

Мерочкин Илья БПИ218

Индивидуальное домашнее задание по архитектуре вычислительных систем №4

Вариант 20

20. Задача о программистах. В отделе работают три программиста. Каждый программист пишет свою программу и отдает ее на проверку другому случайному программисту. Программист проверяет чужую программу, когда его собственная уже написана и передана на проверку. По завершении проверки, программист возвращает программу с результатом (формируемым случайно по любому из выбранных Вами законов): программа написана правильно или неправильно. Программист спит, если отправил свою программу и не проверяет чужую программу. Программист просыпается, когда получает заключение от другого программиста. Если программа признана правильной, программист пишет другую программу, если программа признана неправильной, программист исправляет ее и отправляет на проверку тому же программисту, который ее проверял. К исправлению своей программы он приступает, завершив проверку чужой программы. При наличии в очереди проверяемых программ и приходе заключения о неправильной своей программы программист может выбирать любую из возможных работ. Создать многопоточное приложение, моделирующее работу программистов.

4 балла

Программа лежит в файле `main.cpp`, компилируется командой `g++ main.cpp -o ./main`.

Немного о тонкостях реализации. Все три программиста пишут на особом языке, состоящем из 0 и 1, при этом программа считается корректной, если количество 0 в коде не превосходит количества 1 (код программы генерируется функцией `generateCode`). Когда программист отправляет свой код на проверку он ждет результата, и в процессе ожидания сам может делать код ревью программа, которые пришли к нему. Если он успешно написал программу - то переходит к написанию следующей. При этом, после успешного написания последней программы,

он ожидает, пока все остальные программисты также напишут свои последние программы, так как к нему могут прийти запросы на код ревью.

Входные данные: программа принимает на вход целое число - количество программ, которое должен написать каждый программист (в диапазоне от 1 до 100), а затем 3 строки - имена первого, второго и третьего программиста соответственно. Чтобы ввести данные с консоли, необходимо указать исполняемый файл и число 1, а затем вводить данные.

Пример ввода из консоли:

```
./main 1
```

```
4
```

```
Max
```

```
Tom
```

```
Sam
```

Описание модели параллельных вычислений. Поведение программистов реализовано в функции `writeCode`. Для каждого из трех программистов заводится отдельный поток. В аргументы функции передаются данные программиста - его имя и номер. Также для каждого программиста есть две очереди `codeForReview[num]` (в нее другие программисты записывают код, который необходимо обзреть, и свой номер) и `assert[num]` (в нее программист, который делал код ревью для текущего программиста записывает результат), `num` - номер программиста. К обеим очередям доступ на запись в один момент имеет только один поток (это осуществляется посредством мьютексов). Каждый программист пишет код, затем отправляет ее случайному программисту на код ревью и засыпает до того момента, пока у него не появится запросы на ревью кода или не придет результат его проверки. Если результат успешен, то он идет писать следующую программу. Когда он написал последнюю, то поток все равно не завершается, а ждет, пока все остальные программисты также допишут последнюю задачу (т.к. поток может завершиться, а после этого придет запрос на код ревью). В процессе выполнения программы выводятся данные о совершенных действиях (доступ к потоку вывода также ограничивается посредством семафора).

5 баллов

В программу добавлены подробные комментарии, поясняющие выполняемые действия и используемые переменные.

Описание сценария одновременного действия сущностей. Все три программиста могут одновременно начать писать код - тогда написание каждый из них будет отдельно его писать. Все три программиста могут одновременно захотеть записать элементы в очередь `codeForReview[i]` (например, двое захотят положить туда код для ревью, и третий достать, чтобы сделать ревью). В таком случае сделано это будет по очереди (ограничено мьютексом). Кроме того, в процессе выполнения проверки и написания кода, из каждого потока выводится некоторая информация. Несколько потоков могут одновременно выводить информацию и тогда данные могут быть спутаны, поэтому это ограничено и вывод также осуществляется по очереди.

6 баллов

Реализован ввод из командной строки. Для этого необходимо указать исполняемый файл а затем данные (число программ и три имени). Например: `./main 4 Bob Tom Sam`.

Обобщенный алгоритм. Поведение программистов реализовано в функции `writeCode`. Для каждого программиста заводим отдельный поток, а в аргументы функции передаем класс `ProgrammerInfo`, который хранит номер программиста и его имя. Далее начинаем писать программы - программа генерируется в функции `genrateCode` с задержкой в 8 секунд для имитации реальной работы программистов. После написания кода программист случайным образом выбирает другого, который будет делать код ревью и отдает ему код. Далее программист засыпает до того момента, пока: ему не придет результат проверки кода (если результат успешен, он начинает писать новую программу, иначе исправляет код (с задержкой), отдает обратно на проверку и снова засыпает) или пока другой программист не даст ему код на проверку, тогда он проверяет код (также с задержкой), возвращает результат этому программисту и засыпает. Если программист успешно написал все программы, то он засыпает в ожидании, пока все остальные программисты не напишут последнюю программу (после этого прекращается работа потока), либо ему не придет код на проверку (он проверяет и засыпает обратно).

7 баллов

Добавлен ввод/вывод данных из файла и в файл. Для этого после указания исполняемого файла необходимо указать файл с входными данными и выходной файл. Например, `./main input.txt output.txt`. При этом, результаты работы программы выводятся и в файл, и на экран.

Файлы `input_test1.txt` `output_test1.txt` и `input_test2.txt` `output_test2.txt` - пример входных и выходных файлов. При многократных запусках результат работы отличался, однако всегда удовлетворял поставленному условию.

8 баллов

В программу добавлена генерация случайных данных в допустимых диапазонах (правая граница диапазона уменьшена и данные генерируются от 1 до 10 в целях более быстрой проверки). Также для удобства сгенерированные данные выводятся на экран.

Примеры запуска программы с генерацией входных данных:

Пример 1

```
ilya@ilya-VirtualBox:~/Ilya/AVS/ASM/hm_4$ ./main
Number of programmms - 3
First programmer - Jerry
First programmer - Andrew
First programmer - Alex

Programmer Jerry wrote 1 code
Programmer Andrew wrote 1 code
Programmer Alex wrote 1 code
Programmer Alex made code review for Jerry with result: Accept
Programmer Jerry made code review for Alex with result: Reject
Programmer Alex made code review for Andrew with result: Accept
Programmer Alex corrected his code
Programmer Jerry wrote 2 code
Programmer Andrew wrote 2 code
Programmer Jerry made code review for Alex with result: Accept
Programmer Alex made code review for Andrew with result: Reject
Programmer Andrew made code review for Jerry with result: Reject
Programmer Jerry corrected his code
Programmer Andrew corrected his code
Programmer Alex wrote 2 code
Programmer Andrew made code review for Jerry with result: Accept
Programmer Alex made code review for Andrew with result: Accept
Programmer Andrew made code review for Alex with result: Accept
Programmer Jerry wrote 3 code
Programmer Alex wrote 3 code
Programmer Andrew wrote 3 code
Programmer Jerry made code review for Alex with result: Accept
Programmer Alex made code review for Jerry with result: Reject
Programmer Alex made code review for Andrew with result: Accept
Programmer Jerry corrected his code
Programmer Alex made code review for Jerry with result: Accept
```

Пример 2:

```
Ilya@Ilya-VirtualBox:~/Ilya/AVS/ASM/hm_4$ ./main
Number of programmms - 8
First programmer - Andrew
First programmer - Tom
First programmer - Steve

Programmer Tom wrote 1 code
Programmer Andrew wrote 1 code
Programmer Steve wrote 1 code
Programmer Steve made code review for Andrew with result: Reject
Programmer Tom made code review for Steve with result: Accept
Programmer Andrew made code review for Tom with result: Accept
Programmer Andrew corrected his code
Programmer Steve wrote 2 code
Programmer Tom wrote 2 code
Programmer Steve made code review for Andrew with result: Accept
Programmer Andrew made code review for Steve with result: Reject
Programmer Steve made code review for Tom with result: Reject
Programmer Tom corrected his code
Programmer Steve corrected his code
Programmer Andrew wrote 2 code
Programmer Steve made code review for Tom with result: Accept
Programmer Andrew made code review for Steve with result: Accept
Programmer Steve made code review for Andrew with result: Reject
Programmer Tom wrote 3 code
Programmer Andrew corrected his code
Programmer Steve wrote 3 code
Programmer Andrew made code review for Tom with result: Accept
Programmer Tom made code review for Steve with result: Accept
Programmer Steve made code review for Andrew with result: Accept
Programmer Steve wrote 4 code
Programmer Tom wrote 4 code
Programmer Andrew wrote 3 code
Programmer Tom made code review for Steve with result: Reject
Programmer Andrew made code review for Tom with result: Reject
Programmer Tom made code review for Andrew with result: Accept
Programmer Steve corrected his code
Programmer Tom corrected his code
Programmer Andrew wrote 4 code
Programmer Tom made code review for Steve with result: Accept
Programmer Tom made code review for Andrew with result: Accept
Programmer Andrew made code review for Tom with result: Accept
Programmer Steve wrote 5 code
Programmer Andrew wrote 5 code
Programmer Tom wrote 5 code
Programmer Tom made code review for Steve with result: Accept
Programmer Steve made code review for Tom with result: Accept
Programmer Steve made code review for Tom with result: Accept
Programmer Tom made code review for Andrew with result: Accept
Programmer Steve wrote 6 code
Programmer Andrew wrote 6 code
Programmer Tom wrote 6 code
Programmer Tom made code review for Andrew with result: Accept
Programmer Andrew made code review for Steve with result: Reject
Programmer Steve corrected his code
Programmer Andrew wrote 7 code
Programmer Andrew made code review for Tom with result: Reject
Programmer Tom made code review for Andrew with result: Reject
Programmer Andrew made code review for Steve with result: Accept
Programmer Tom corrected his code
Programmer Andrew corrected his code
Programmer Steve wrote 7 code
Programmer Andrew made code review for Tom with result: Accept
Programmer Tom made code review for Andrew with result: Accept
Programmer Tom wrote 7 code
Programmer Andrew wrote 8 code
Programmer Tom made code review for Steve with result: Accept
Programmer Steve made code review for Tom with result: Accept
Programmer Steve wrote 8 code
Programmer Tom wrote 8 code
Programmer Steve made code review for Andrew with result: Reject
Programmer Tom made code review for Steve with result: Reject
Programmer Steve made code review for Tom with result: Accept
Programmer Andrew corrected his code
Programmer Steve corrected his code
Programmer Steve made code review for Andrew with result: Accept
Programmer Tom made code review for Steve with result: Accept
```

9 баллов (1 вариант)

Попробуем отключить различные синхропримитивы, и посмотрим, как это повлияет на нашу программу.

- 1) Давайте уберем семафор, отвечающий за синхронизацию вывод (outSem), немного изменим программу (файл main_withoutSync.cpp), и посмотрим, как это повлияет на работу.

После нескольких запусков удалось достичь следующей ситуации:

```
ilya@ilya-VirtualBox:~/Ilya/AVS/ASH/hm_4/mark_9$ ./main
Number of programs - 8
First programmer - Alex
First programmer - Steve
First programmer - Ilya

Programmer Programmer AlexIlya wrote wrote 1 code
1 code
Programmer Steve wrote 1 code
Programmer Ilya made code review for Alex with result: Accept
Programmer Alex made code review for Ilya with result: Accept
Programmer Ilya wrote 2 code
Programmer Alex wrote 2 code
Programmer Ilya made code review for Alex with result: Reject
Programmer Alex made code review for Steve with result: Reject
Programmer Alex corrected his code
Programmer Steve corrected his code
Programmer Ilya made code review for Alex with result: Accept
Programmer Alex made code review for Ilya with result: Reject
Programmer Ilya corrected his code
Programmer Alex wrote 3 code
Programmer Steve made code review for Alex with result: Reject
Programmer Alex made code review for Steve with result: Accept
Programmer Steve wrote 2 code
Programmer Alex corrected his code
Programmer Alex made code review for Ilya with result: Accept
Programmer Ilya made code review for Steve with result: Accept
Programmer Steve made code review for Alex with result: Accept
Programmer Programmer Steve wrote 3 code
Alex wrote 4 code
Ilya wrote 3 code
Programmer Alex made code review for Steve with result: Accept
Programmer Programmer IlyaSteve made code review for made code review for Ilya with result: AlexReject
with result: Accept
Programmer Alex wrote 5 code
Programmer Steve wrote 4 code
Programmer Ilya corrected his code
Programmer Steve made code review for Ilya with result: Accept
Programmer Ilya made code review for Alex with result: Reject
Programmer Ilya wrote 4 code
Programmer Alex corrected his code
Programmer Steve made code review for Ilya with result: Accept
Programmer Ilya made code review for Steve with result: Accept
Programmer Ilya wrote Programmer 5 code
Steve wrote 5 code
Programmer Steve made code review for Ilya with result: Reject
Programmer Ilya made code review for Alex with result: Accept
Programmer Ilya corrected his code
Programmer Alex made code review for Steve with result: Accept
Programmer Steve made code review for Ilya with result: Accept

Programmer Alex wrote 6 code
Programmer Ilya wrote 6 code
Programmer Steve wrote 6 code
Programmer Programmer Steve made code review for Ilya with result: Accept
Ilya made code review for Alex with result: Accept
Programmer Alex made code review for Steve with result: Accept
Programmer Programmer AlexSteve wrote wrote 7 code
7 code
Programmer Alex made code review for Steve with result: Accept
Programmer Ilya wrote 7 code
Programmer Programmer Alex made code review for IlyaIlya made code review for with result: AlexAccept
with result: Reject
Programmer Steve wrote 8 code
Programmer Ilya wrote 8 code
Programmer Alex corrected his code
Programmer Ilya made code review for Steve with result: Reject
Programmer Alex made code review for Ilya with result: Reject
Programmer Ilya made code review for Alex with result: Accept
Programmer Steve corrected his code
Programmer Ilya corrected his code
Programmer Alex wrote 8 code
Programmer Programmer Alex made code review for Ilya made code review for Steve with result: Accept
Ilya with result: Accept
Programmer Ilya made code review for Alex with result: Accept
```

То есть наш вывод сильно съехал, так как несколько потоков одновременно выводили данные, из-за этого достаточно сложно понять, что произошло в программе.

- 2) Попробуем убрать мьютексы `reviewMutex`, который ограничивает доступ к очереди `codeForReview` (файл `main_withoutSync.cpp`). Однако, из-за того, что в один момент в очередь могут записывать только два потока, и это маловероятно, программа при многократных запусках ни разу не дала сбой. Однако при увеличении числа потоков такая проблема могла бы возникнуть.

10 баллов (1 вариант)

В моей изначальной программе уже используется два вида синхропримитива (и мьютексы, и семафоры). Поэтому, чтобы выполнить условие, заменю мьютексы в программе на семафоры и наоборот.

Файл с кодом `main_anotherSync.cpp`. Программа также корректно работает, тестовые файлы (`input_test` `output_test`) также приведены, и, в целом, особых различий в работе с первой программой не обнаружено.