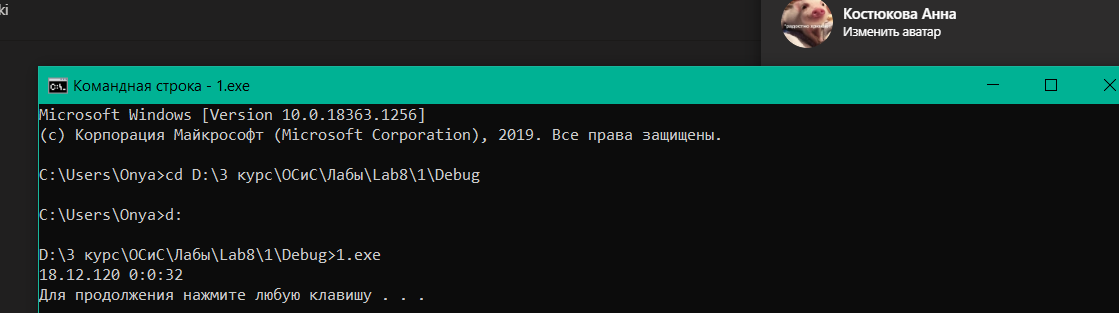
**Задание 01. Windows**

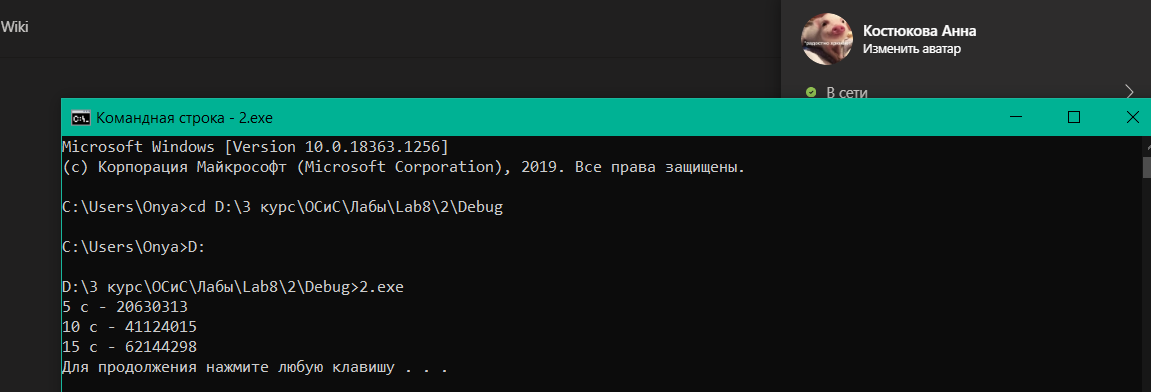
- Скриншот с запуском приложения и результатом.



Функция time() возвращает текущее календарное время системы. Если аргумент не является нулевым указателем, ей передается значение времени типа time\_t.

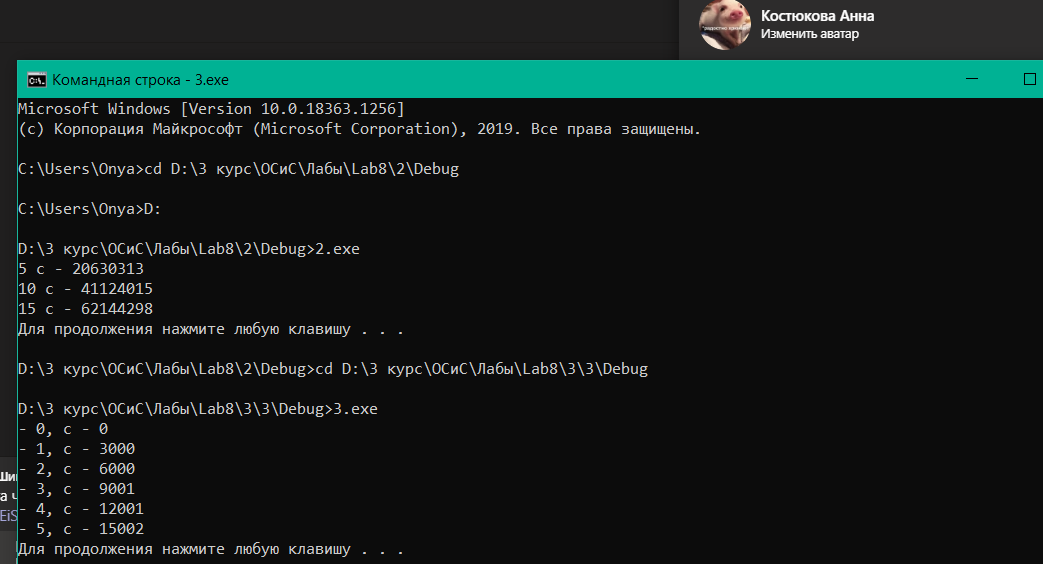
localtime\_s(const tm\* tm, const time\_t \* time) - функция преобразовывает текущее значение времени, передаваемое как аргумент, через указатель time на time\_t в структуру tm. Время указанное в time используется для заполнения структуры tm значениями текущей даты и времени.

**Задание 02. Windows**



Функция clock возвращает количество временных тактов, прошедших с начала запуска программы. С помощью [макроса CLOCKS\_PER\_SEC](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-ctime-time-h/makros-clocks_per_sec/) функция получает количество пройденных тактов за 1 секунду. Таким образом, зная сколько выполняется тактов в секунду, зная время запуска программы можно посчитать время работы всей программы или отдельного её фрагмента, что и делает данная функция.

**Задание 03. Windows**



function CreateWaitableTimer(

  lpTimerAttributes: PSecurityAttributes;     // Адрес структуры

                                              // TSecurityAttributes

  bManualReset: BOOL;  // Задает, будет ли таймер переходить в

                       // сигнальное состояние по завершении функции

                       // ожидания

  lpTimerName: PChar   // Имя объекта

) – создание таймера ожидания

Когда параметр bManualReset равен TRUE, то таймер после срабатывания функции ожидания остается в сигнальном состоянии до явного вызова SetWaitableTimer, если FALSE-таймер автоматически переходит в несигнальное состояние.

Если lpTimerName совпадает с именем уже существующего в системе таймера, то  функция возвращает его идентификатор, позволяя использовать объект для синхронизации между процессами. Имя таймера не должно совпадать с именем уже существующих объектов типов event, semaphore, mutex, job или file-mapping.

BOOL SetWaitableTimer(

HANDLE hTimer, - дескриптор таймера

const LARGE\_INTEGER \*lpDueTime, определяет время перехода таймера в свободное состояние (signaled) или время срабатывания таймера

LONG lPeriod, определяет режим работы и одновременно период повторения срабатываний ожидающего таймера

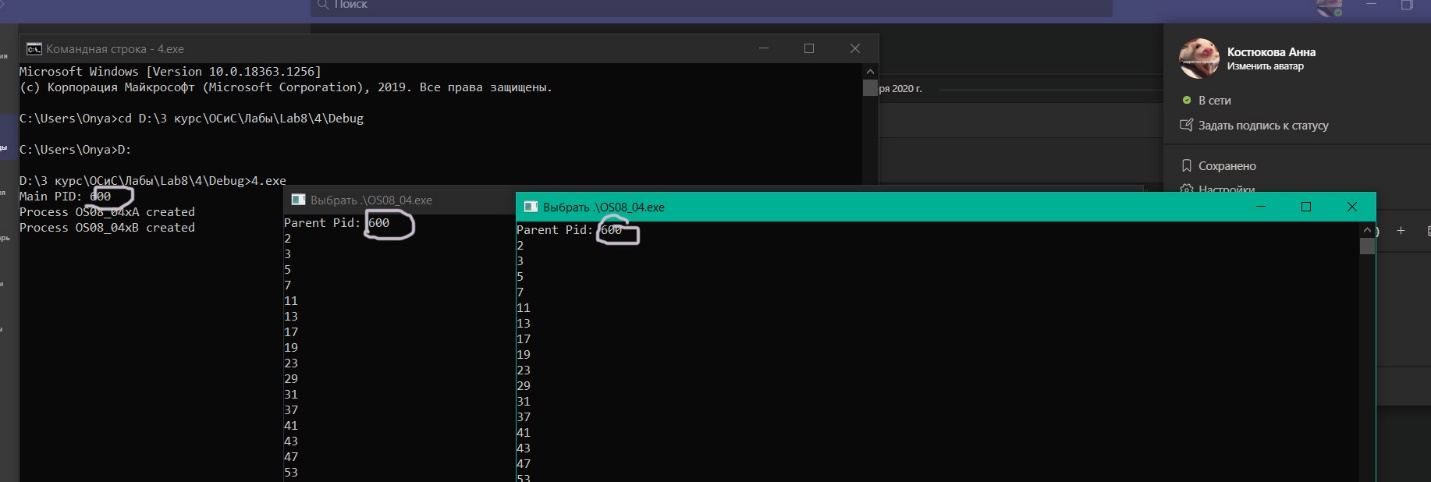
PTIMERAPCROUTINE pfnCompletionRoutine, определяет указатель на необязательную функцию асинхронного вызова (APC), которая помещается в очередь функций APC в момент срабатывания таймера и затем вызывается

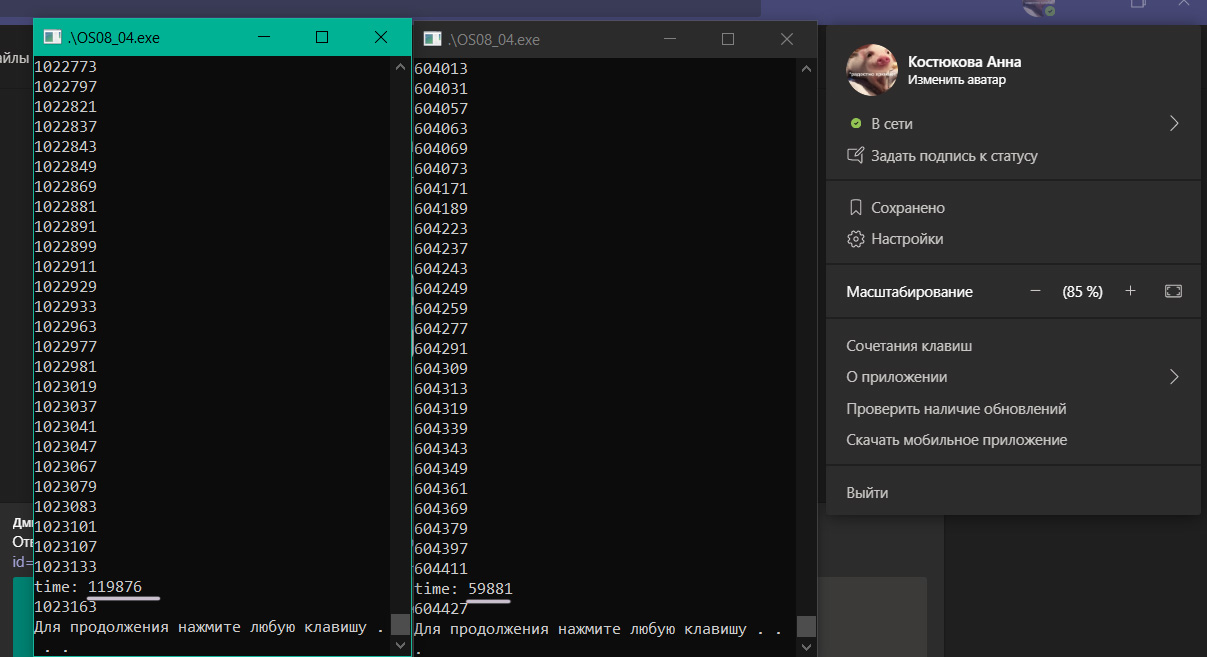
LPVOID lpArgToCompletionRoutine, параметр передает в функцию асинхронного вызова (APC) произвольный аргумент, например указатель на объект или структуру

BOOL fResume - если он не нуль, выводит машину из спящего состояния по срабатыванию таймера

); - Активирует указанный таймер ожидания. Когда приходит должное время, таймер сигнализируется, и поток, который установил таймер, вызывает необязательную процедуру завершения.

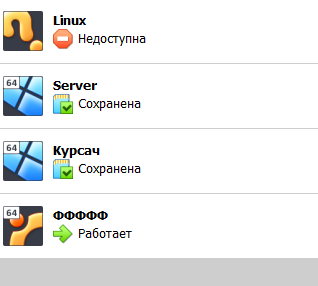
**Задание 04. Windows**



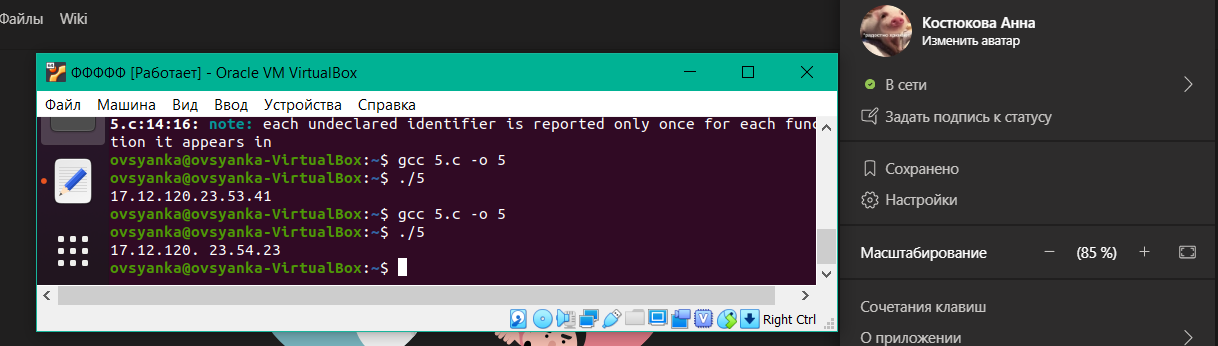


**Задание 05. Linux**

Перед выполнением последних заданий первый линукс решил отдохнуть, и я установила новый.



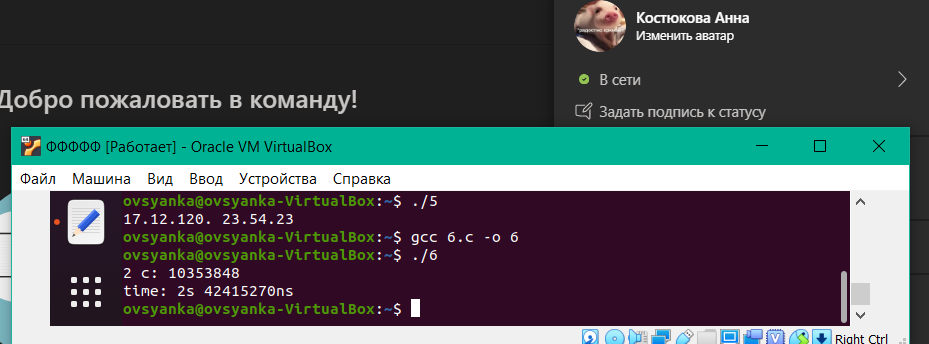
Поэтому расхождение в имени виртуалки в 7 и 8 отчетах.

****

time(time\_t \*second) – эта функция возвращает время с 00:00:00 UTC 1 января 1970 г. (временная метка Unix) в секундах. Если second не является нулевым указателем, возвращаемое значение также сохраняется в объекте, на который указывает second.

Библиотечная функция struct tm \* gmtime (const time\_t \* timer) использует значение, указанное таймером, для заполнения структуры tm значениями, которые представляют соответствующее время, выраженное в универсальном координированном времени (UTC) или часовом поясе GMT.

**Задание 06. Linux**

****

Функция int clock\_gettime (clockid\_t clock\_id, struct timespec \*t\_time) - получает текущее время по часам, идентифицированным часами, сохраняя его в секундах и наносекундах в \* t\_time.

clock\_id – идентификатор часов.

t\_time – структура, в которую будет записано определенное время.