

Задание 2. Менеджер рабочих потоков.

Описание.

Допустим, что имеется несколько процессов, а внутри каждого из них несколько рабочих потока (далее “воркер”), которые выполняют одинаковую абстрактную работу. Во время своей работы воркер может иметь разные состояния. Все воркеры работают независимо друг от друга.

Допустим, что имеется отдельный процесс “менеджер”, который следит за всеми процессами и их воркерами.

Возможности менеджера.

1. Запуск дополнительного процесса с нужным количеством воркеров.
2. Посылка команды процессу для мягкого окончания работы(при этом воркеры должны сначала полностью закончить текущую итерацию работы)
3. Посылка команды процессу для жесткого окончания работы(при этом процесс завершает работу принудительно)
4. В каждый момент времени обладает информацией о текущем состоянии воркера.

Все вышесказанные операции и информация о состоянии воркеров и процессов, доступны через web-интерфейс или интерфейс командной строки. (выбрать один по желанию)

Задача.

Нужно реализовать менеджер.

В интерфейсе, должно быть максимально понятно отображено следующее:

1. информация о запущенных процессах.
2. информация о запущенных воркерах.
3. информация о текущем состоянии каждого воркера.
4. кнопка для мягкой остановки процесса.
5. кнопка для жесткой остановки процесса.
6. кнопка для запуска еще одного процесса.

Интерфейс должно быть понятно и легко использовать.

Интерфейс должен быть в меру интерактивный.

Допущения.

1. Для простоты, пусть каждый воркер имеет по 5 состояний, между которыми он переходит по кругу через случайное время от 1 до 10 секунд.
2. Процесс может принадлежать только одному менеджеру.
3. Все происходит в пределах одной системы.
4. На одной системе может быть только один менеджер.
5. Можно использовать какие угодно библиотеки, только надо обосновать их использование.

Требования.

1. Пишите максимально понятный код. Нужно чтобы человек прочитавший это задание, мог легко разобраться в Вашем коде.
2. Адекватные комментарии в коде крайне желательны. Комментировать каждую строку не надо, только если очень сложный кусок. Комментируйте смысл, а не операции.
3. Лучше делать тесты, чем не делать.
4. Используйте Ruby 1.9.3-p280.
5. Давайте изначально конкретный комментарий о том, что нужно сделать, чтобы проверить Ваш вариант на работоспособность.