Міністерство освіти і науки України

Харківський національний університет імені В.Н. Каразіна

Факультет комп’ютерних наук

ЛАБОРАТОРНА РОБОТА № 6

з дисципліни «Операційні системи»

Тема «Потоки. Управління»

Виконала:

студентка 3 курсу

групи КС-32

Дібцева Анна Миколаївна

Перевірив: Споров О.Є.

Харків – 2020

Целью данной работы является продолжение изучения потоков исполнения и знакомство с дополнительными возможностями управления, такими как атрибуты потоков, отмена потоков, потоковые данные и обработчики очистки.

# ХОД РАБОТЫ

Задание 1

В задании 1 требуется написать программу с использованием отсоединенного потока (потока, который не сохраняет информацию о себе после завершения, т.е. его не нужно ожидать). Программа создаёт указанное количество расчетных потоков, которые при создании засыпают на число секунд, соответствующее их номеру, генерируют случайное число в диапазоне [0, 10], помещают его в глобальный массив и засыпают уже на это сгенерированное число секунд. В тоже время отсоединённый поток раз в секунду выводит состояние массива, пока он не окажется заполнен.

При запуске программы число расчётных потоков получается из аргументов командной строки (по умолчанию - 1). После происходит выделение памяти под динамический глобальный массив. Далее создаётся массив идентификаторов потоков и вместе с количеством передаётся в функцию createThreads, в которой проиходит запуск и ожидание потоков

Потоки выполняют свою работу внутри функции threadFunc, а отсоединенный поток – внутри detachedThreadFunc.

Отсоединённый поток завершается, как только значение последнего єлемента станет не равным -1.

Задания 2-4 демонстрируют различные режимы и типы «отменяемости» потока.

Задание 2

В задании 2 рассматривается процесс создания асинхронно отменяемого потока (т.е. потока, который можно отменить в любой момент).

С консоли программа получает время задержки в секундах (по умолчанию – 1). Далее основной поток создаёт дочерний, засыпает на переданное количество секунд, просит дочерний поток прекратить свою работу и ожидает его завершения. После этого выводится сообщение о том, каким образом был завершен дочерний поток.

В дочернем потоке устанавливается тип отменяемости – асинхронный и бесконечно выводятся информационные сообщения о текущей итерации потока, которые производятся раз в секунду. Таким образом, на самом деле, поток может завершить вою работу только если будет прерван.

Задание 3

Задание 3 является точной копией задания 2, за исключением того факта, что теперь дочерний поток делается неотменяемым вовсе, и, для того, что бы он не зациклился на бесконечность, итерации он выводит строго определенное количество раз, передаваемое потоковой функции через аргументы, и равное удвоенному времени задержки.

Задание 4

Задание 4 так же является модификацией задания 2 с тем, что бы применить в нём синхронно отменяемый поток (т.е. поток, который можно отменить только в конкретных местах, точках выхода). Главный поток работает также, как и в предыдущих программах.

Дочерний поток устанавливает тип «отменяемости» - синхронный, и вычисляет приближение числа Пи по формуле Лейбница. При помощи функции pthread\_testcancel() устанавливаются места, в которых поток может быть прерван – до начала вычислений и после них. На каждой итерации выводится текущее приближение.

Задание 5

Задание 5 рассчитано на работу с потоковыми данными. В программе создаётся указанное количество потоков и ожидается завершение их работы. При этом каждый поток выводит в стандартный поток вывода случайное количество строк вида «This string is x for thread\_id, some text, random number: y», где thread\_id – идентификатор потока, х – номер текущей итерации, y – случайное число, генерирующее на всё том же диапазоне. При этом неизменяемая в цикле часть строки хранится в потоковых данных.

Потоковые данные подключаются при помощи ключа str\_key. С ним так же ассоциирована функция очистки, освобождающая память, которая была выделена под строку.

Пример работы программы (создаётся 3 потока, диапазон случайных чисел – от 1 до 5):

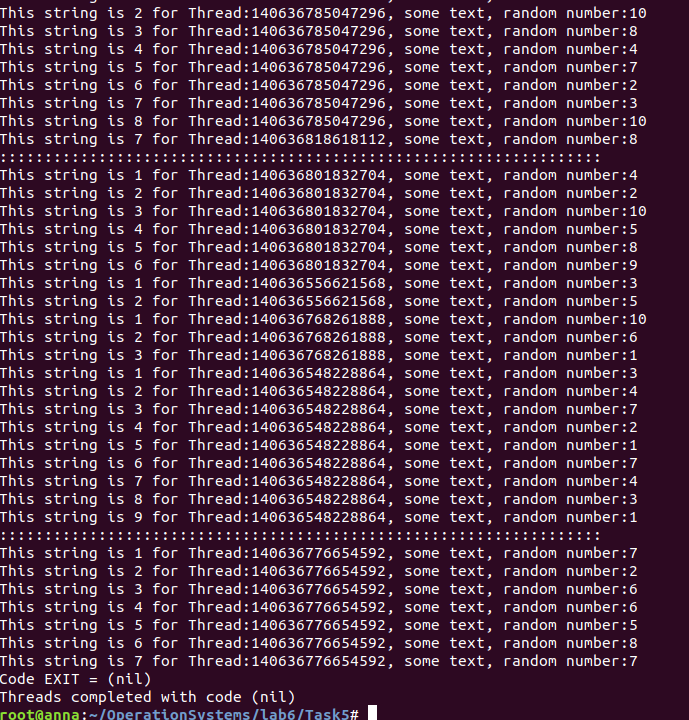
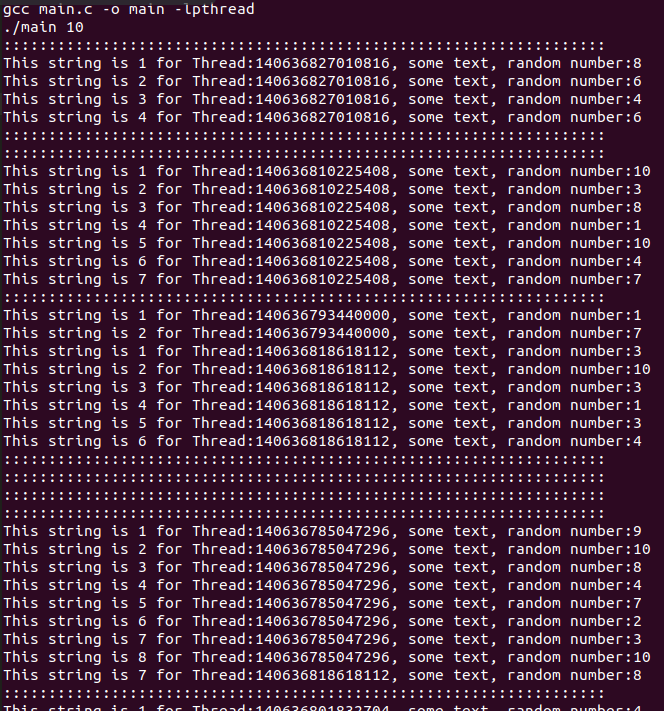


Рисунок 1 - Пример работы программы с использованием потоковых данных

Задание 6

Задание 6 заключается в перестройке задания 2 с использованием обработчиков очистки. Иными словами, с добавлением функции, которая будет вызываться автоматически по завершению потока.

К программе добавляется функция-обработчик завершения потока, которая показывает, на какой итерации поток завершил свою работу, когда произошла отмена потока.

В потоковой функции регистрируются обработчик очистки макросом pthread\_cleanup\_push(). Из-за особенностей реализации добавляем так же макрос pthread\_cleanup\_pop().

# ВЫВОДЫ

Таким образом, в ходе выполнения данной работы были рассмотрены возможности управления потоками, такие как атрибуты потоков, потоковые данные, обработчики очистки, а также различные способы отмены потока. Соответствующие исходные файлы, исполняемые файлы и скрипты сборки помещены в архив.