

Анализ разработанных задач в системе автоматической проверки для MOOC "Программирование модулей ядра Linux"

Канушин Максим

19 декабря 2017 г.

1 Аннотация

В данной работе исследуется статистика прохождения студентами массового открытого онлайн курса «Программирование модулей ядра Linux». Проверка заданий, присланных студентами проходит в автоматическом режиме разработанной ранее системой. Задания представляют собой лабораторные работы, в рамках которых студенту необходимо разработать программу на языке C и Makefile. Целью исследования является определить насколько хорошо разработанные задачи и система их проверки соответствуют видео-лекциям.

2 Введение

3 Описание проверяющей системы

3.1 Характеристики системы

Система автоматической проверки лабораторных работ представляет из себя комплекс взаимосвязанных программных средств, позволяющих студенту загрузить разработанную им программу и получить ответ о ее корректности, а так же логи сборки, запуска и информацию о произошедших ошибках при их наличии.

Система обеспечивает:

- Получение решения от студента

- Проверку корректности формата решения
- Сборку решения
- Запуск решения
- Проверку работы решения
- Отправку результатов решения обратно студенту

Система обеспечивает изолированность проверяемого решения от внешней среды, чтобы обеспечить достаточный уровень надежности и безопасности исполнения. Помимо этого исключается взаимное влияние решений друг на друга для того, чтобы не дать возможность студенту случайно или намеренно повлиять на работу другого студента или просмотреть его. Среда, в которой происходит проверка, одинакова для всех решений, чтобы обеспечить как одинаковые условия проверки различных решений, так и многократную воспроизводимость результатов проверки одного и того же решения.

3.2 Архитектура системы

Система состоит из двух связанных компонент: веб-приложения и демона.

Алгоритм работы системы состоит из нескольких этапов. При получении решения, веб-приложение сохраняет его в локальное хранилище на сервере. Демон, обнаружив, что в локальной директории появилось новое решение, в зависимости от задания выбирает соответствующий сценарий проверки и запускает виртуальную машину, на которой запускается набор скриптов, собирающих и запускающих решение, а так же проверяющего его. После окончания проверки, результаты, включая логи сборки и исполнения, выгружаются из виртуальной машины, после чего она выключается.

Полная схема проверяющей системы представлена на рис. 1.

4 Обзор результатов прохождения онлайн-курса

Онлайн курс «Программирование модулей ядра Linux» был запущен в тестовом режиме. Курс состоял из 9 практических задач. В сумме по всем задачам было получено 795 решений из которых 270 правильных.

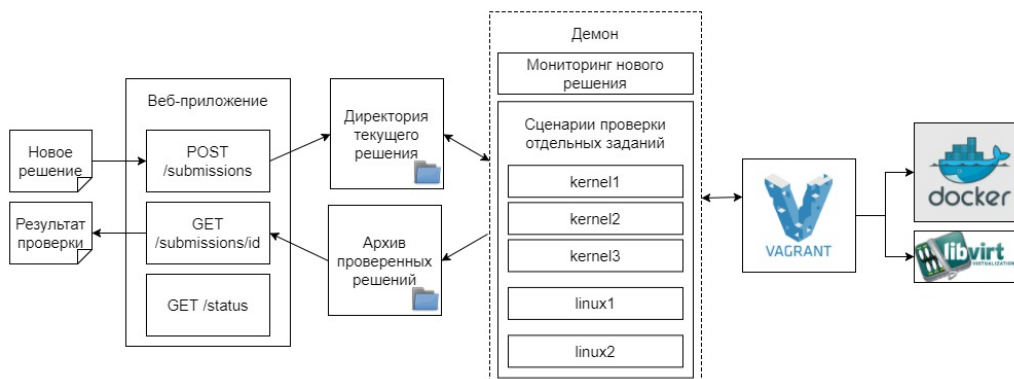


Рис. 1: Архитектура системы

4.1 Stopout

Stopout - величина, показывающая, сколько человек прекратили прохождение курса на той или иной задаче. На рис. 2 представлен график, показывающий распределение студентов, бросивших курс, по задачам до которых они добрались.

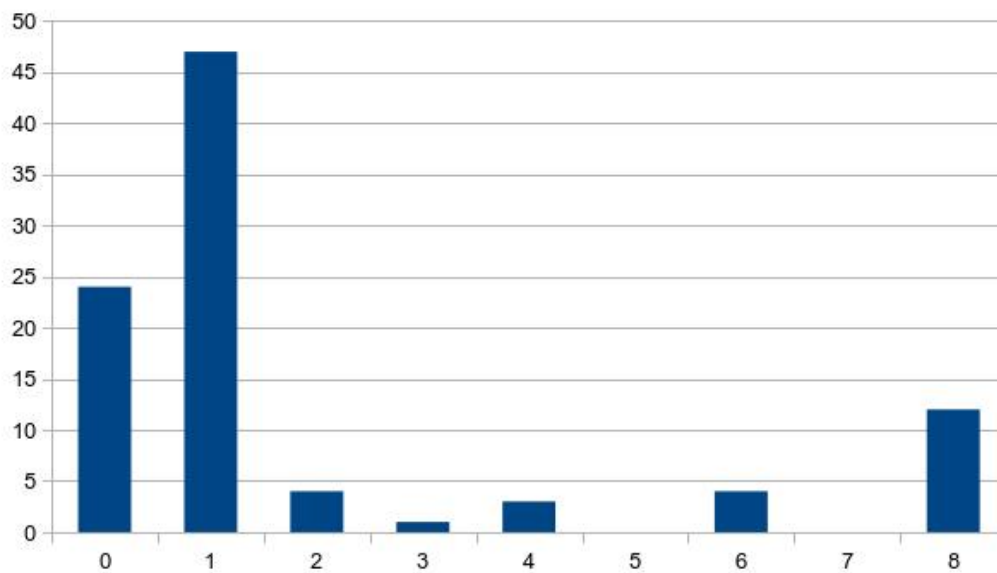


Рис. 2: Stopout

Первые два задания являются довольно простыми и схожими по сложности, что объясняет, почему многие студенты, выполнившие первое за-

дание, решили и второе.

Данный критерий был посчитан для сравнения с исследованием из статьи *Likely to stop? Predicting Stopout in Massive Open Online Courses* [1]. На рис 3 представлены его результаты на значительно более объемной выборке (154 тыс. студентов, 17.8 млн загруженных решений).

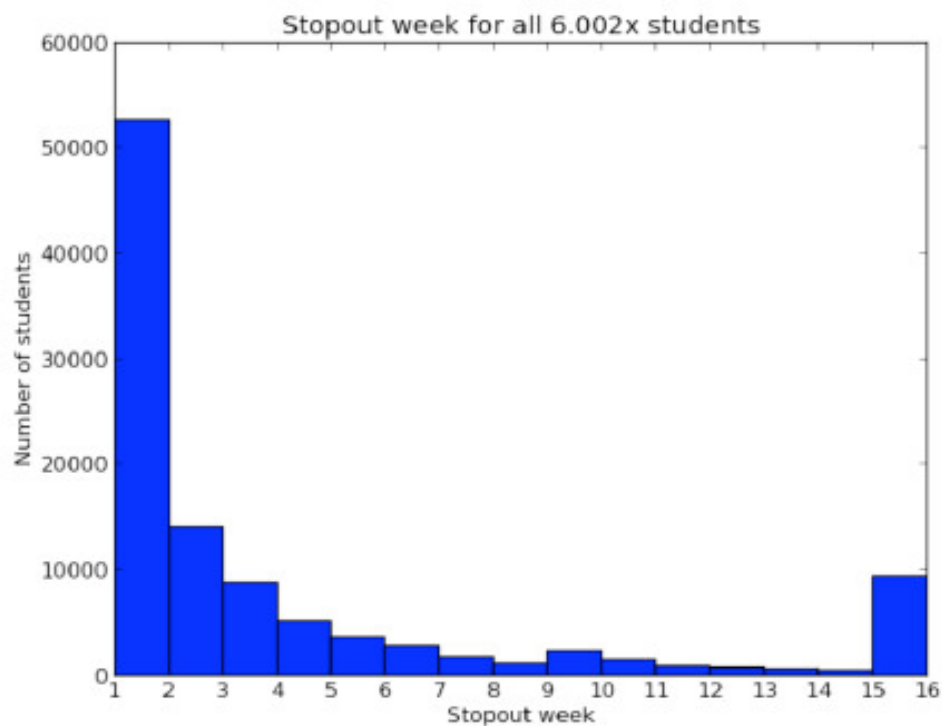


Рис. 3: Predicting Stopout in Massive Open Online Courses

4.2 Пройденные студентами этапы в рамках задач

По результатам проверки решений были подсчитаны этапы в рамках каждой задачи, которые удалось пройти студенту.

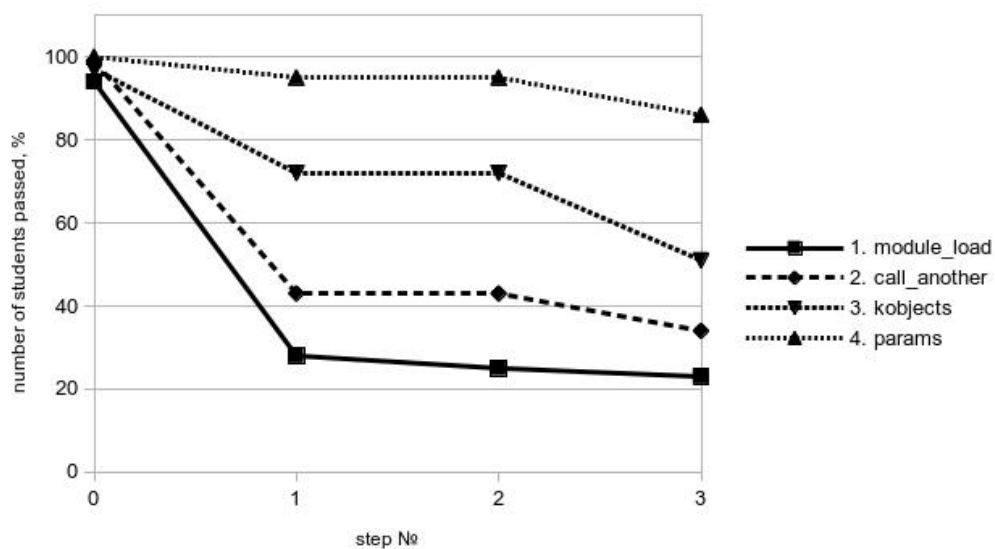


Рис. 4: Распределение студентов

5 Заключение

Список литературы

- [1] Taylor, C. and Veeramachaneni, K. and O'Reilly, U.-M. *Likely to stop? Predicting Stopout in Massive Open Online Courses*. ArXiv e-prints, 2014.