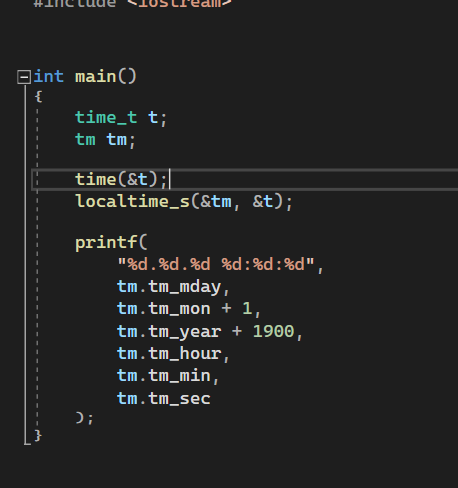
Лабораторная работа 08

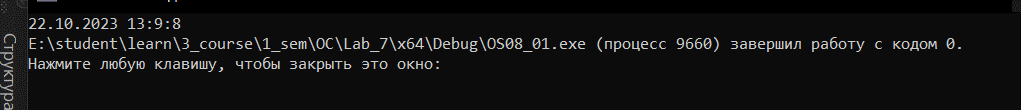
Компьютерное время

OC, ПОИТ-3

**Задание 01. Windows**

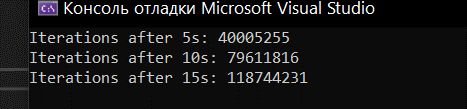
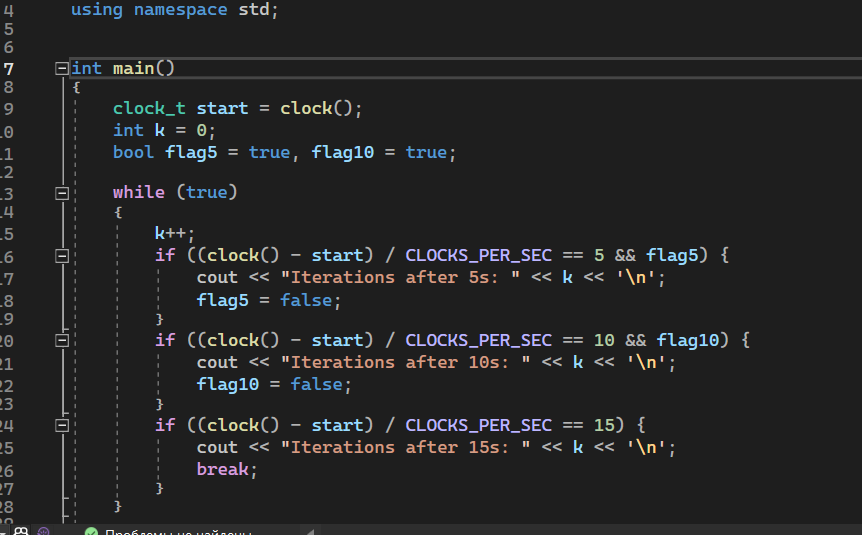
1. Разработайте приложение **OS08\_01**.
2. Приложение **OS08\_01** выводит на консоль текущую локальную дату и время в формате ***дд.мм.ггг чч:мин:сек***.

****



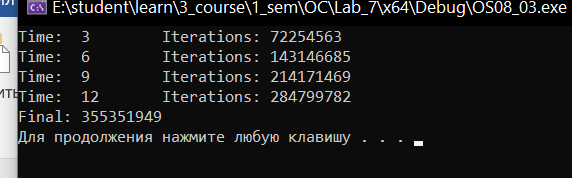
**Задание 02. Windows**

1. Разработайте приложение **OS08\_02,** выполняющее бесконечный цикл.
2. В теле цикла подсчитывается количество итераций.
3. Выведите на консоль значения счетчика итераций через 5 сек. и 10 сек.
4. Корректно завершите работу цикла и приложения через 15 сек., выведите итоговое значение счетчика итераций.



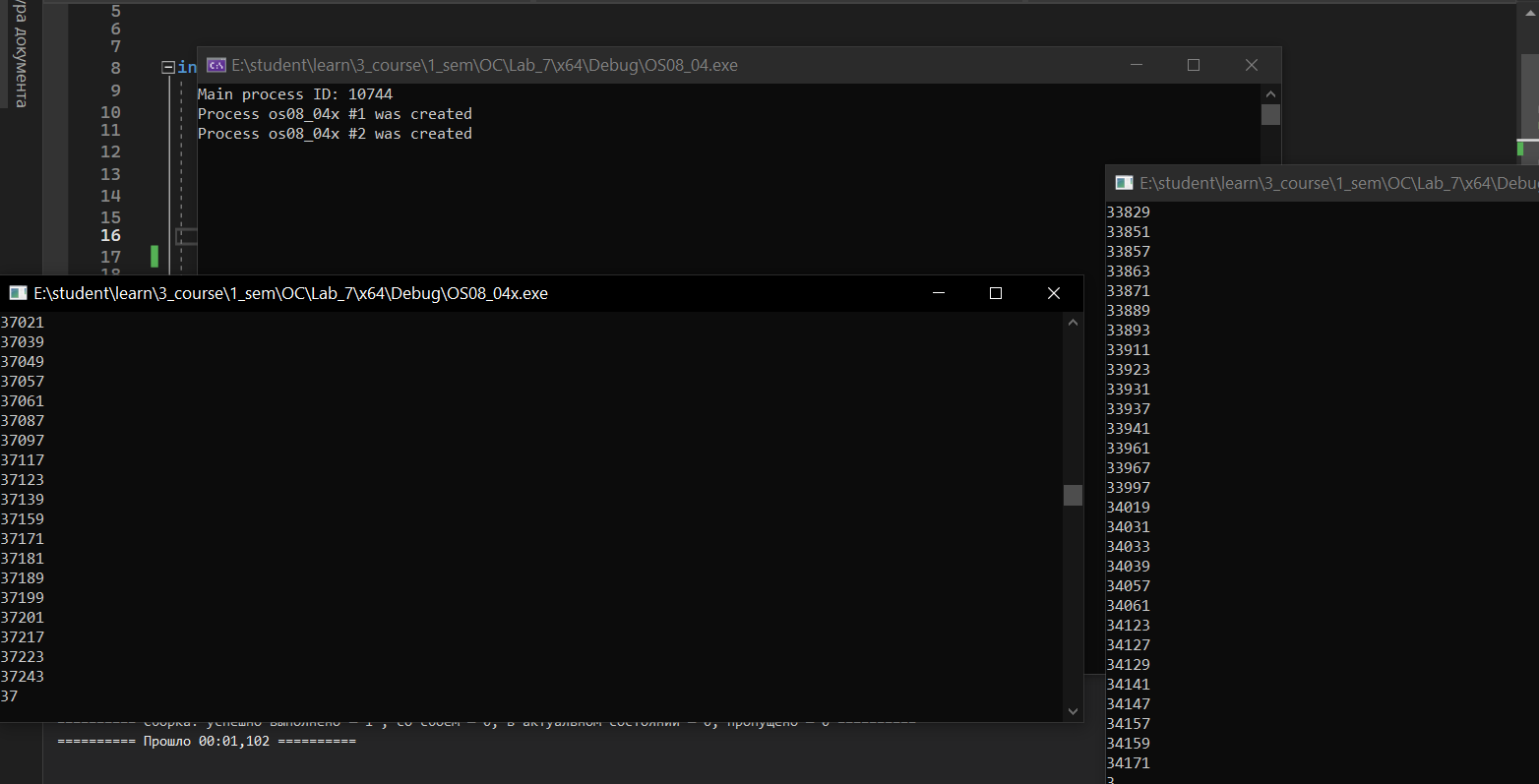
**Задание 03. Windows**

1. **Указание: самостоятельно освойте и примените периодический ожидающий таймер**
2. Разработайте приложение **OS08\_03,** выполняющее бесконечный цикл.
3. В теле цикла с задержкой подсчитывается количество итераций.
4. Выведите на консоль значения счетчика итераций каждые 3 сек.
5. Корректно завершите работу цикла и приложения через 15 сек., выведите итоговое значение счетчика итераций.



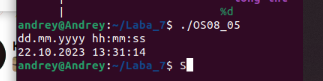
**Задание 04. Windows**

1. Разработайте приложение **OS08\_04,** запускающее два одинаковых дочерних процесса **OS08\_04\_X**.
2. Процессы **OS08\_04\_X** вычисляют и выводят на консоль (каждый в свою) пронумерованный ряд простых положительных чисел (простое число делится нацело только на себя и 1).
3. Первый дочерний процесс должен выполняться 1 минуту и корректно завершаться. 0020
4. Первый дочерний процесс должен выполняться 2 минуты и корректно завершаться.
5. Приложение **OS08\_04** завершается после завершения дочерних процессов.

****

**Задание 05. Linux**

1. Разработайте приложение **OS08\_05**.
2. Приложение **OS08\_05** выводит на консоль текущую локальную дату и время в формате ***дд.мм.ггг чч:мин:сек***.

****

**Задание 06. Linux**

1. Разработайте приложение **OS08\_06,** выполняющее бесконечный цикл.
2. В теле цикла подсчитывается количество итераций.
3. Выведите на консоль значения счетчика итераций через 2 сек. **процессорного** времени и корректно завершите цикл.
4. Приложение **OS08\_06 должно** выполнять замер реального затраченного на работу цикла времени и выводить его значения на консоль.



**Задание 07.** Ответьте на следующие вопросы

1. Поясните понятие «социальное время» и почему оно не монотонное?
2. Поясните понятие «эпоха Linux», назовите стартовую дату «эпохи Linux» и в каких единицах изменяется время?
3. Поясните понятие «Universal Coordinated Time (UCT)».
4. Поясните понятия «относительное время» и «абсолютное время».
5. Поясните понятие «тик».
6. Поясните понятие «ожидающий таймер», перечислите типы таймеров, перечислите состояния, в которых может находится таймер.
7. Перечислите типы часов, используемых в Linux, поясните их назначение.
8. Поясните назначение констант HZ, CLOCKS\_PER\_SEC.

23. \*\*Социальное время\*\* - это то, как мы воспринимаем и измеряем время в повседневной жизни, и оно может отличаться у разных людей и культур. Это происходит потому, что каждый человек и культура имеют свои собственные представления о том, как время должно течь. Например, для одних людей важно быть всегда точно вовремя, а для других гибкость и отсутствие строгого расписания важнее. Поэтому социальное время не является однородным и не всегда движется равномерно.

24. \*\*Эпоха Linux\*\* - это специальная точка во времени, с которой начинается отсчет времени в операционной системе Linux. Эта точка - полночь 1 января 1970 года (01.01.1970 00:00:00 по всемирному координированному времени, или UTC). Время в Linux измеряется в секундах, прошедших с этой даты.

25. \*\*Universal Coordinated Time (UTC)\*\* - это мировое координированное время, которое используется по всему миру как единое стандартное время. Оно синхронизировано с атомными часами и предназначено для обеспечения единообразия времени во всем мире. Важно, потому что оно не зависит от местоположения и является точным мировым временным стандартом.

26. \*\*Относительное время\*\* - это способ измерения времени относительно какой-то событийной точки или начала. Например, когда мы говорим, что что-то произошло "через 5 минут", мы используем относительное время, указывая на прошедшее время относительно текущего момента. \*\*Абсолютное время\*\* - это измерение времени без ссылки на другие события и точки отсчета. Например, когда мы указываем, что сейчас 10 часов, это абсолютное время, потому что оно не зависит от других событий.

27. \*\*Тик\*\* - это одна единица времени или интервала, которую мы используем для измерения времени или событий. Это может быть коротким периодом времени, например, одна секунда.

28. \*\*Ожидающий таймер\*\* - это специальный механизм в компьютерных системах, который позволяет выполнять задачи или события через определенное время или с определенным интервалом. Он может быть настроен на работу по разным правилам, таким как:

- \*\*Относительные таймеры\*\*: срабатывают через определенное время после установки.

- \*\*Абсолютные таймеры\*\*: срабатывают в определенный момент времени.

- \*\*Повторяющиеся таймеры\*\*: срабатывают периодически, чередуясь через определенные интервалы.

- \*\*Одноразовые таймеры\*\*: срабатывают только один раз после установки.

Таймеры могут находиться в разных состояниях, например, ожидания, активации, сработавшие и так далее.

29. В Linux существуют разные типы часов:

- \*\*Часы системного времени (System Time)\*\*: Они показывают текущее время в операционной системе, включая дату и время с начала эпохи Linux.

- \*\*Часы реального времени (Real-Time Clock, RTC)\*\*: Эти часы обычно находятся на материнской плате компьютера и позволяют хранить текущее время и дату даже после выключения компьютера. Они используются для сохранения времени между перезагрузками.

- \*\*Часы ядра (Kernel Clock)\*\*: Это часы, используемые ядром операционной системы для внутренних задач.

30. \*\*HZ\*\* - это число, которое определяет, как часто операционная система будет выполнять определенные задачи в секунду. Оно влияет на то, насколько часто система будет выполнять планирование задач и обработку прерываний. Обычно в старых версиях Linux HZ равно 100, а в новых - 1000. \*\*CLOCKS\_PER\_SEC\*\* - это константа, которая определяет, сколько тактов системного времени приходится на одну секунду. Она используется при работе с функциями времени в программировании на языке C и обычно равна 1000000 (один миллион), что означает, что время измеряется в микросекундах.