

tiger compiler

Lab4

- gc 中使用了 Linux C 的库函数，测试时需要在 Linux 系统中进行；
- 命令行参数的格式如下：

```
./a.out @tiger -heapSize 1024 @  
./a.out @tiger -logGc @
```

- gc_log.txt 中记录的是收集的垃圾大小，即from 中已分配空间的大小减去复制到to中的大小；原来的测试用例如果发生了GC，往往是把from完全复制到to中，收集到的垃圾为0 Bytes. test目录下的testGC.java可以作为很好的测试用例：

```
./a.out @tiger -heapSize 500 -logGc @
```

- 求数组长度在lib.c中定义为库函数,声明为:

```
int len(char *array);
```

Enjoy!

Lab3

1. 由于 glue code 中使用了 Linux 中的指令,测试时需要在Linux系统中进行.
2. 若系统为Linux,默认设置为直接编译运行,输入:

```
$java Tiger inputfile -codegen C  
$java Tiger inputfile -codegen bytecode
```

时,会直接执行 link 和 run 步骤,输出在控制台.

1. 如果您的系统不是 Linux,需要手动编译,运行生成的代码.
2. Tiger_new_array 与老师提供方法不同,我们的实现方式是通过库函数调用新建数组,求长度等.

Lab2

由于我们采用的语法与MiniJava不太一样，为了方便您更好地理解代码，对程序做如下说明：

1. statement 语句 与 vardecl 可以混合在一起,写程序时可以随处定义变量；
2. 变量可以先使用后声明:

```
i = 1;  
int i;
```

在程序中是合法的，因为在生成 ast 时会先收集 decs 信息，然后处理 stms；

1. 在 PrettyPrintVisitor 时，仍会把所有声明打印在所有语句之前，即内部的语法解释和 MiniJava 相同，只是在写的时候增加了程序员的自由度；
2. 为了处理方便，定义 ast.DecOrStm 类作为 ast.dec.T 和 ast.stm.T 的抽象父类；
3. 把 Block 作为重要结构处理:

```
Block -> { statements | vardecls }
```

1. 定义 BlockTable 类，作为每个 Block 的符号表；
2. MainClass 里面的 main 重新定义为：

```
main -> public static void main(String[] a) Block
```

1. 测试 Sum 时和 Fac 类似:

```
java Tiger -testSum
```
