**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»**

**(БГТУ им. В.Г.ШУХОВА)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

дисциплина: **Объектно - ориентированное программирование**

Лабораторная работа № 6

тема: «Потоки в C++»

Выполнил: ст. группы ВТ-22

Воскобойников Илья

Проверил: ст. пр.

Буханов Дмитрий Геннадьевич

Белгород 2020

**Цель работы:** изучение основных возможностей потоков управления и потоков ввода-вывода. Получение навыков работы со стандартными средствами управления потоками в С++11. Знакомство с классом Thread и стандартными средствами синхронизации потоков. .

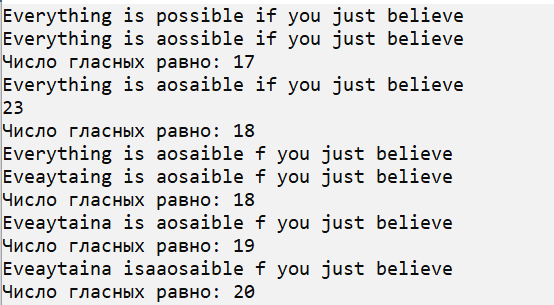
Один поток удаляет пробелы в строке и вставляет их в случайное место, а другой поток выполняет циклический сдвиг текста. Произвести синхронный вывод при каждой итерации. Показать выполнение работы программы в синхронном и асинхронном режимах.

Один поток выполняет подсчет количества гласных букв в тексте, адругой вставляет или удаляет случайным образом гласную букву.Произвести синхронный вывод при каждой итерации. Показатьвыполнение работы программы в синхронном и асинхронном режимах.

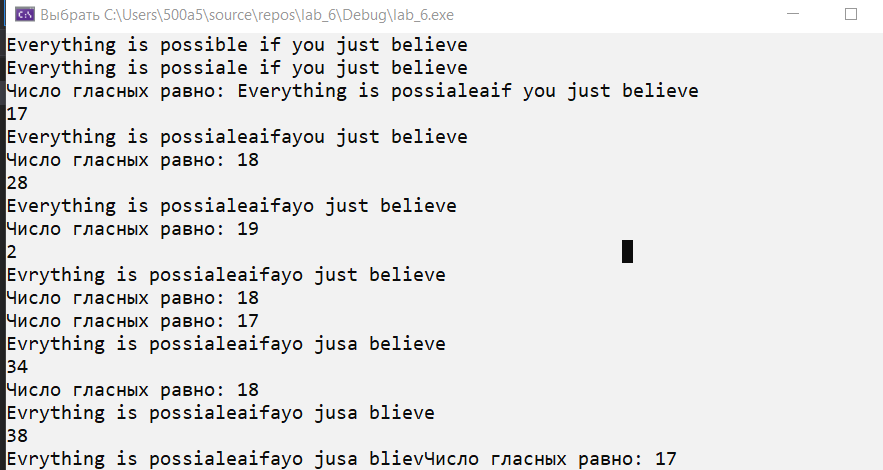
Задание 2

1. #include <mutex>
2. #include <thread>
3. #include <iostream>
4. #include <chrono>
5. #include <cstdlib>
6. #include <string>
7. #include <ctime>
8. #define MAXSIZE 255
9. using namespace std;
10. class Body {
11. protected:
12. bool search\_glas(char);
13. void del\_now(char[], int);
14. };
15. bool Body::search\_glas(char c)
16. {
17. int i = 0;
18. switch (c)
19. {
20. case 'a':
21. case 'e':
22. case 'o':
23. case 'u':
24. case 'i':
25. case 'y':
26. return true;
27. default:
28. return false;
29. };
30. }
31. class Num\_nowel : public Body { //подсчет числа гласных в тексте
32. public:
33. void operator()(char text[], mutex & m)
34. {
35. int j = 0;
36. for (int i = 0; i < 100; i++)
37. {
38. int count = 0;
39. m.lock();
40. for (int j = 0; text[j] != '\0'; j++)
41. {
42. if (search\_glas(text[j]))
43. count++;
44. }
45. m.unlock();
46. this\_thread::sleep\_for(chrono::seconds(1));
47. cout << "Число гласных равно: " << count << endl;
48. }
49. }
50. };
51. void Body::del\_now(char text[], int j)
52. {
53. int i;
54. for (i = j; text[i] != '\0'; i++)
55. text[i] = text[i + 1];
56. text[i] = '\0';
57. }
58. class Del\_add\_vowel : public Body {
59. public:
60. void operator()(char text[], mutex & m) {
61. for (int i = 0; i < 100; i++)
62. {
63. m.lock();
64. srand(unsigned(time(NULL)));
65. int j = 0;
66. if (strlen(text) != 0)
67. {
68. j = int(rand() % strlen(text));
69. if (search\_glas(text[j]))
70. {
71. cout << j << endl;
72. del\_now(text, j);
73. }
74. else
75. {
76. text[j] = 'a';
77. }
78. m.unlock();
79. this\_thread::sleep\_for(chrono::seconds(1));
80. }
81. else
82. {
83. cout << "Все символы закончились" << endl;
84. break;
85. }
86. }
87. }
88. };
89. void show(char text[], mutex & m) {
90. for (int i = 0; i < 100; i++)
91. {
92. m.lock();
93. cout << text << endl;
94. m.unlock();
95. this\_thread::sleep\_for(chrono::seconds(1));
96. }
97. }
98. int main()
99. {
100. setlocale(LC\_ALL, "rus");
101. mutex m;
102. Del\_add\_vowel del\_add;
103. Num\_nowel num;
104. char text[255] = "Everything is possible if you just believe";
105. cout << text << endl;
106. thread threadDel\_add(del\_add, text, ref(m));
107. thread threadNum\_now(num, text, ref(m));
108. thread threadShow(show, text, ref(m));
109. threadNum\_now.detach();
110. threadDel\_add.detach();
111. threadShow.join();
112. cout << "\n\n";
113. return 0;
114. }

Результаты выполнения программы(в асинхронном режиме)



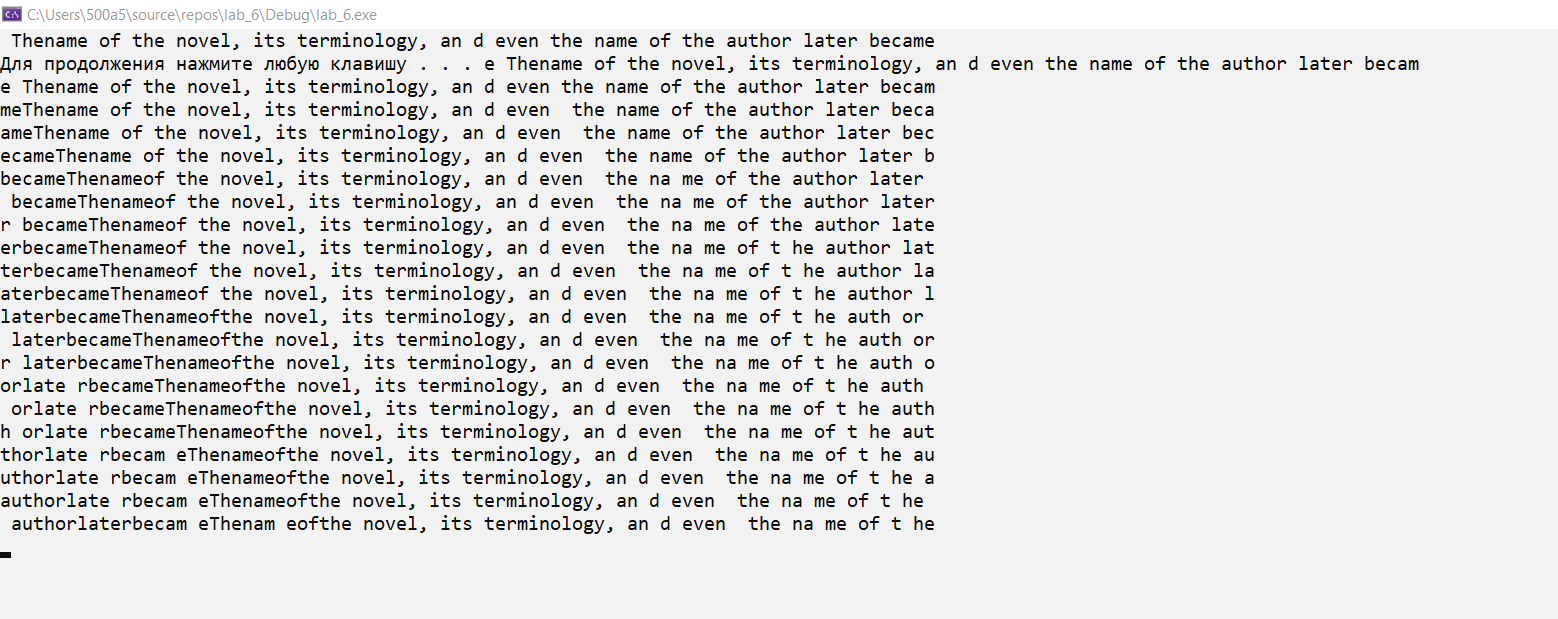
Результат выполнения программы (в синхронном режиме)



Задние 3

1. #include <Windows.h>
2. #include <iostream>
3. #include <cstdlib>
4. #include <string>
5. #include <ctime>
6. #include <tchar.h>
7. #define MAXSIZE 255
8. using namespace std;
9. char text[MAXSIZE] = "The name of the novel, its terminology, and even the name of the author later became ";
10. DWORD WINAPI thread\_1(LPVOID lpParam)
11. {
12. int i = 0;
13. for (int i = 0; i < 100; i++)
14. {
15. HANDLE mutex = OpenMutex(SYNCHRONIZE, FALSE, \_T("mutex"));
16. int pos = strlen(text);
17. for (int j = pos - 1; j != -1; j--)
18. {
19. text[j + 1] = text[j];
20. }
21. text[0] = ' ';
22. text[pos + 2] = '\0';
23. i++;
24. cout << text << endl;
25. ReleaseMutex(mutex);
26. }
27. return 0;
28. }
29. DWORD WINAPI thread\_2(LPVOID lpParam)
30. {
31. int i = 0;
32. for (int i = 0; i < 100; i++)
33. {
34. HANDLE mutex = OpenMutex(SYNCHRONIZE, FALSE, \_T("mutex"));
35. srand(unsigned(time(NULL)));
36. int j = 0;
37. if (strlen(text) != 0)
38. {
39. j = int(rand() % strlen(text));
40. if (text[j] == ' ')
41. {
42. int i;
43. for (i = j; text[i] != '\0'; i++)
44. text[i] = text[i + 1];
45. text[i] = '\0';
46. j = int(rand() % strlen(text));
47. int pos = strlen(text);
48. for (int k = pos - 1; k != j - 1; k--)
49. {
50. text[k + 1] = text[k];
51. }
52. text[j] = ' ';
53. text[pos + 2] = '\0';
54. }
55. }
56. ReleaseMutex(mutex);
57. }
58. return 0;
59. }
60. int main()
61. {
62. setlocale(LC\_ALL, "rus");
63. HANDLE arr[2];
64. int data\_1 = 0, data\_2 = 0;
65. HANDLE handle\_1 = CreateThread(NULL, 0, thread\_1, &data\_1, 0, NULL);
66. HANDLE handle\_2 = CreateThread(NULL, 0, thread\_2, &data\_2, 0, NULL);
67. arr[0] = handle\_1;
68. arr[1] = handle\_2;
69. WaitForMultipleObjects(2, arr, TRUE, INFINITY);
70. CloseHandle(handle\_1);
71. CloseHandle(handle\_2);
72. return 0;
73. }

Синхронный вывод



Асинхронныйвывод

