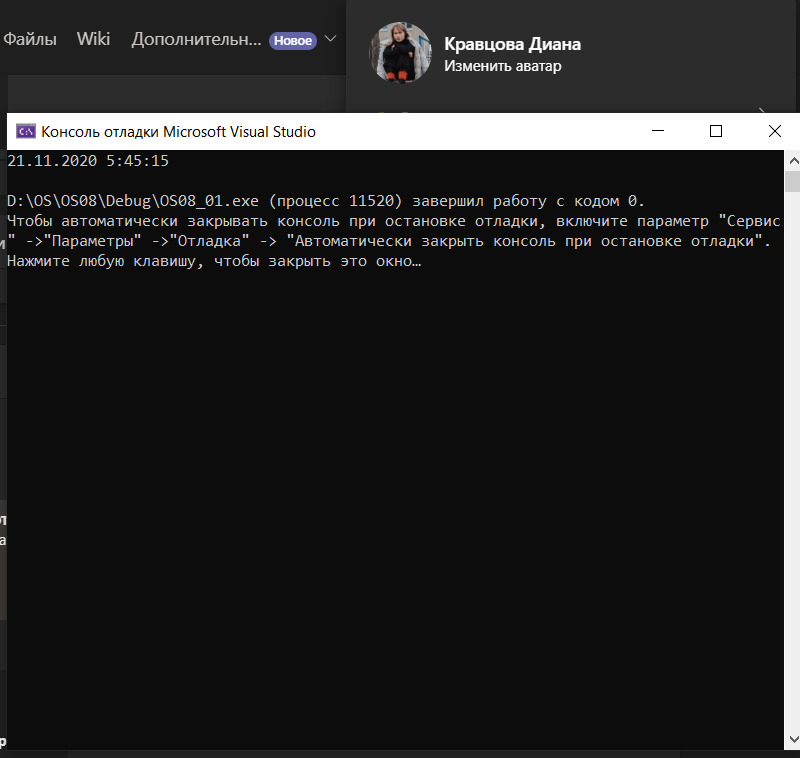
**Задание 1.**

****

**Расписать использумые функции для работы со временем.**

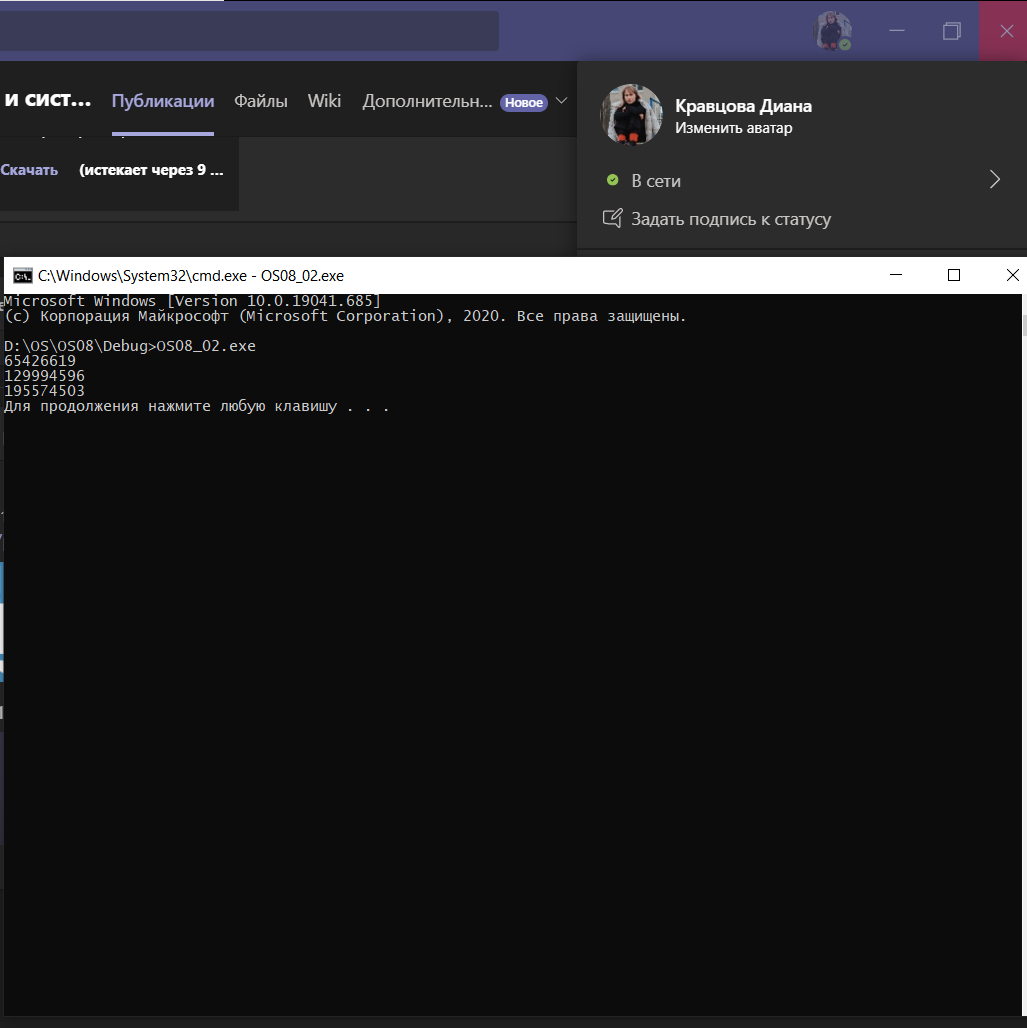
**Функция time() возвращает текущее календарное время системы.**

**Функция time() может вызываться либо с указателем NULL, либо с указателем на переменную time\_t. В последнем случае этот аргумент также получает значение календарного времени.**

**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 2.**

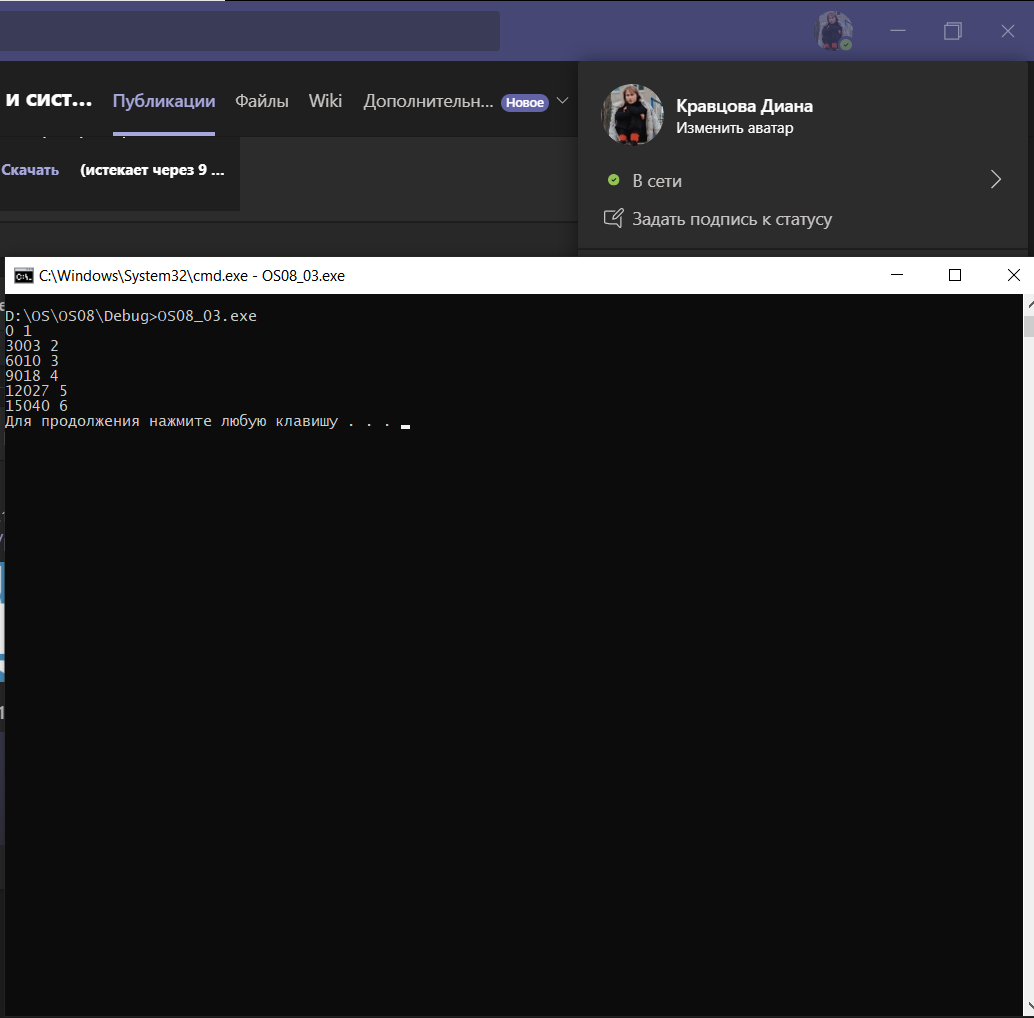
**Скриншот с запуском приложения и результатом.**

****

**clock() - Возвращает количество временных тактов, прошедших с начала запуска программы.**

**Задание 3.**

**Скриншот с запуском приложения и результатом.**

****

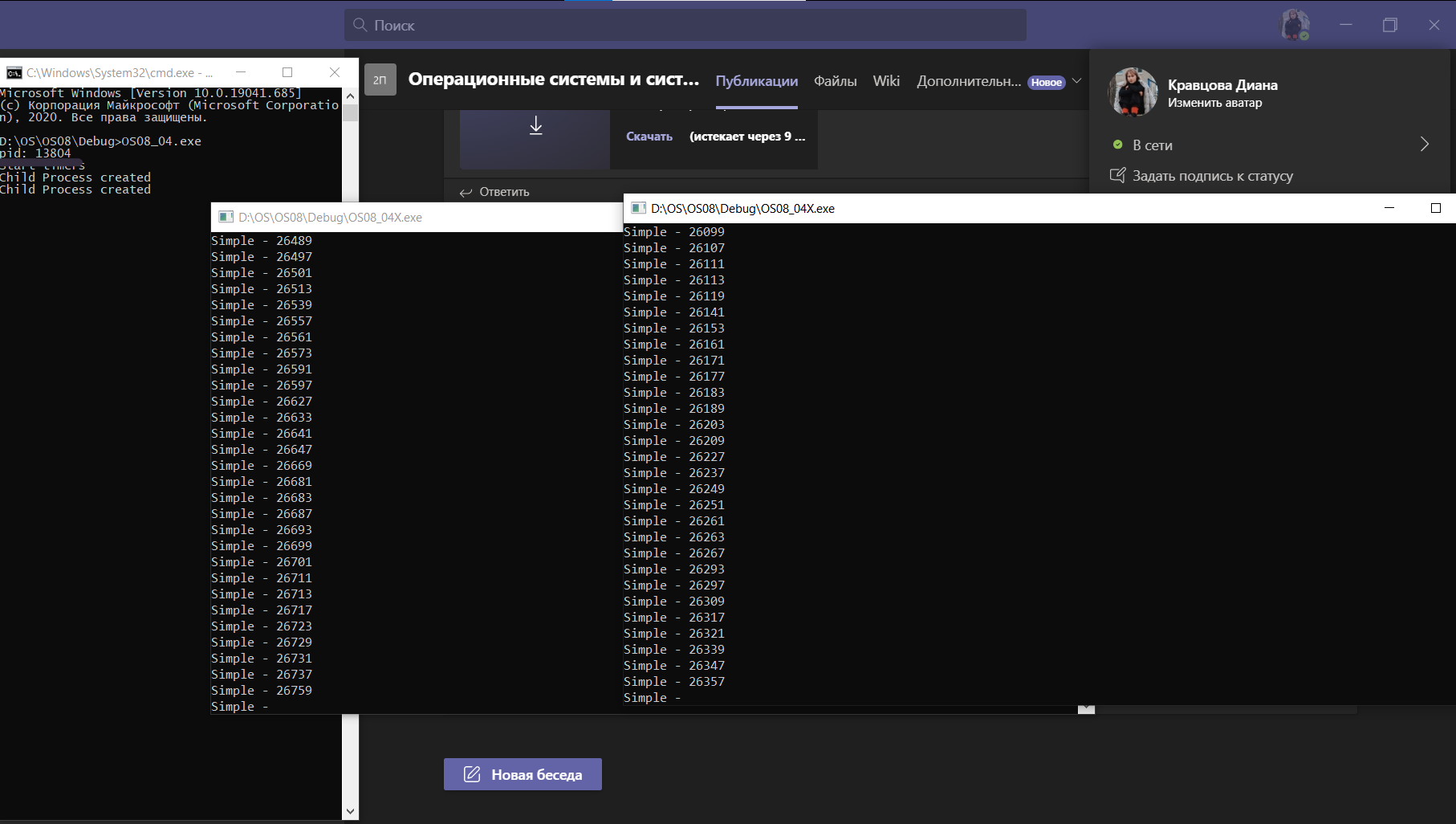
**CreateWaitableTimer - функция создает ожидающий таймер в занятом состоянии, из которого он выводится принудительно, т.е. после создания объект не активен. Таймер может быть двух типов с автосбросом или ручным сбросом, определяется параметром bManualReset. Если таймер с ручным сбросом, то при переходе в свободное состояние запускаются все потоки, которые его ожидали. Таймер с автосбросом запускает только один поток. Первый параметр обычно всегда NULL, последний - имя таймера. Имя используется для совместного разделения объекта между процессами.**

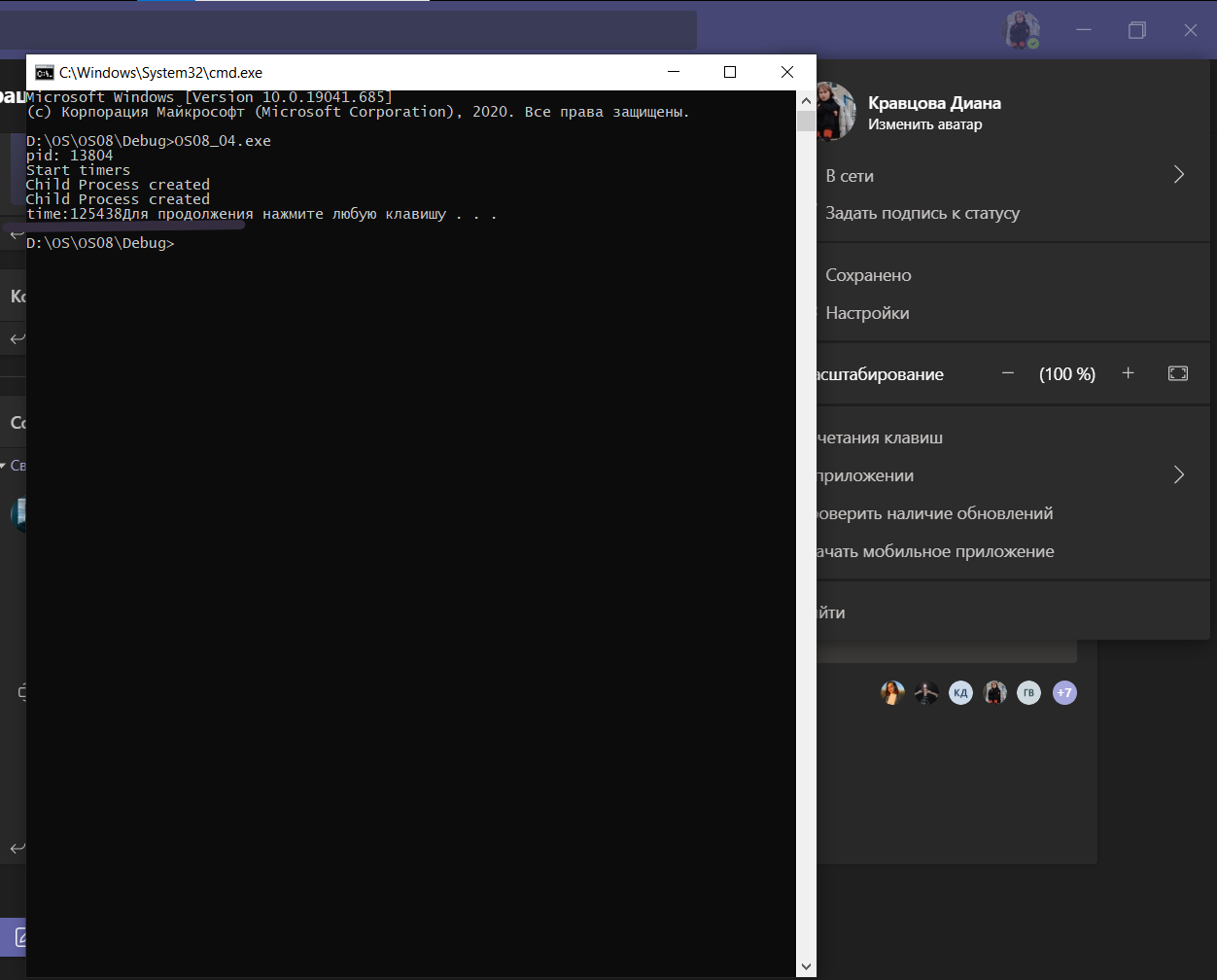
**SetWaitableTimer - функция запускает таймер и определяет все его параметры:**

1. **дескриптор таймера:**
2. **время перехода таймера в свободное состояние (signaled) или время срабатывания таймера;**
3. **режим работы и одновременно период повторения срабатываний ожидающего таймера;**
4. **указатель на необязательную функцию асинхронного вызова (APC), которая помещается в очередь функций APC в момент срабатывания таймера и затем вызывается;**
5. **передает в функцию асинхронного вызова (APC) произвольный аргумент, например указатель на объект или структуру;**
6. **bResume, если он не нуль, выводит машину из спящего состояния по срабатыванию таймера.**

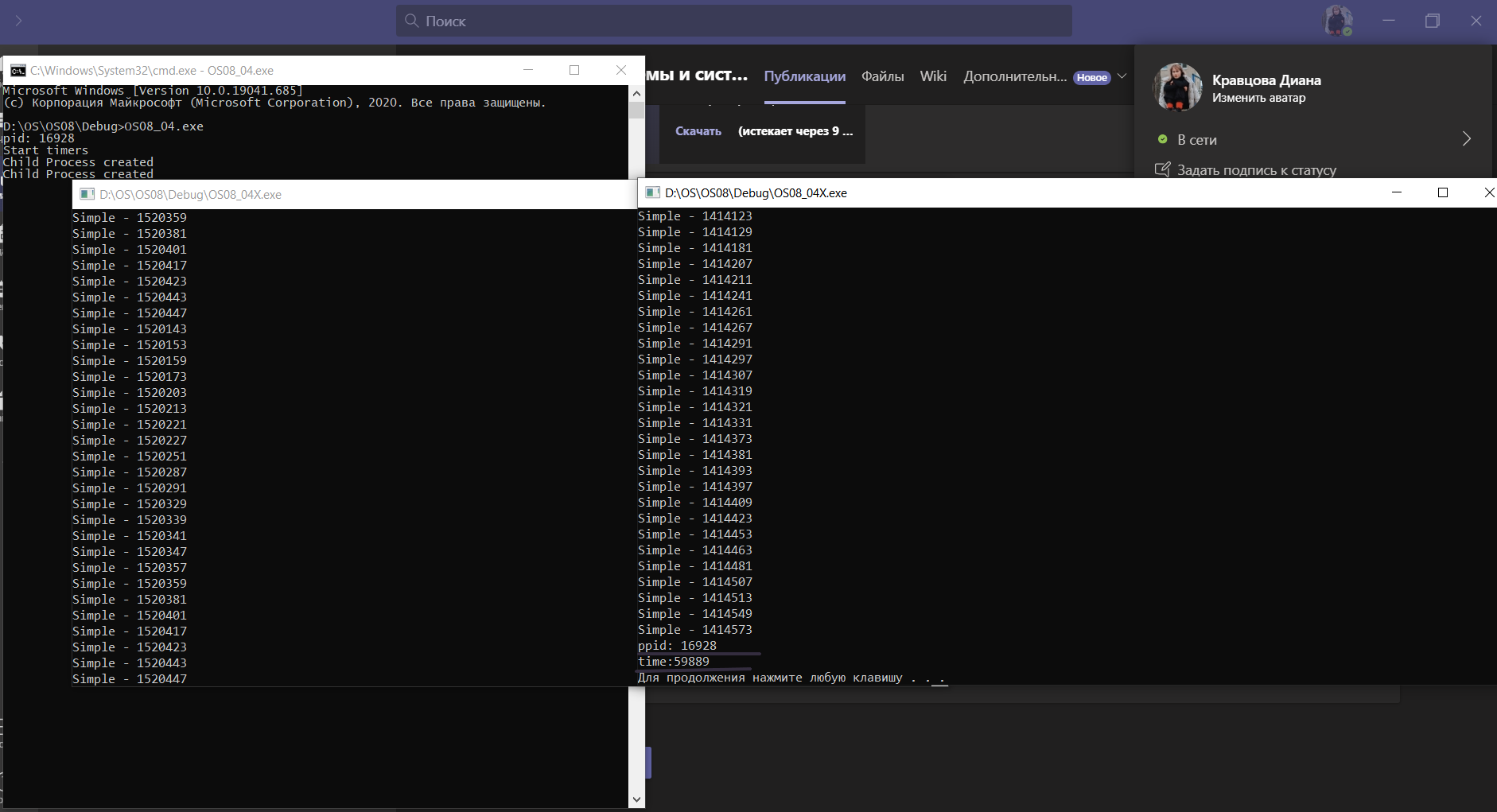
**Задание 4.**

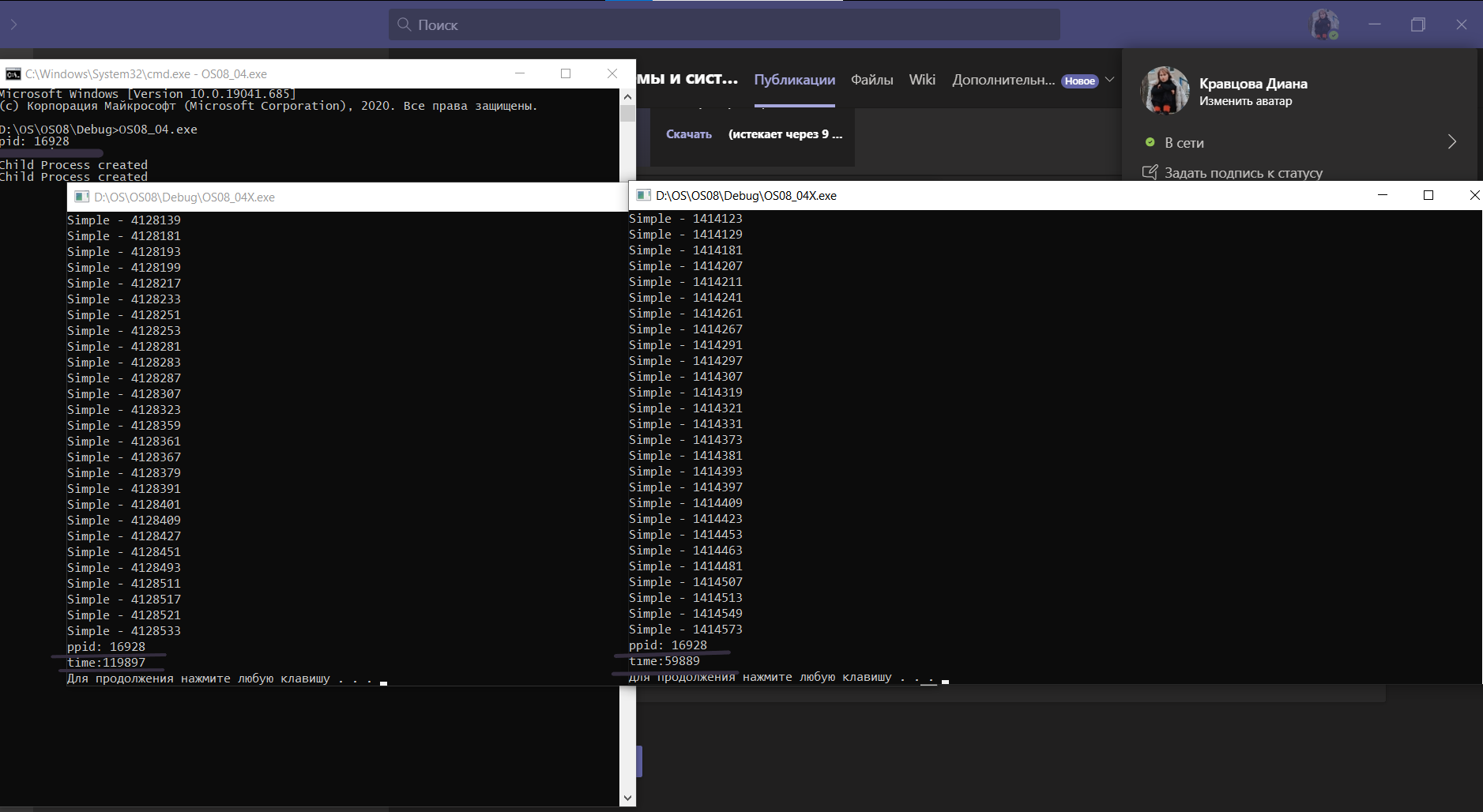
**Скриншот с запуском приложения и результатом.**

****

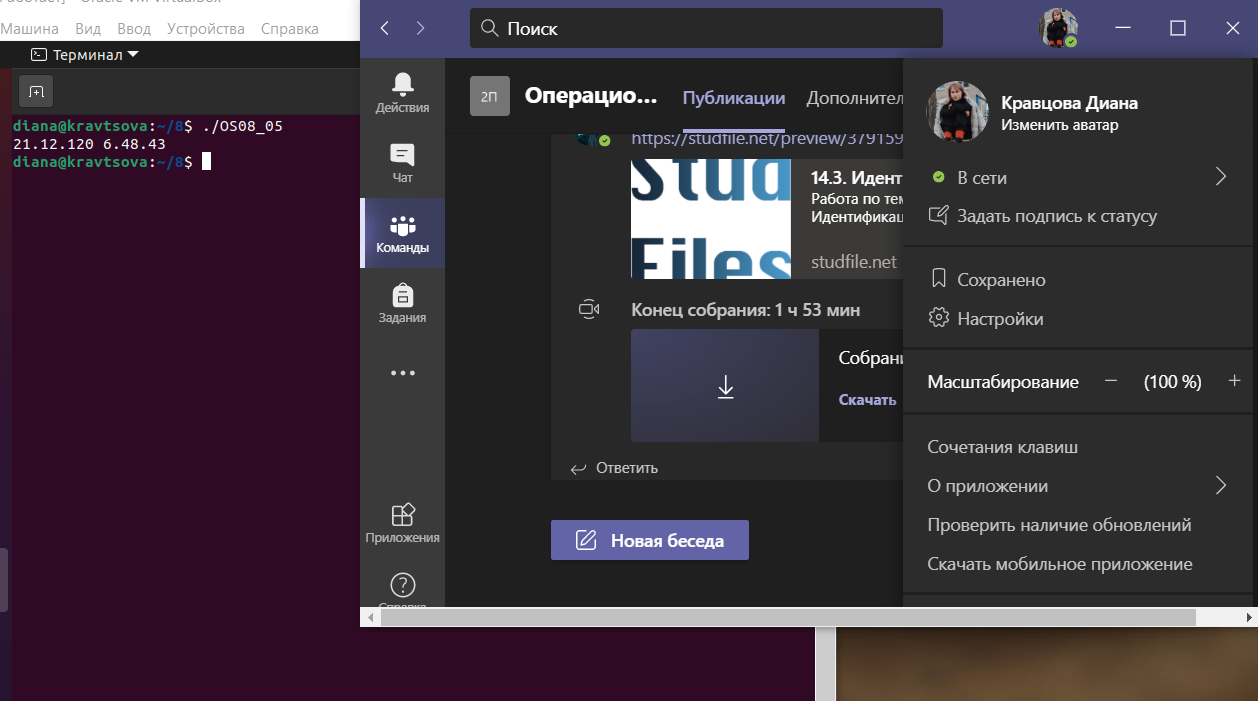


**2 скриншота с результатами дочерних процессов.**

****



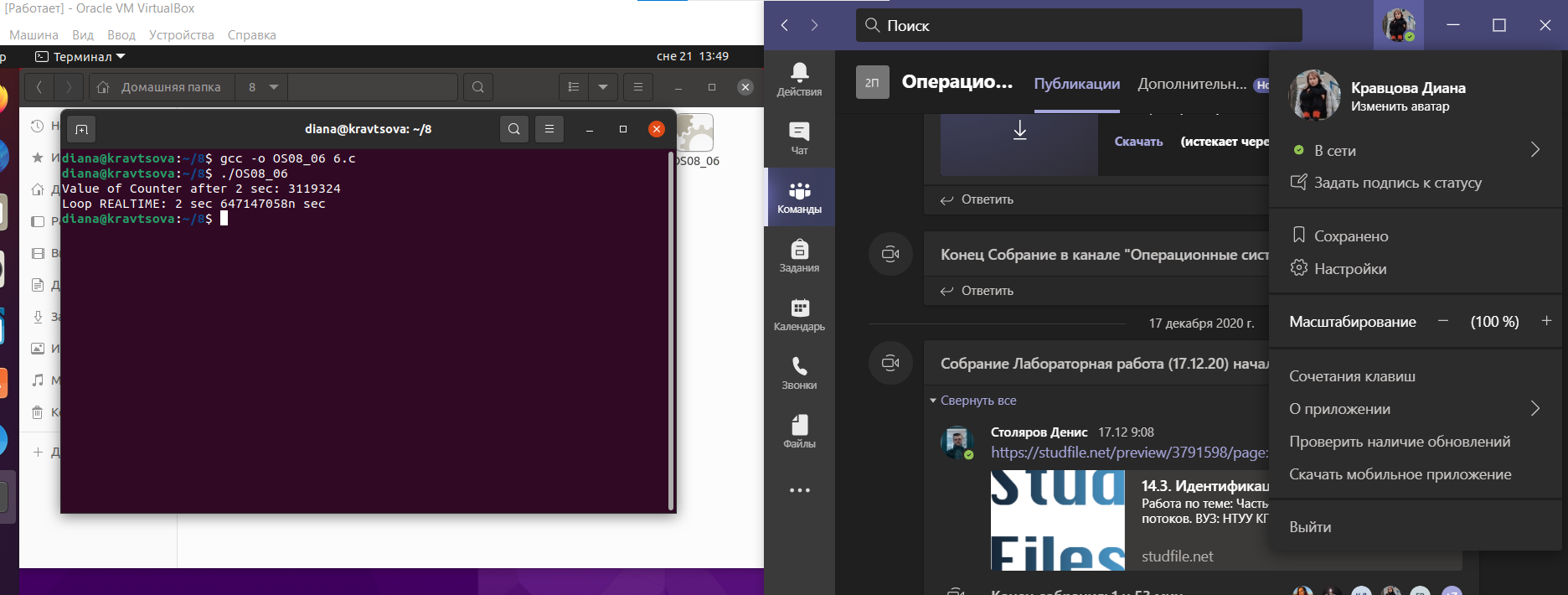
**Задание 5.**

****

**time – функция, которая 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 gmtime использует значение, указанное таймером, для заполнения структуры tm значениями, которые представляют соответствующее время, выраженное в универсальном координированном времени (UTC) или часовом поясе GMT.**

**Задание 6.**

**** Функция **clock\_gettime()** записывает текущие показания часов, определенных идентификатором clock\_id в структуру, на которую указывает t\_time.