Міністерство освіти і науки України

Харківський національний університет імені В.Н. Каразіна

Факультет комп’ютерних наук

# ЛАБОРАТОРНА РОБОТА № 5

з дисципліни «Операційні системи»

Тема: «Потоки. Основні операції»

Виконав:

студент 3 курсу

групи КС-32

Безрук Юрій Русланович

Перевірив: Споров О.Є.

Харків – 2020

Целью данной работы является ознакомление с понятием потока исполнения и основными операциями, связанными с потоками: созданием и завершением работы потока, передачей параметров и возвращение вычисленного результата, а также ожиданием одним потоком завершения другого.

# ХОД РАБОТЫ

Во всех заданиях по умолчанию отключена буферизация стандартного потока вывода в консоль.

**Задание 1**

В задании 1 требуется написать программу, создающую новый поток исполнения с атрибутами по умолчанию. Первоначальный и созданный поток должны вывести десять строк текста с некоторой задержкой.

«Дочерний поток» запускается в потоковой функции child\_thread(). Каждый из потоков выводит десять раз строку вида «name. Iteration: n». После каждого вывода поток засыпает на некоторое число секунд. Устанавливая разные значения, можно наблюдать различные ситуации конкуренции потоков. К примеру, если первый поток будет делать задержку в 2с. После каждого вывода, а второй – 1с., ожидается, что на каждую итерацию «главного» потока придется две итерации «дочернего». Однако, в следствии условий гонок, это не всегда так:

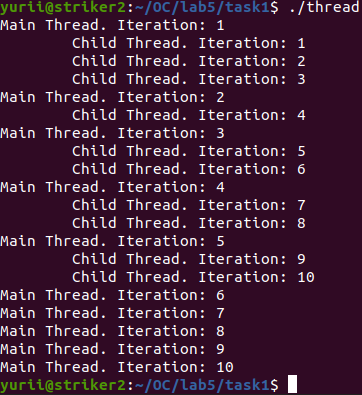


Рисунок 1 - пример конкуренции потоков

**Задание 2**

В задании 2 подробнее рассматривается передача параметров потоку, а именно: требуется создать 4 потока, которые выполняют одну и ту же потоковую функцию, которая принимает в качестве параметров имя потока, строку для вывода и количество ее повторений. Каждый поток должен получить свой набор параметров.

Т.к. потоковой функции можно передавать только указатель типа void\*, создаем структуру thread\_args, полями которой будут указанные выше поля. Далее, для каждого потока создаем переменную типа этой структуры, инициализируем ее нужными значениями и передаём в функцию pthread\_create(). В конце функции main() добавляем задержку в 1 секунду, для того, что бы созданные потоки успели завершить свою работу. Снова можно наблюдать конкуренцию потоков:

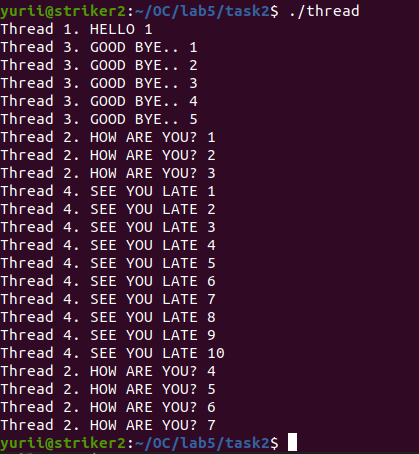


Рисунок 2 - условия гонок 4 потоков.

**Задание 3**

Задание 3 рассчитано на обучение использованию ожидания завершения потока. В ходе его выполнения необходимо создать два новых потока исполнения, один из которых выводит заданное количество раз строку аналогично тому как это было в задании 1. Другой поток генерирует заданное количество целых псевдослучайных чисел из заданного диапазона и выводит каждое из них в консоль похожим образом. Если во время его выполнения будет получено заданное число, поток завершается раньше времени. Основной поток дожидается завершения всех дочерних потоков.

Для удобства использования случайных чисел используются функции seed() и get\_random(). Первой потоковой функции в качестве аргументов передаётся просто адрес переменной, в которой хранится необходимое количество итераций цикла. Второй же потоковой функции передается адрес структуры аргументов, в которой содержится диапазон для псевдослучайных чисел, их количество, «цель» (значение, при возникновении которого поток досрочно завершится), и переменная, через которую будет передан код выхода потока. Все вышеперечисленные параметры имеют значения по умолчанию, но могут и быть считанными с консоли (для упрощения работы, можно либо передать все параметры при запуске, либо не передавать никаких).

После запусков потоков главный поток ожидает их завершения. В случае со вторым потоком, помимо завершения, главный поток получает адрес переменной, которая содержит код завершения потока. Если он равен коду досрочного завершения, поток выводит об этом информацию.

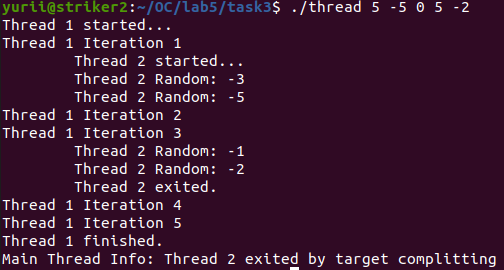


Рисунок 3 - пример ожидания потока

В данном примере первый поток выполняет пять итераций, второй генерирует пять псевдослучайных целых чисел в диапазоне от -5 до 0 и досрочно завершается, когда новым числом стаёт -2. Главный поток выводит соответствующее сообщение.

**Задание 4**

В задании 4 нужно написать программу, которая создает два новых потока исполнения, один из которых вычисляет заданное количество первых чисел Каталана, а другой – заданное количество первых простых чисел. Вычисленные массивы возвращаются в главный поток, который выводит их на экран.

Для передачи аргументов используется структура thread\_args, которая содержит количество чисел, которые нужно сгенерировать потоку и массив, который будет возвращён. Функция is\_simple() используется вторым потоком для определения, простое ли очередное число. При переборе делителей, они перебираются до значения корня анализируемого числа в целях экономии времени. Функция fact() используется первым потоком и рекурсивно считает факториал числа. Стоит отметить, что настоящие числа Каталана в программе можно получить только до 11 номера (n = 10), т.к. при вычислении выражения (2\*11)! получается число, превосходящее размеры даже типа данных unsigned long long.

Количество чисел, которые нужно вывести передаётся при помощи опций: -с для чисел Каталана, -s для простых чисел. (по умолчанию - 10)

Пример выполнения:



Рисунок 4 - числа Каталана и простые

# ВЫВОДЫ

Таким образом, в ходе выполнения данной работы были рассмотрены функции для создания новых потоков исполнения, их досрочного завершения, и ожидания. Исходные, исполняемые файлы и скрипты сборки для каждого задания помещены в архив.