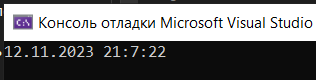
Лабораторная работа 08

Компьютерное время

OC, ПОИТ-3

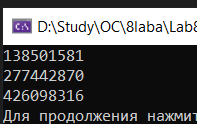
**Задание 01. Windows**

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***.



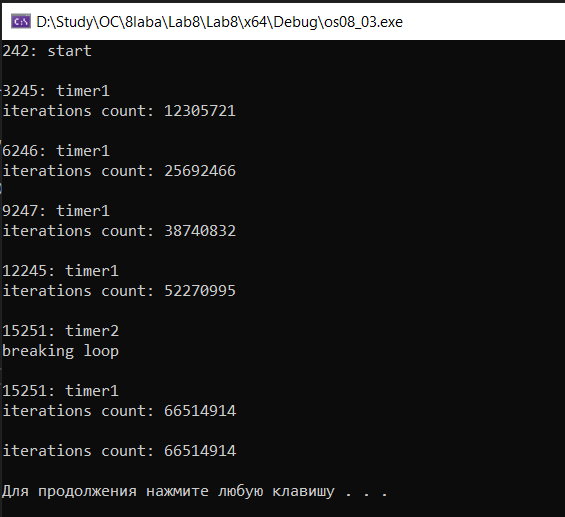
**Задание 02. Windows**

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.



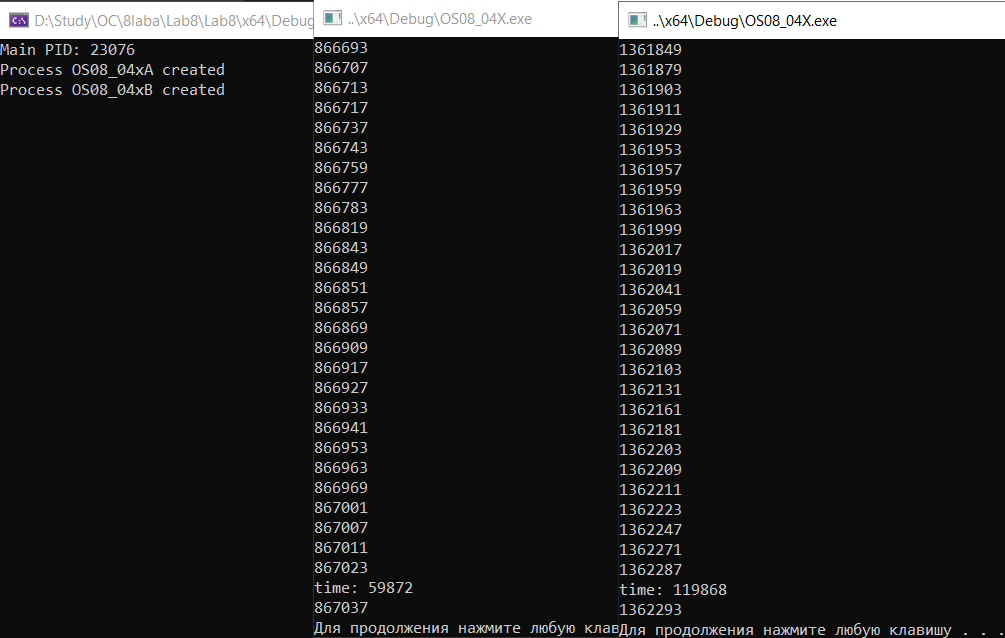
**Задание 03. Windows**

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.
4. Первый дочерний процесс должен выполняться 2 минуты и корректно завершаться.
5. Приложение **OS08\_04** завершается после завершения дочерних процессов.



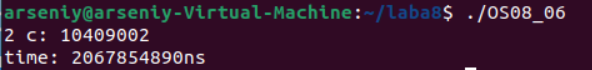
**Задание 05. Linux**

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. Дайте определение понятию «синхронизация потоков».**

Синхронизация - механизм упорядочивания выполнения программных блоков двух или более потоков. (Лекция)

Синхронизация потоков (Thread Synchronization) — это обобщенный термин, относящийся к процессу взаимодействия и взаимосвязи потоков. (<https://naparah.com/evm/09131501.html>)

Синхронизация потоков – механизм, позволяющий потокам согласовывать свою работу с общими ресурсами. Этот механизм представляет собой набор объектов операционной системы, которые создаются и управляются программно, являются общими для всех нитей в системе (некоторые - для нитей, принадлежащих одному процессу) и используются для координирования доступа к ресурсам. В качестве ресурсов может выступать все, что может быть общим для двух и более нитей - файл на диске, порт, запись в базе данных, объект GDI, и даже глобальная переменная программы (которая может быть доступна из нитей, принадлежащих одному процессу). (http://www.codenet.ru/progr/cpp/process-threads-sync.php)

**2. Объясните понятие «взаимная блокировка».**

**Взаимная блокировка** – это ситуация в которой, два или более процесса(потока) занимая некоторые ресурсы, пытаются заполучить некоторые другие ресурсы, занятые другими процессами/потоками и ни один из процессов/потоков не может занять необходимый им ресурс, и соответственно освободить занимаемый. (<https://javarush.ru/groups/posts/1422-vzaimnaja-blokirovkadeadlock-v-java-i-metodih-borjhbih-s-ney>)

**Взаимная блокировка** (deadlock) - явление при котором все потоки находятся в режиме ожидания. Происходит, когда достигаются состояния:

· взаимного исключения: по крайней мере один ресурс занят в режиме неделимости и, следовательно, только один поток может использовать ресурс в любой данный момент времени.

· удержания и ожидания: поток удерживает как минимум один ресурс и запрашивает дополнительные ресурсов, которые удерживаются другими потоками.

· отсутствия предочистки: операционная система не переназначивает ресурсы: если они уже заняты, они должны отдаваться удерживающим потокам сразу же.

· цикличного ожидания: поток ждёт освобождения ресурса другим потоком, который в свою очередь ждёт освобождения ресурса заблокированного первым потоком.

3. Перечислите механизмы авторизации(синхранизации) OS.

- Critical section ;

- Mutex;

- Semaphore;

- Atomic operation (interlocking function)

- Event;

- **Waitable** timer.

(лекция)

В книге

Критические области

Взаимное исключение с активным ожиданием

Приостановка и активизация .

Семафоры

Мьютексы

Мониторы

Передача сообщений

Барьеры

**4. Поясните в чем разница между механизмом mutex и semaphore.**

Семафоры — это объекты режима ядра, поэтому они имеют дескрипторы безопасности и описатели. Мьютексы — это тоже объекты режима ядра, используемые для синхронизации, но они проще семафоров, поскольку не имеют счетчиков. (Книга)

В основе семафора лежит счётчик, над которым можно производить две [атомарные операции](https://ru.wikipedia.org/wiki/%D0%90%D1%82%D0%BE%D0%BC%D0%B0%D1%80%D0%BD%D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F): увеличение и уменьшение значения на единицу, мьютекс можно представить в виде переменной, которая может находиться в двух состояниях: в заблокированном и в незаблокированном. При входе в свою критическую секцию поток вызывает функцию перевода мьютекса в заблокированное состояние. При выходе из критической секции поток вызывает функцию перевода мьютекса в незаблокированное состояние.

5. Почему **mutex,** **semaphore, event** создают объект ядра OS, а **critical section** нет.

Critical section – механизм синхронизации нескольких потоков одного процесса, НЕ ЯВЛЯЕТСЯ объектом ядра OS. (Лекция)

6. Поясните понятие «социальное время» и почему оно не монотонное?

Социальное время это отсчет времени принятый в обществе

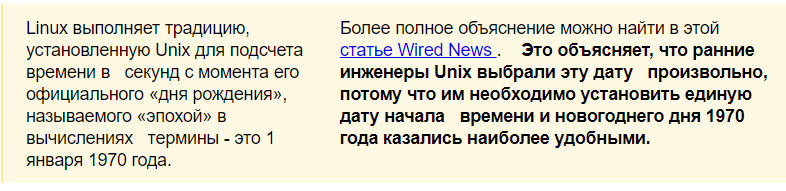
Социальное время не монотонное: так как каждый солнечный год увеличивается на 3 мс, люди измеряют время не точно, поэтому время от времени делались коррекции

(Лекция)

7. Поясните понятие «эпоха Linux», назовите стартовую дату «эпохи Linux» и в каких единицах изменяется время?

с полуночи (00:00:00 UTC) 1 января 1970 года (четверг); этот момент называют «эпохой Unix» (англ. Unix Epoch).

**эпоха Unix (POSIX-время)** c 01.01.1970 0:00:00 в секундах(из лекции).



**8. Поясните понятие «Universal Coordinated Time (UCT)».**

**Universal Coordinated Time (UCT):** универсальное согласованное время (на Гринвичском меридиане), усредненное значение полученное на основе данных 50 лабораторий, оборудованных атомными часами у которых расхождение с солнечными часами примерно 3мс (атомные часы отстают) в сутки,

9. Поясните понятия «относительное время» и «абсолютное время».

Относительное время — это время протекания одного действия по отношению ко времени другого действия.

Время, определяемое по отношению к моменту речи, принято называть абсолютным временем, а формы глагола, выражающие его,— формами абсолютного времени. Абсолютное время в грамматике противопоставляется относительному.

Относительное время - это время относительно какой-то точки, например, текущего времени. “Через два дня, 5 минут назад”

Абсолютное время - это конкретное время. 17:32 17 декабря 2020

10. Поясните понятие «тик».

Неофициальная единица измерения времени, равна продолжительности одного импульса тактового генератора (часов). (<https://stopudov.info/units/tick-1/>)

для измерения времени, прошедшего с какого-то начального момента. Например, в некоторых системах времени считается в тиках процессора.

11. Поясните понятие «ожидающий таймер», перечислите типы таймеров, перечислите состояния, в которых может находится таймер.

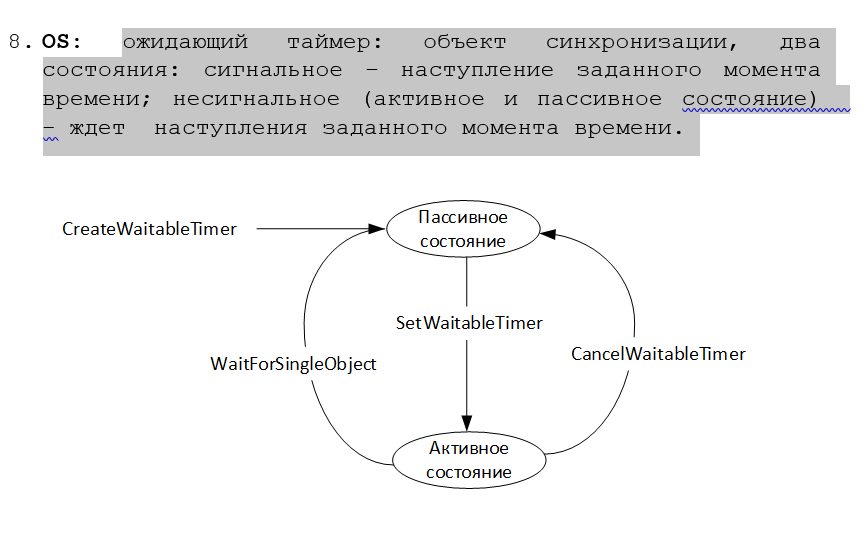
Ожидающие таймеры - объекты ядра, которые предназначены для отсчета промежутков времени, используется для синхронизации.

может быть двух типов с автосбросом или ручным сбросом

1. **Одноразовые (One-Shot) таймеры:** Эти таймеры срабатывают только один раз после установки интервала и ожидают наступления события. После срабатывания таймера его нужно переустановить, если требуется следующее событие.
2. **Периодические таймеры:** Эти таймеры срабатывают периодически с определенным интервалом. Они ожидают наступления событий и срабатывают снова и снова, пока не будут явно отключены.

Состояния, в которых может находиться таймер:

1. **Активный (Active):** Таймер активен и ожидает срабатывания в заданный момент времени.
2. **Ожидание (Waiting):** Таймер ожидает наступления своего события. В этом состоянии он находится до момента, когда произойдет событие.
3. **Отключен (Disabled):** Таймер может быть временно отключен, что означает, что он не будет срабатывать, пока не будет явно включен.
4. **Завершен (Expired):** Таймер сработал, т.е., произошло событие, для которого он был установлен.



(http://dslev.narod.ru/waittimer.html)

**12. Перечислите типы часов, используемых в Linux, поясните их назначение.**

Linux: четыре типа часов: **REALTIME** – системное время(настенное), **MONOTONIC** – с начала загрузки OS (монотонно возрастает), **PROCESS** – процессорное время (затраченное процессом), **THREAD** – процессорное время (затраченное потоком). (Лекция)

**13. Поясните назначение констант HZ, CLOCKS\_PER\_SEC.**

HZ – частота системного таймера (обычно, 100,250, 1000), параметр ядра (Лекция)

прерывание таймера генерируется с частотой константы HZ, т.е. HZ раз в секунду . (https://it.wikireading.ru/1847)

CLOCKS\_PER\_SEC - некое число, обозначающее количество тиков в секунду.

Используется для получения количества секунд(Разделив общее количество тиков на число тиков в секунду мы получаем количество секунд:

clock()/CLOCKS\_PER\_SEC = количество секунд. )