Отчёт по лабораторной работе №3

Морозов Андрей 3.3 МО

*1. Текст задания*

Программы живут не только линейно - до завершения алгоритма, но и отвечая на события: нажатие кнопок в интерфейсе; события ОС или сигналы устройств. В их жизненном цикле участвуют и события, и потоки.

#include <iostream>

#include <conio.h>

#include <stdio.h>

int main()

{

clock\_t start, end;

start = clock();

std::cout << "Press anykey to start\n";

char x = \_getch();

std::cout << "start - " << x << std::endl;

while (!\_kbhit()) // буфер не пустой?

{

end = clock();

if ((double)(end - start) / CLOCKS\_PER\_SEC >= 1) //прошла ли секунда?

{

std::cout << "the program is still alive!\n";

start = clock();

}

}

}

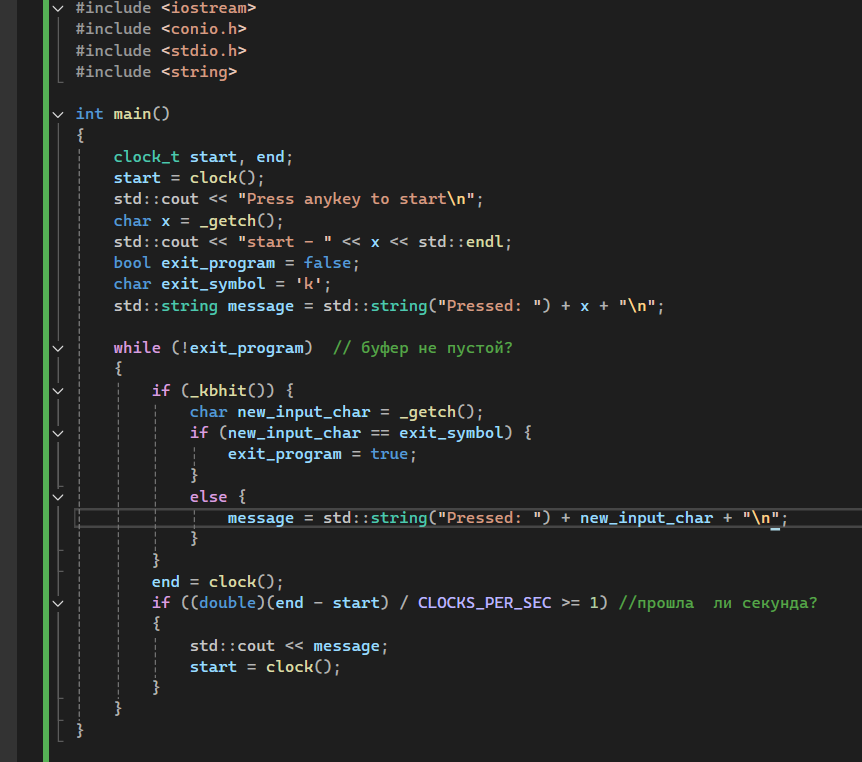
Каждую секунду программа выводит текст. Модифицируйте её так, чтобы по нажатию кнопки изменялся выводимый текст. А по нажатию определённой был выход из программы.  
 О методе*\_kbhit -* чтения из буффера ввода:

<https://docs.microsoft.com/ru-ru/cpp/c-runtime-library/reference/kbhit?view=vs-2017>

О методе*\_getch*

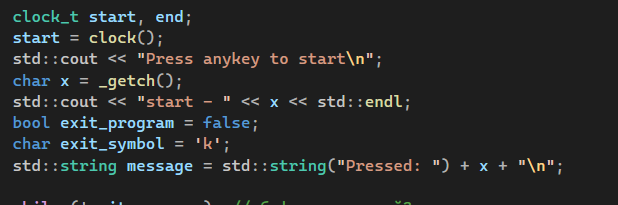
<https://docs.microsoft.com/ru-ru/cpp/c-runtime-library/reference/getch-getwch?view=msvc-160>

*2. Отредактированный код программы*

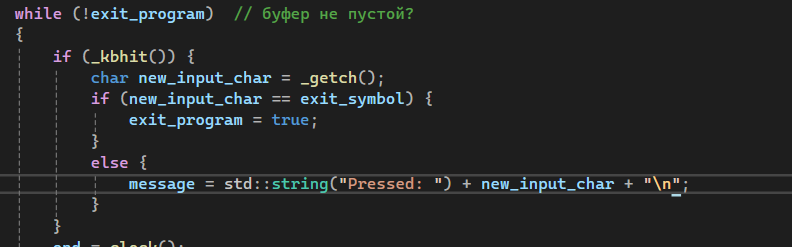


*3. Пояснение фрагментов кода*

Первоначальный ввод данных и инициализация переменных

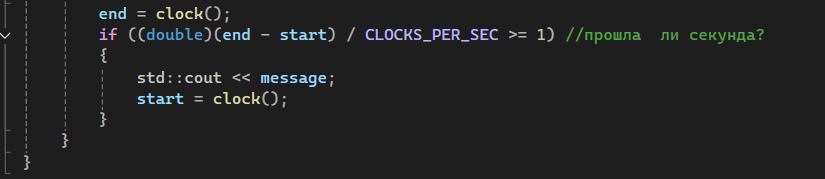
Отличается от исходной программы созданием флага **exit\_program** равным false по умолчанию, чтобы он стал условием завершения while цикла в дальнейшем вместо **!\_kbhit()**, так как в противном случае при повторном нажатии клавиши **!\_kbhit()** обращается в false пока нажата клавиша. За это время происходит выход из цикла и завершение программы. Нам же нужно выйти из программы только в случае нажатия на **exit\_symbol,** который в моём случае равен **’k’**.

Проверка факта нажатия и считывание символа



По той причине, что **\_getch()** блокирует вызывающий поток, я добавил проверку на факт нажатия на кнопку: if (\_kbhit()) {...}, чтобы не замораживать работу программы при отсутствии факта нажатия клавиши. Далее в переменную new\_input\_char кладём новый символ и сверяем с exit\_symbol: совпадают - выход, нет - меняем текст выводимого сообщения.

Финальный фрагмент кода, остался неизменным



Вывод сообщения по прошествию секунды

*4. Пример работы программы (вывод в консоли)*

