

## 目标代码生成实验报告

211220079 谷石磊

本次实验完成了中间代码向目标代码（MIPS32）生成的工作，其中设计的栈帧如下：

```
*
* high address
* +-----+
* | arg2 | <- $fp + 12
* +-----+
* | arg1 | <- $fp + 8
* +-----+
* | ret   | <- $fp + 4 (return address)
* +-----+
* | old $fp | <- $fp (old frame pointer)
* +-----+
* | var1   | <- $fp - 4
* +-----+
* | var2   | <- $fp - 8
* +-----+
* | ...    |
* | ...    |
* +-----+
* | ...    | <- $sp (stack pointer)
* +-----+
* Low address
*
```

传参时所有参数都位于栈中。使用了朴素的寄存器分配算法，针对每一条中间代码进行指令的翻译。

为了编译该程序，在文件根目录下，使用指令 `cd Code && make` 即可生成可执行文件 `parser`，之后使用 `./parser filename [as_file] [ir_file]`，即可运行程序，在 `[as_file]` 生成对应的 mips32 代码，若给第三个参数 `[ir_file]`，则会在 `[ir_file]` 生成对应的中间代码。