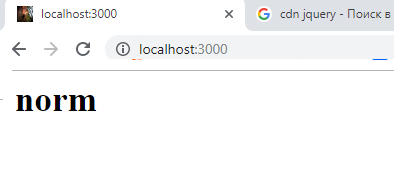
Лабораторная работа 03

ПСКП

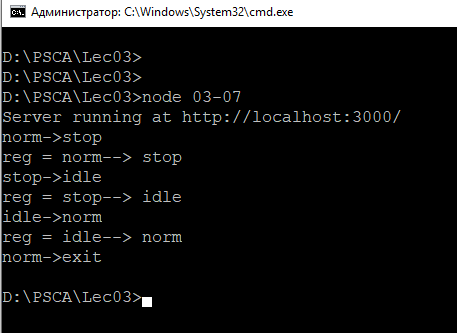
ПОИТ-3

**Задание 01**

1. Разработайте серверное приложение **03-01**, которое на запрос [**http://localhost:5000**](http://localhost:5000)возвращает страницу, отражающую состояние приложения (см. рис.).



1. Приложение может находиться в четырех состояниях: **norm**, **stop**, **test**, **idle**.
2. Состояние приложение переключается с помощью стандартного системного ввода, который назначен на консоль. Консоль в приглашении (prompt) указывает текущее состояние приложения.
3. Пользователь может ввести новое состояние (**norm**, **stop**, **test**, **idle**). При корректном вводе состояния осуществляется переключение состояния приложения.
4. При ошибочном вводе режима, ошибочная введенная последовательность символов просто отображается, но переключение режима не осуществляется.
5. Допускается ввод состояния exit, которое приводит к завершению приложения.
6. См. рис.

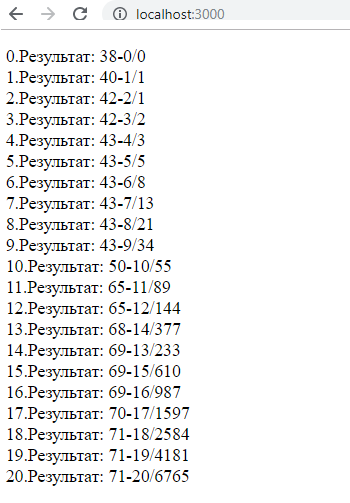


**Задание 02**

1. Разработайте серверное приложение **03-02**, которое на GET-запрос вида [**http://localhost:5000**](http://localhost:5000)**/fact?k=3** возвращает ответ, в теле которого содержится сообщение в json-формате вида {k:3, fact:6}, где **k** – полученное в качестве параметра значение, а **fact** – значение факториала.
2. Для расчета факториала используйте рекурсивный алгоритм.
3. Проверьте работоспособность приложения с помощью **POSTMAN**.

**Задание 03**

1. Доработайте приложение **03-02** таким образом, чтобы на GET-запрос приложение отправляло HTML-страницу, содержимое которой формировалось бы с помощью JS.
2. JS в цикле x = 1,…,20 с помощью функции fetch делает GET-запросы к [**http://localhost:5000/fact?k=x**](http://localhost:5000/fact?k=x) и содержимое ответа выводит в окно браузера, примерно так, как это представлено на следующем рисунке.



1. Результаты вычислений должны иметь следующий вид **t-k/fac**,где t – количество миллисекунд прошедшее с момента начала работы цикла запросов, k – параметр пересылаемый серверу, **fac** факториал k.
2. Выполните приложение запишите общую продолжительность всего цикла запросов.
3. Запустите приложение поочередно еще в двух вкладках браузера и запишите продолжительность всего цикла запросов для каждой вкладки.
4. Запустите приложение одновременно в трех вкладках браузера и запишите продолжительность всего цикла запросов для каждой вкладки.

**Задание 04**

1. Разработайте приложение **03-04** на основе приложения **03-02,** но функцию для вычисления факториала реализуйте асинхронной с помощью механизма **process.nextTick**.
2. Выполните аналогичные заданию 3 замеры.

**Задание 05**

1. Разработайте приложение **03-05** на основе приложения **03-02,** но функцию для вычисления факториала реализуйте асинхронной с помощью механизма **setImmediate**.
2. Выполните аналогичные заданию 3 замеры.

**Задание 06.Ответьте на следующие вопросы**

1. Перечислите основные свойства глобальные объекты Node.js и поясните их предназначение.

Глоб.объект – пер-ная и ф-ция доступна в любом месте проги

Global хранит var-д-е на уровне модуля

Process – инфа о среде выполнения

Buffer – для работы с двоичными д-ми

1. Поясните понятие «асинхронная функция».

Опред.с пом.ключ.слова async

Если async -> м.исп-ть await

Await -> приостанавл.ф-цию пока промисы не вып-нены или отклонены.

Ф-ция, после вызова кот JS прилож продолж.работать т.к.ф-ция сразу вып-ет возврат, а ее рез-т известен позже.

1. Поясните понятие стандартные «системные потоки».

Сущ.4 вида потоков (чтение, запись, транф-щие, дуплексные)

Ввод/вывод + ошибки

1. Поясните назначение функций **process.nextTick**, **setImmediate,** поясните в чем разница.

Event Loop (цикл событий) – то что позвол.вып-ть неблок.п-ции ввода/вывода путем выгрузки оп-ций в ядро системы когда это возможно.

Т.к.большинство ядер многопоточно и м.вып-ть неск.оп-ций -> когда 1 из этих оп-ций заверш, ядро сообщ, что соотв.ф-ция колбэка мб добавлена в очередь опроса и в итоге выполнена.

nextTick помещ.в начало очереди след.цикла event loop

setImmediate – в конец очереди след.цикла event loop

nextTick позвол.сделать что-то после того, как вы изменили д-е, но до того как браузер отобразил д-е на странице.