

ACS. OS. HW10

Горбачева Маргарита Валерьевна | БПИ-245

Задание:

Есть 100 параллельных источников информации (потоки), порождающих целые числа от 1 до 100, которые с разной скоростью поступают в общий буфер, достаточный для их хранения и обработки. Скорость определяется случайной задержкой от 1 до 7 секунд (функция `sleep`). Поток, отслеживающий поступление данных (работает быстро и без задержек), получив пару любых поступивших чисел, сразу же отправляет их на суммирование, создав для этого поток сумматор, который в течение случайного времени (от 3 до 6 секунд) осуществляет суммирование поступивших чисел и направляет их в тот же входной буфер для повторного использования (то есть, для суммирования с новыми поступившими числами или другими промежуточными результатами). То есть, одновременно может быть запущено несколько сумматоров. Вычисления завершаются, когда будет получен окончательный результат. Разработать программу, используя библиотеку POSIX threads, которая обеспечивает выполнение асинхронных параллельных вычислений по описанной схеме. Для понимания происходящих в системе событий и процессов необходимо сформировать вывод в консоль протокола, в котором подробно фиксируются выполняемые программой действия, описывающие продвижение данных, получение промежуточных результатов, состояние поступивших в буфер необработанных данных. Форма протокола свободная, но понятная для чтения. Еще одна ассоциация с приведенной схемой. Талоны в очередь берутся в любой момент, обслуживающих устройств (сумматоров) много. Очередь – буфер. По талону могут перенаправить другому обслуживающему устройству или можно взять еще один талон...

Решение: решение данного домашнего задания лежит в папке GitHub репозитория по ссылке: <https://github.com/Misss-Lacoste/ACS-OS/tree/main/hw10>

Nota Bene!! Прошу заметить, что комментарии в кодах написаны на английском языке, т.к. задание выполнялось на виртуальной машине линукса, которая не поддерживает русский язык.

Запуск программы: `gcc -o main main.c -lpthread` и после `./main`

Демонстрация работы программы:

1 прогон

```
hw10 > C main.c > main()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
margarita@margarita:~/workspace/qwerty$ cd /home/margarita/workspace/qwerty/homeworks/hw10
margarita@margarita:~/workspace/qwerty$ cd /home/margarita/workspace/qwerty/hw10
margarita@margarita:~/workspace/qwerty/hw10$ gcc -o main main.c -lpthread
margarita@margarita:~/workspace/qwerty/hw10$ ./main
Starting position:
Sources number is 100
The thread 2 has just generated a number: 48 with delay in 1 seconds.
Our buffer contains 1 elements now!
The thread 16 has just generated a number: 43 with delay in 1 seconds.
Our buffer contains 2 elements now!
Dispatcher tells that it has created the summator 0
The summator 0 took 48 from source 2 & 43 from source 16
Buffer tells that 0 elements are left
The thread 26 has just generated a number: 94 with delay in 1 seconds.
Our buffer contains 1 elements now!
The thread 34 has just generated a number: 75 with delay in 1 seconds.
Our buffer contains 2 elements now!
The thread 42 has just generated a number: 32 with delay in 1 seconds.
Our buffer contains 3 elements now!
The thread 92 has just generated a number: 64 with delay in 1 seconds.
Our buffer contains 4 elements now!
The thread 94 has just generated a number: 71 with delay in 1 seconds.
Our buffer contains 5 elements now!
The thread 95 has just generated a number: 23 with delay in 1 seconds.
Our buffer contains 6 elements now!
The thread 98 has just generated a number: 71 with delay in 1 seconds.
Our buffer contains 7 elements now!
Dispatcher tells that it has created the summator 1
The summator 1 took 94 from source 26 & 75 from source 34
Buffer tells that 5 elements are left
Dispatcher tells that it has created the summator 2
The summator 2 took 32 from source 42 & 64 from source 92
Buffer tells that 3 elements are left
Dispatcher tells that it has created the summator 3
The summator 3 took 71 from source 94 & 23 from source 95
Buffer tells that 1 elements are left
The thread 3 has just generated a number: 60 with delay in 2 seconds.
Our buffer contains 2 elements now!
Dispatcher tells that it has created the summator 4
The summator 4 took 71 from source 98 & 60 from source 3
Buffer tells that 0 elements are left
The thread 11 has just generated a number: 11 with delay in 2 seconds.
Our buffer contains 1 elements now!
The thread 9 has just generated a number: 26 with delay in 2 seconds.
```

```
margarita@margarita:~/workspace/qwerty/hw10$ ./main
Summator 12 calculates the sum as 52 + 26 = 78 (within 4 seconds)
Summator 12 took the result 78 to the buffer
Now there are 16 elements in a buffer
Dispatcher tells that it has created the summator 50
The summator 50 took 52 from summator 19 & 87 from source 30
Buffer tells that 14 elements are left
Summator 7 calculates the sum as 69 + 70 = 139 (within 5 seconds)
Summator 7 took the result 139 to the buffer
Now there are 15 elements in a buffer
Summator 13 calculates the sum as 5 + 22 = 27 (within 4 seconds)
Summator 13 took the result 27 to the buffer
Now there are 16 elements in a buffer
Dispatcher tells that it has created the summator 51
The summator 51 took 82 from source 37 & 73 from source 52
Buffer tells that 14 elements are left
Dispatcher tells that it has created the summator 52
The summator 52 took 56 from source 47 & 90 from source 51
Buffer tells that 12 elements are left
Dispatcher tells that it has created the summator 53
The summator 53 took 27 from source 54 & 8 from source 66
Buffer tells that 10 elements are left
Dispatcher tells that it has created the summator 54
The summator 54 took 64 from source 69 & 90 from source 82
Buffer tells that 8 elements are left
Dispatcher tells that it has created the summator 55
The summator 55 took 28 from source 90 & 65 from source 88
Buffer tells that 6 elements are left
Dispatcher tells that it has created the summator 56
The summator 56 took 60 from source 93 & 169 from summator 1
Buffer tells that 4 elements are left
Dispatcher tells that it has created the summator 57
The summator 57 took 32 from summator 21 & 78 from summator 12
Buffer tells that 2 elements are left
Dispatcher tells that it has created the summator 58
The summator 58 took 139 from summator 7 & 27 from summator 13
Buffer tells that 0 elements are left
Summator 27 calculates the sum as 14 + 9 = 23 (within 3 seconds)
Summator 27 took the result 23 to the buffer
Now there are 1 elements in a buffer
The result is 23
The final result is 23! :)
It has been generated 100 numbers!
margarita@margarita:~/workspace/qwerty/hw10$
```

2 прогон

```
margarita@margarita:~/workspace/qwerty/hw10$ ./main
Summator 30 calculates the sum as 79 + 85 = 164 (within 3 seconds)
Summator 30 took the result 164 to the buffer
Now there are 13 elements in a buffer
Dispatcher tells that it has created the summator 53
The summator 53 took 2 from source 79 & 27 from source 76
Buffer tells that 11 elements are left
Summator 22 calculates the sum as 80 + 78 = 158 (within 4 seconds)
Summator 22 took the result 158 to the buffer
Now there are 12 elements in a buffer
The summator 54 took 1 from source 80 & 105 from summator 10
Buffer tells that 10 elements are left
Dispatcher tells that it has created the summator 54
Dispatcher tells that it has created the summator 55
The summator 55 took 23 from source 90 & 23 from source 96
Buffer tells that 8 elements are left
Dispatcher tells that it has created the summator 56
The summator 56 took 18 from summator 25 & 95 from summator 12
Buffer tells that 6 elements are left
Dispatcher tells that it has created the summator 57
The summator 57 took 97 from summator 3 & 91 from summator 14
Buffer tells that 4 elements are left
Dispatcher tells that it has created the summator 58
The summator 58 took 145 from summator 28 & 89 from summator 20
Buffer tells that 2 elements are left
Dispatcher tells that it has created the summator 59
The summator 59 took 164 from summator 30 & 158 from summator 22
Buffer tells that 0 elements are left
Summator 32 calculates the sum as 43 + 88 = 131 (within 3 seconds)
Summator 32 took the result 131 to the buffer
Now there are 1 elements in a buffer
The result is 131
Summator 24 calculates the sum as 2 + 87 = 89 (within 4 seconds)
Summator 11 calculates the sum as 68 + 39 = 107 (within 6 seconds)
Summator 16 calculates the sum as 18 + 97 = 115 (within 5 seconds)
Summator 17 calculates the sum as 4 + 64 = 68 (within 5 seconds)
Summator 36 calculates the sum as 61 + 90 = 151 (within 3 seconds)
Summator 37 calculates the sum as 21 + 107 = 128 (within 3 seconds)
Summator 23 calculates the sum as 44 + 73 = 117 (within 5 seconds)
Summator 31 calculates the sum as 48 + 81 = 129 (within 4 seconds)
Summator 15 calculates the sum as 66 + 18 = 84 (within 6 seconds)
The final result is 131! :)
It has been generated 100 numbers!
margarita@margarita:~/workspace/qwerty/hw10$
```

Краткое резюме программки:

Имеем 100 параллельных источников информации (потоков), каждый генерирует целые числа от 1 до 100. Далее сгенерированные числа поступают в общий буфер. При появлении двух и более элементов диспетчер создаёт поток-сумматор, который извлекает два самых старых элемента из буфера (FIFO-очередь), вычисляет сумму этих чисел, затем добавляет ее обратно в буфер как новый элемент(но уже с авторскими правами сумматора). Процесс продолжается до тех пор, пока в буфере не останется один элемент – это и есть окончательный результат – сумма всех 100 исходных чисел. Потом программа успешно останавливается 😊

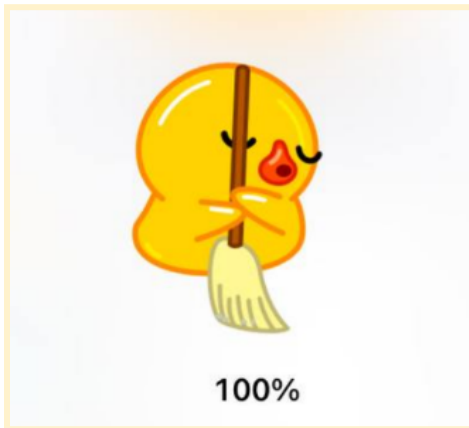
Архитектура программки:

1. Создаем массив для очереди элементов по правилам FIFO(снова “привет” алгоритмам), в нем есть сами числа, id источника или сумматора, флажок для отслеживания, who is who (true - источник, false - сумматор).
2. Что же касается потока-источника: он генерирует случайное число, затем отдыхает, снова генерирует случайное число и снова отдыхает, причем чисел у

нас 100, согласно заданию. После добавления числа и когда переменная `bufferCount` показывает, что элементов как раз стало больше или равно 2, оповещает об этом диспетчера, чтобы обеспечивать операции по сложению чисел и выводу этого всего на консоль.

3. А вот поток-сумматор ожидает, когда же ему скажут наконец, когда появится 2 и более чисел, чтобы начать свою миссию по сложению чисел.
4. Диспетчер у нас за всем следит. Работает в цикле, ожидает появления минимум двух чисел в буфере, после чего налаживает связь с ребятами сумматором и буфером. Также он создает новый поток-сумматор, в общем, обеспечивает бесперебойную работу.
5. В главной функции `main` (как раз самое ее начало) инициализируется генератор случайных чисел. Запускается босс ака диспетчер, далее идет запуск наших 100 потоков. Естественно, потом мы ждем завершения всех источников, пока программа не вычислит нам конечный результат суммы случайных чисел.

Обязательно очищаем память программы потом!



P.S. специально зашла почистить кэш в тг, чтобы вставить сюда фотографию уточки 😊

Отчет получился самым коротким в истории моих отчетов, но, к сожалению, декабрь и его стремящееся к $+\infty$ число дедлайнов не оставляют выбора :(