

Университет ИТМО, факультет инфокоммуникационных  
технологий Отчетная работа по «Информатике»: аннотация к статье

Выполнил Шишминцев Дмитрий Владимирович, К3121, 27.09.2022

<b>Название статьи/главы книги:</b> Приручаем многопоточность в Node.js (часть 1: базовые концепты)		
<b>ФИО автора статьи:</b> Боровиков Кирилл	<b>Дата публикации:</b> 21.09.2022	<b>Размер статьи</b> 1266 слов
<b>Прямая полная ссылка на источник и сокращенная ссылка:</b> <a href="https://habr.com/ru/company/tensor/blog/689144/">https://habr.com/ru/company/tensor/blog/689144/</a> <a href="https://shorturl.at/ABS35">https://shorturl.at/ABS35</a>		
<b>Тэги, ключевые слова или словосочетания</b> Высокая производительность, Программирование, JavaScript, NodeJS, многопоточность		
<b>Перечень фактов, упомянутых в статье:</b> Технология многопоточности заключается в том, что процесс может выполняться параллельно на нескольких потоках. Многопоточность актуальна для задач, нагружающих центральный процессор, для I/O нагруженных задач использовать многопоточность не имеет смысла. Все операции происходят асинхронно, понятия порядка операций не существует. Реализуя многопоточную программу разработчику следует считать количество ответов что бы узнать, когда поток закончит выполнение задачи. Не стоит создавать много потоков для каждой маленькой задачи, большое количество потоков конфликтуют за использование ресурсов процессора. Для эффективного использования вычислительных ресурсов необходимо реализовать алгоритм распределения задач между потоками. В статье используется алгоритм распределения задач между потоками «Round Robin», когда задачи распределяются между потоками по кругу. Грамотно распределив задачи между потоками, автору статьи удалось сократить время выполнения в 2,5 раза.		
<b>Позитивные следствия и/или достоинства описанной в статье технологии</b> Значительно уменьшается время выполнения. Можно создавать больше потоков, чем ядер процессора, при правильном распределении задач можно достичь минимального времени выполнения. При правильном использовании многопоточность позволяет эффективнее использовать вычислительные ресурсы.		
<b>Негативные следствия и/или недостатки описанной в статье технологии</b> Издержки на общение с потоками могут оказаться слишком большими и использование многопоточности станет невыгодным. Большое количество потоков конфликтуют между собой за вычислительные ресурсы. Потоки могут простаивать если их не успевает нагружать основной поток		
<b>Ваши замечания, пожелания преподавателю или анекдот о программистах</b>		