Отчет
о работе многопоточной программы, суммирующей числа массива
(на 10 баллов)

Выполнила
студентка 2 курса ПИ БПИ228
Гакал Анжелика Сергеевна

Методы, реализующий функционал писателей

```
38  void write_data(int data) {
39      sem_wait(&empty);
40      buf[rear] = data;
41      rear = (rear + 1) % bufSize;
42      sem_post(&full);
43  }
44
45  void *producer(void *param) {
46      int producer_thread_num = *((int *) param);
47      int data = get_random_int(1, 20);
48
49      sleep(get_random_int(1, 7));
50
51      pthread_mutex_lock(&mutex_write);
52
53      pthread_mutex_lock(&mutex_output);
54      printf("Producer %d: Writes a new value = %d to cell [%d]\n", producer_thread_num, data, rear);
55      pthread_mutex_unlock(&mutex_output);
56
57      write_data(data);
58
59      pthread_mutex_unlock(&mutex_write);
60
61      return nullptr;
62  }
```

Метод, реализующий функционал сумматора

```
63  void *adder(void *param) {
64      Adder adder_obj = *(Adder *) (param);
65      int adder_num = adder_obj.adder_num;
66      int sum = adder_obj.number1 + adder_obj.number2;
67      sleep(get_random_int(3, 6));
68
69      pthread_mutex_lock(&mutex_write);
70
71      pthread_mutex_lock(&mutex_output);
72      printf("Adder %d: Calculate sum = %d to cell [%d]\n", adder_num, sum, rear);
73      pthread_mutex_unlock(&mutex_output);
74
75      write_data(sum);
76
77      pthread_mutex_unlock(&mutex_write);
78
79      return nullptr;
80  }
```

Метод, реализующий функционал читателя

```
82  void *consumer(void *param) {
83      int adder_cnt = 0;
84      for (int i = 0; i < 19; ++i) {
85          sem_wait(&full);
86          sem_wait(&full);
87
88          int number1 = buf[front];
89          front = (front + 1) % bufSize;
90
91          pthread_mutex_lock(&mutex_output);
92          printf("Reader reads value = %d from cell [%d]\n", number1, front - 1 < 0 ? bufSize - 1 : front - 1);
93          pthread_mutex_unlock(&mutex_output);
94
95          int number2 = buf[front];
96          front = (front + 1) % bufSize;
97
98          pthread_mutex_lock(&mutex_output);
99          printf("Reader reads value = %d from cell [%d]\n", number2, front - 1 < 0 ? bufSize - 1 : front - 1);
100         pthread_mutex_unlock(&mutex_output);
101
102         Adder *adder_obj = new Adder{adder_cnt, number1, number2};
103         pthread_create(&adder_threads.emplace_back(), nullptr, adder, (void *) (adder_obj));
104         adder_cnt++;
105
106         sem_post(&empty);
107         sem_post(&empty);
108     }
109     for (auto &x: adder_threads)
110         pthread_join(x, nullptr);
111
112     printf("\nSum = %d", buf[rear]);
113
114     return nullptr;
115 }
```

Метод main

```
118 int main() {
119     srand(seed);
120     pthread_mutex_init(&mutex_write, nullptr);
121
122     pthread_mutex_init(&mutex_output, nullptr);
123
124     sem_init(&empty, 0, bufSize);
125     sem_init(&full, 0, 0);
126
127     pthread_t reader;
128     pthread_create(&reader, nullptr, consumer, nullptr);
129
130     pthread_t threadP[20];
131     int producers[20];
132     for (int i = 0; i < 20; i++) {
133         producers[i] = i + 1;
134         pthread_create(&threadP[i], nullptr, producer, (void *) (producers + i));
135     }
136
137     for (auto x: threadP)
138         pthread_join(x, nullptr);
139
140     pthread_join(reader, nullptr);
141
142     std::cout << '\n'
143               << rear - 1 << '\n';
144     return 0;
145 }
```

В реализации решения задачи используются два мьютекса mutex_write и mutex_output. Первый мьютекс регулирует корректную запись в массив в методах adder и producer (чтобы 20 работающих потоком-писателей и поток-сумматор не перезаписывали и не перезатерали актуальные элементы). mutex_output регулирует корректность вывода в консоль (чтобы одно выводимое сообщение не перебивалось другим).

Так же используются 2 семафора empty и full, чтобы поток читателя мог начать считывать 2 числа только тогда, когда они уже были записаны в массив.

Пример работы программы:

```
Producer 9: Writes a new value = 12 to cell [0]
Producer 17: Writes a new value = 9 to cell [1]
Producer 1: Writes a new value = 3 to cell [2]
Reader reads value = 12 from cell [0]
Reader reads value = 9 from cell [1]
Producer 20: Writes a new value = 7 to cell [3]
Producer 13: Writes a new value = 20 to cell [4]
Reader reads value = 3 from cell [2]
Reader reads value = 7 from cell [3]
Producer 16: Writes a new value = 2 to cell [5]
Reader reads value = 20 from cell [4]
Reader reads value = 2 from cell [5]
Producer 4: Writes a new value = 13 to cell [6]
Producer 18: Writes a new value = 2 to cell [7]
Producer 15: Writes a new value = 20 to cell [8]
Reader reads value = 13 from cell [6]
Reader reads value = 2 from cell [7]
Producer 11: Writes a new value = 16 to cell [9]
Reader reads value = 20 from cell [8]
Reader reads value = 16 from cell [9]
Producer 14: Writes a new value = 8 to cell [10]
Producer 6: Writes a new value = 11 to cell [11]
Reader reads value = 8 from cell [10]
Reader reads value = 11 from cell [11]
Producer 7: Writes a new value = 11 to cell [12]
Producer 2: Writes a new value = 4 to cell [13]
Producer 5: Writes a new value = 1 to cell [14]
Reader reads value = 11 from cell [12]
Reader reads value = 4 from cell [13]
Producer 8: Writes a new value = 15 to cell [15]
Reader reads value = 1 from cell [14]
Reader reads value = 15 from cell [15]
Producer 10: Writes a new value = 2 to cell [16]
Adder 3: Calculate sum = 15 to cell [17]
Reader reads value = 2 from cell [16]
Reader reads value = 15 from cell [17]
Adder 0: Calculate sum = 21 to cell [18]
Producer 19: Writes a new value = 13 to cell [19]
Reader reads value = 21 from cell [18]
Reader reads value = 13 from cell [19]
Producer 3: Writes a new value = 2 to cell [0]
Producer 12: Writes a new value = 19 to cell [1]
```

```
Reader reads value = 2 from cell [0]
Reader reads value = 19 from cell [1]
Adder 1: Calculate sum = 10 to cell [2]
Adder 2: Calculate sum = 22 to cell [3]
Reader reads value = 10 from cell [2]
Reader reads value = 22 from cell [3]
Adder 6: Calculate sum = 15 to cell [4]
Adder 7: Calculate sum = 16 to cell [5]
Reader reads value = 15 from cell [4]
Reader reads value = 16 from cell [5]
Adder 9: Calculate sum = 34 to cell [6]
Adder 4: Calculate sum = 36 to cell [7]
Adder 5: Calculate sum = 19 to cell [8]
Reader reads value = 34 from cell [6]
Reader reads value = 36 from cell [7]
Adder 8: Calculate sum = 17 to cell [9]
Reader reads value = 19 from cell [8]
Reader reads value = 17 from cell [9]
Adder 10: Calculate sum = 21 to cell [10]
Adder 11: Calculate sum = 32 to cell [11]
Reader reads value = 21 from cell [10]
Reader reads value = 32 from cell [11]
Adder 14: Calculate sum = 36 to cell [12]
Adder 12: Calculate sum = 31 to cell [13]
Reader reads value = 36 from cell [12]
Reader reads value = 31 from cell [13]
Adder 13: Calculate sum = 70 to cell [14]
Adder 15: Calculate sum = 53 to cell [15]
Reader reads value = 70 from cell [14]
Reader reads value = 53 from cell [15]
Adder 17: Calculate sum = 123 to cell [16]
Adder 16: Calculate sum = 67 to cell [17]
Reader reads value = 123 from cell [16]
Reader reads value = 67 from cell [17]
Adder 18: Calculate sum = 190 to cell [18]

Sum = 190
Process finished with exit code 0
```