1. Задачи

Здесь приведен список семестровых задач. Вам предстоит выбрать задачу, язык реализации, а также дополнительную часть. Занести данные в табличку(см. ссылку в письме), и дождаться одобрения или просьбы её некотором образом видоизменить. Цель данного действа в том, чтобы ваши задачи были максимально разнообразны. Т.о. в Ваших интересах везде, где это возможно, проявить инициативу.
Вы можете выбрать одну из нижеперечисленных задач:

- компилятор из описанного ниже языка в
  - ARM;
  - ASM x86;
  - LLVM;
  - JVM;
  - Собственный низкоуровневый язык (+ его интерпретатор);
  - Другое (если у Вас имеются идеи/пожелания, пишите — попробуем согласовать).
- JIT-compiler;
- Bootstrap-compiler;
- Нечто другое (например, самоприменимый специализатор, абстрактный интерпретатор и т.д.).

2. Общая часть

То, что содержится в данной части, является обязательным для реализации всеми, вне завимости от выбранного проекта и языка (буть может кроме некоторых исключительных случаев). Обратите внимаение на то, что при выборе языка вам необходимо помимо синтаксиса некоторым образом описать его семантику (в т.ч. парадигму(-ы))

2.1. Язык L: синтаксис

Здесь приводится некоторое описание синтаксиса языка. Для начала Вам предстоит придумать конкретный синтаксис, его формально описать, описать его семантику и прислать всё это дело в виде pdf файла.

1. Переменные: локальные(в пределах блока) и глобальные;
2. Функции (с рекурсией);
3. Примитивные типы $\mathbb{T} = Integer \mid Bool \mid String$, массивы, объединения и некоторые конструкторы типов, позволяющие создавать новые типы;
4. Арифметические выражения;
5. Логические выражения;
6. Операторы S;
   (a) Пустой оператор;
   (b) Присваивание; $\mathbb{V} := E$
   (c) $\sim$ Comma operator; (ex: $S; S$)
   (d) Операторы ветвления;
   (e) Операторы цикла;
   (f) Операторы $switch$, $continue$ и $break$;
   (g) Оператор чтения $read(\mathbb{V})$;
   (h) Оператор записи $write(E)$;
   (i) Вызов функции;
   (j) Комментарии.
7. Блоки $\mathbb{B}$;
8. Объявления функций;
9. extern;
10. Программы P.

Пример: функция быстрого возведение в степень на языке L в некотором конкретном синтаксисе:

```
void power () {
     int k, n;
     read(k);
     read(n);
     r = 1;
     while k > 0 do
          if k % 2 == 1
          then r = r * n
          else skip;
```

```
          n = n * n;
          k = k / 2
     done;
     write(r)
}
```

2.2. Оптимизации

Вам также предстоит выбрать некоторый набор оптимизаций, которые Вы будете производить в Вашем компиляторе. Это будет необходимо сделать после лекций по соответствующим темам.

3. Дополнительная часть

Добавить в язык что-то из нижеследующего:

- Потоки;
- Классы (с наследованием);
- Функции высших порядков (с замыканиями);
- Другое (если у Вас имеются идеи/пожелания, пишите — попробуем согласовать).

P.S.

Напоминаю:

1. Задачи, как я уже отмечал, вполне себе могут быть изначально не равнозначны; в ходе Вашей работы они могут быть немного скорректированы с нашей стороны, дабы они не выраждались и не становились чересчур сложными ;
2. Письма пишутся на почту ”daniil.berezun@gmail.com”, тема письма должна начинаться с набора символов ”[АУ-Компиляторы]” и далее тема Вашего письма;
3. К 1 октября у Вас должны быть готовы некоторые промежуточные результаты, которые Вы должны быть готовы обсудить с нами воочию, если потребуется; естественно, Вам следует

прислать письмо с описанием того, чего Вы уже добились несколько заранее (например, за 2 дня);

4. К 15 октября у Вас должен быть написан некоторый рабочий front-end (т.е. генерироваться промежуточное представление)
5. Дальнейшие сроки будут известны чуть позже;
6. Не стоит тянуть кота за хвост: мы всегда готовы к общению, поэтому пишите/спрашивайте и т.п. заранее, а не в последний момент.