Национальный исследовательский университет "Высшая школа экономики".
Факультет компьютерных наук.

Пояснительная записка к домашнему заданию № 4 по предмету "Архитектура вычислительных систем".

Выполнила
студентка 2 курса группы БПИ192
Цимбалистая Ульяна Игоревна

17 ноября 2020 г.

Содержание

1. Текст задания

Вариант 26.
Вторая задача об Острове Сокровищ. Шайка пиратов под предводительством Джона
Сильвера высадилась на берег Острова Сокровищ. Несмотря на добытую карту старого Флинта, местоположение сокровищ по-прежнему остается загадкой, поэтому искать клад приходится практически на ощупь. Так как Сильвер ходит на деревянной ноге, то самому бродить по джунглям ему не с руки. Джон Сильвер поделил остров на участки, а пиратов на небольшие группы. Каждой группе поручается искать клад на нескольких участках, а сам Сильвер ждет на берегу. Группа пиратов, обшарив одну часть острова, переходит к другой, еще необследованной части. Закончив поиски, пираты возвращаются к Сильверу и докладывают о результатах. Требуется создать многопоточное приложение с управляющим потоком, моделирующее действия Сильвера и пиратов. При решении использовать парадигму портфеля задач.

2. Описание программы

Остров разбивается на 15 участков, каждый из которых также разбивается на 15 частей, в которых может быть спрятан клад. Для этого формируется двумерный массив типа bool размером 15 на 15, где на случайной позиции стоит true — это и есть клад.
В консоли пользователь задаёт количество групп пиратов (потоков). Количество возможных потоков должно быть не меньше 1 и не больше количества участков. Момент входа в функцию поиска отмечет директивой *#pragma omp parallel*, которая создает заданное пользователем число потоков. Каждый поток начинает выполнение функции, начиная с участка равного номеру потока.
В методе searching происходит поиск по переданной строке массива, и если клад находится, то программа завершится. Если же в переданной строке нет элемента true, то группа пиратов (поток) снова переходит в этот метод, но со строкой равной (номер строки + количество групп).

3. Текст программы

```
1.  #include <iostream>
2.  #include <omp.h>
3.
4.  using namespace std;
5.
6.  // Количество групп пиратов.
7.  int groups;
8.  // Количество участков.
9.  const int DIM = 15;
10. // Массив, представляющий собой остров, где количество строк - участки,
11. // а столбцы - места, где могут лежать сокровища.
12. bool place[DIM][DIM];
13. // Флаг, сигнализирующий о нахождении клада.
14. bool flag = false;
15.
16. // Координаты сокровищ.
17. int xPos;
18. int yPos;
19.
20. // Заполнение массива, в случайную ячейку помещяется true (клад).
21. void initPlaces() {
22.         int dim1 = rand() % 15;
23.         int dim2 = rand() % 15;
24.
25.         for (int i = 0; i < DIM; ++i) {
26.                 for (int j = 0; j < DIM; ++j) {
27.                         place[i][j] = false;
28.                 }
29.         }
30.
31.         place[dim1][dim2] = true;
32. }
33.
34. void search(int cell) {
35.         if (!flag) {
36.                 bool done = false;
37.                 for (int i = 0; i < DIM; ++i) {
38.                         if (place[cell][i]) {
39.                                 flag = true;
40.                                 done = true;
41.                                 xPos = cell;
42.                                 yPos = i;
43.
44.                                 cout << "Group of pirates # " << omp_get_thread_num() + 1
                                    << " found treasure at the X: " << xPos << " Y: " << yPos
                                    << "\n" << endl;
45.
46.                                 exit(0);
47.                         }
48.                 }
49.
50.                 cell = cell + groups;
51.                 // Если клад еще не найден и обследованы не все участки,
52.                 // то пираты переходят к другому участку.
53.                 if (!done && cell < DIM) {
54.                         search(cell);
55.                 }
56.         }
57. }
58.
59. int getNumber() {
60.         while (true) {
61.                 cout << "Number of groups of pirates in John Silver's gang:" << endl;
62.
63.                 int n;
```

```
64.                 cin >> n;
65.
66.                 if (cin.fail() || n < 1 || n > DIM) {
67.                         cin.clear();
68.                         cin.ignore(32767, '\n');
69.                 } else
70.                         return n;
71.         }
72. }
73.
74. int main() {
75.         srand(time(0));
76.
77.         cout << "The pirates, led by John Silver, have landed on Treasure Island.\n"
78.         "Leader of the pirate gang sent groups of pirates to find treasure deep to
                    jungles of island.\n\n";
79.
80.         groups = getNumber();
81.         initPlaces();
82.
83.         // Распараллеливает выполнение на потоки.
84.  #pragma omp parallel num_threads(groups)
85. {
86.         auto numb = omp_get_thread_num();
87.         search(numb);
88.
89. }
90.
91. return 0;
92. }
```

4. Примеры выполнения

```
/Users/ulyanatsimbalistaya/CLionProjects/homeworks/hw_4/cmake-build-debug/hw_4
The pirates, led by John Silver, have landed on Treasure Island.
Leader of the pirate gang sent groups of pirates to find treasure deep to jungles of island.

Number of groups of pirates in John Silver's gang:
5
Group of pirates # 2 found treasure at the X: 6 Y: 8


Process finished with exit code 0
```

```
/Users/ulyanatsimbalistaya/CLionProjects/homeworks/hw_4/cmake-build-debug/hw_4
The pirates, led by John Silver, have landed on Treasure Island.
Leader of the pirate gang sent groups of pirates to find treasure deep to jungles of island.

Number of groups of pirates in John Silver's gang:
7
Group of pirates # 2 found treasure at the X: 1 Y: 11


Process finished with exit code 0
```

```
/Users/ulyanatsimbalistaya/CLionProjects/homeworks/hw_4/cmake-build-debug/hw_4
The pirates, led by John Silver, have landed on Treasure Island.
Leader of the pirate gang sent groups of pirates to find treasure deep to jungles of island.

Number of groups of pirates in John Silver's gang:
4
Group of pirates # 2 found treasure at the X: 5 Y: 9


Process finished with exit code 0
```