Андрей Козлов, гр. 4538

**Централизованная система распространения и сдачи домашних заданий.**

1. **Постановка задачи.**

Система позволяет упростить цикл работы над домашними заданиями в учебном процессе. Данный цикл состоит из следующих пунктов:

* преподаватель задает домашнее задание, а также устанавливает крайний срок сдачи;
* студенты выполняют домашнее задание и установленным образом передают решения преподавателю (сдают на листе бумаги/отправляют по почте и т.д.);
* преподаватель проверяет домашние задания, выставляет баллы в соответствующий документ и сообщает результаты студентам.

Система имеет два активных сервера с поддержкой синхронизации, клиенты могут подключаться к любому из серверов.

1. **Описание системы.**

Система состоит из сервера и клиентов. Пользователи делятся на две категории: «студенты» и «преподаватели». Цикл из пункта описывается следующей последовательностью действий:

* преподаватель задает домашнее задание отправляя соответствующее сообщение на сервер;
* сервер сохраняет задание и отправляет его студенту , когда он появляется в сети;
* студент выполняет задание и отправляет на сервер сообщение, содержащее решение задания ;
* сервер сохраняет решение и отправляет его преподавателю , когда он появляется в сети;
* преподаватель проверяет решение и отправляет результат проверки студенту .

1. **Служебные объекты.**
   1. FileWrapper(String fileName, byte[] content), где fileName – название файла, content – его содержание;
   2. Signature(byte[] bytes), где bytes – некоторый md5-хеш, возвращаемый клиенту при аутентификации на сервере.
2. **Модель.**

Модель состоит из следующих объектов:

* 1. Task(String courseName, String taskName, String text, long deadline), где courseName – название курса, taskName – название задания, text – текст задания, deadline – крайний срок сдачи;
  2. Solution(String courseName, String taskName, FileWrapper wrapper), где courseName – название курса, taskName – название задания, wrapper – пересылаемый файл решения;
  3. Verdict(String studentName, String taskName, boolean accepted, String comments), где studentName – имя студента, taskName – название задания, accepted – принято ли задание, comments – возможные комментарии;
  4. AuthenticationRequest(String login, int passwordHashCode), где login – логин клиента, passwordHashCode – хеш-код пароля;
  5. AuthenticationResponse(boolean passed, String name, Signature signature), где passed – пройдена ли аутентификация, name – имя клиента, signature – подпись.

Все задания, решения и вердикты подписываются полученной при аутентификации подписью. Далее будем называть их подписанными объектами.

1. **Сообщения.**

Каждый объект модели

Сообщение формируются следующим образом:

1. 4 байта – длина сообщения;
2. 1 байт – уникальный тип сообщения;
3. поля объекта модели.

Примитивные значения записываются без преобразований, остальные записываются согласно следующим правилам:

1. массив байтов: 4 байта – длина массива length, length байт – элементы массива;
2. строка выводится, как массив байтов (UTF-8);
3. файл: строка – название, массив байтов – содержимое.
4. **Протокол «клиент-сервер».**

* Клиент периодически отправляет запрос аутентификации на один из серверов;
* При получении запроса аутентификации от клиента сервер отправляет сообщение о прохождении аутентификации;
* Если аутентификация прошла успешно, то также отправляются подписанные объекты, которые поступили на сервер с момента предыдущей аутентификации данного клиента;
* Отправленные подписанные объекты помечаются, как отправленные;
* Для того, чтобы отправить подписанный объект, клиент отправляет на сервер соответствующее сообщение;
* Соответствующий подписанный объект сохраняется на сервере и пересылается клиентам при следующих подключениях.

1. **Синхронизация, протокол «сервер-сервер».**

* При поступлении на сервер нового подписанного объекта, он пересылается второму серверу и помечается, как отправленный.