Домашнее задание 5

Многопоточность

Маринченко ИГорь БПИ207 вариант 33

2021

**Домашнее задание №5**

Многопоточное приложение.

1. **Описание задания:**

***И снова пляшущие человечки.*** Узнав о планах преступников озвученных в задаче 32, Шерлок Холмс предложил лондонской полиции специальную машину для дешифровки сообщений злоумышленников. Реализовать многопоточное приложение, дешифрующее кодированный текст. В качестве ключа используется известная кодовая таблица, устанавливающая однозначное соответствие между каждой буквой и каким-нибудь числом. Процессом узнавания кода в решении задачи пренебречь. Каждый поток дешифрует свои кусочки текста. При решении использовать парадигму портфеля задач.

1. **Структурная схема программы:**
   1. **Модель**

**Взаимодействующие равные** – модель, в которой исключен не занимающийся непосредственными вычислениями управляющий поток. Распределение работ в таком приложении либо фиксировано заранее, либо динамически определяется во время выполнения. Одним из распространенных способов динамического распределения работ является **«портфель задач»**. Портфель задач, как правило, реализуется с помощью разделяемой переменной, доступ к которой в один момент времени имеет только один процесс.

**Парадигма портфель задач**

Вычислительная задача делится на конечное число подзадач, являющихся независимыми единицами работы. Каждая подзадача может выполнить однотипные действия над разными данными. В данной работе подзадачей выступает дешифровка одного отдельного символа строки. Подзадачи нумеруются, т. е. каждому номеру соответствует свой набор данных, точнее над множеством номеров задач определяется функция, которая однозначно отражает номер задачи на соответствующий ему набор данных. Создается переменная для хранения номера задачи - номера символа, которую следует выполнять. Каждый поток независимо выполняет следующую цепочку действий: 1) обращается к портфелю задач для выяснения текущего номера задачи; 2) увеличивает портфель задач на единицу; 3) выполняет задачу, используя соответствующие данные; 4) обращается к портфелю задач для получения следующего номера задачи. Должен быть предусмотрен механизм остановки процессов при исчерпывании всего множества задач. Узкое место данной парадигмы - это момент получения очередной задачи и увеличения номера последующей задачи на единицу. Если между этими двумя действиями работа выполняющего их потока прервется, то некоторую задачу, возможно, отработают несколько потоков. Данные две операции должны считаться одной неделимой (атомарной). Для этого в данной работе был применён механизм синхронизации с помощью мьютекса.

* 1. **Классы**

Class Decrypter – класс для дешифровки заданной строки. Содержит внутри себя:

1. Таблицу соответствия кодов (двузначное число) и букв английского алфавита обоих регистров, а также пробельного символа.
2. Метод encrypt, возвращающий расшифрованную строку.
3. **Основные характеристики программы**

Размер исходных текстов – 7кб.

Число модулей – 2.

Размер исполняемого файла – 469кб.

**Формат входных данных:**

Вводится строка с зашифрованной текстом

**Формат выходных данных:**

Сначала выводится строка до расшифровки

**Пример:**

Encrypted string: 2501181510082023031114022205061716260424090712131921

**Затем каждый поток с уникальным номером, номер задания и результат его работы:**

Thread ID: 1, Task 0: 25 -> a

Thread ID: 2, Task 1: 01 -> b

Thread ID: 8, Task 8: 03 -> i

Thread ID: 4, Task 3: 15 -> d

Thread ID: 5, Task 4: 10 -> e

Thread ID: 1, Task 5: 08 -> f

Thread ID: 6, Task 6: 20 -> g

Thread ID: 19, Task 24: 19 -> y

Thread ID: 3, Task 2: 18 -> c

Thread ID: 2, Task 9: 11 -> j

Thread ID: 9, Task 10: 14 -> k

Thread ID: 10, Task 11: 02 -> l

Thread ID: 8, Task 12: 22 -> m

Thread ID: 11, Task 13: 05 -> n

Thread ID: 12, Task 14: 06 -> o

Thread ID: 4, Task 15: 17 -> p

Thread ID: 13, Task 16: 16 -> q

Thread ID: 14, Task 17: 26 -> r

Thread ID: 5, Task 18: 04 -> s

Thread ID: 15, Task 19: 24 -> t

Thread ID: 16, Task 20: 09 -> u

Thread ID: 1, Task 21: 07 -> v

Thread ID: 17, Task 22: 12 -> w

Thread ID: 18, Task 23: 13 -> x

Thread ID: 7, Task 7: 23 -> h

Thread ID: 6, Task 25: 21 -> z

**И в конце результат расшифровки:**

Decrypted string: abcdefghijklmnopqrstuvwxyz

**Тесты:**

|  |  |  |
| --- | --- | --- |
| Номер тестового набора | Входные данные | Время работы, миллисекунды |
| 1 | 24100424 | 0,004 |
| 2 | 43514943 | 0,004 |
| 3 | 3300120051002600430019 | 0,006 |
| 4 | 2310020206004210020206 | 0,006 |
| 5 | 2501181510082023031114022205061716260424090712131921 | 0.023 |
| 6 | сгенерированы программой | 0,005 |
| 7 | сгенерированы программой | 0,002 |

**Источники**

<https://studref.com/702392/informatika/paradigmy_parallelnyh_prilozheniy>

<staff.mmcs.sfedu.ru>