error!\n");
        }
    }

    for(int i = 0; i < amount; i++) {
        if (pthread_join(*(tid_arr+i), NULL) != 0) {
            fprintf(stderr, "[main] Join error\n");
        }
        sleep(1);
    }

}

void * myFunc(void * args) {
    char * str = (char *)args;

    int num = rand()%(5-1);

    for(int i = 0; i < num; i++) {
        int random = rand()%(10-1);
        printf("tid(%u), str: %s %d\n", (unsigned int) pthread_self(),
strConstr(str), random);
    }

    return NULL;
}

static void destructor(void * buf) {
    printf("\tFree buffer at %p from thread %u\n", buf, (unsigned
int)pthread_self());
    free(buf);
    buf = NULL;
}

static void create_key(void) {
    int res;

    printf("\tTSD Key creation from thread %u\n", (unsigned
int)pthread_self());
    res = pthread_key_create(&str_key, destructor);
    if (res) {
        fprintf(stderr, "str_to_upper: TSD Key Creation ERROR!!!\n");
        exit(EXIT_FAILURE);
    }
}

char *strConstr(char *string) {
    static pthread_once_t once = PTHREAD_ONCE_INIT;

    int res = pthread_once(&once, create_key);
    if (res) {
        fprintf(stderr, "str_to_upper: Once Initialization ERROR!!!\n");
        exit(EXIT_FAILURE);
    }

    char * buf = pthread_getspecific(str_key);
    if (buf == NULL) { //First call from this thread, create buffer and
store address
        buf = (char*)malloc(STR_MAX_LEN);
        if (buf == NULL) {
            fprintf(stderr, "str_to_upper: Buffer Allocation ERROR!!!\n");
            exit(EXIT_FAILURE);
        }
        printf("\tCreate buffer at %p from thread %u\n", buf, (unsigned
int)pthread_self());
        res = pthread_setspecific(str_key, buf);
        if (res) {
            fprintf(stderr, "str_to_upper: setspecific ERROR!!!\n");
            exit(EXIT_FAILURE);
        }
        int index;

        for (index = 0; string[index]; index++) {
            buf[index] = string[index];
        }
        buf[index] = '\0';
    }

    return buf;
}
```

Завдання №6:

Результат:



```
[thread] Iteration №1
[thread] Iteration №2
[thread] Iteration №3
[thread] Iteration №4
[thread] Iteration №5
[thread] Iteration №6
[thread] Iteration №7
[thread] Iteration №8
[thread] Iteration №9
[main] Send cancel command to thread 2512975616
Thread finalizer tid(2512975616).
        Final number 9
[main] The thread was really cancelled
```

Результати виконання програми

Код:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <unistd.h>

void * threadFunc(void * args);
void finalize(void * arg);

int main(int argc, char *argv[]) {
    if (argc == 1) {
        fprintf(stderr, "\n[main] You didn't enter the number\n");
        return 0;
    }

    int sec = atoi(argv[1]);

    pthread_t thread;
    int result = pthread_create(&thread, NULL, &threadFunc, NULL);

    if(result != 0) {
        fprintf(stderr, "[main] Error!\n");
        return 1;
    }

    sleep(sec);
    printf("[main] Send cancel command to thread %u\n", (unsigned
int)thread);
    pthread_cancel(thread);

    void* iscanceled;

    if (pthread_join(thread, &iscanceled) != 0) {
        fprintf(stderr, "[main] Join error\n");
        return 2;
    }

    if (iscanceled == PTHREAD_CANCELED)
        fprintf(stderr, "[main] The thread was really cancelled\n");
    else
        fprintf(stderr, "[main] The thread was finished usually\n");

}

void * threadFunc(void * args){
    int i = 0;
    int *arr = &i;

    pthread_cleanup_push(finalize, arr);

    if(pthread_setcanceltype(PTHREAD_CANCEL_ASYNCHRONOUS, NULL) != 0) {
        fprintf(stderr, "[thread] Cannot change cancel type\n");
        exit(EXIT_FAILURE);
    }

    while(1) {
        fprintf(stderr, "[thread] Iteration №%d\n", ++i);
        sleep(1);
    }

    pthread_cleanup_pop(1);

}

void finalize(void * arg) {
    int num = *((int*)arg);
    fprintf(stderr,"Thread finalizer tid(%u).\n\tFinal number %d\n",
           (unsigned int)pthread_self(), num);
}
```