

编译原理 Project4 实验报告

高睿齐 161250026

2019 年 6 月 9 日

1 完成的功能点

本次实验我完成了所有必做部分，使用朴素寄存器分配算法

2 实现方法

本次实验我实现了朴素寄存器分配算法，在每个函数的汇编代码生成阶段开始时，首先扫描一边函数体记录记录该函数体内部所有变量的内存位置，因为所有变量均储存在函数的活动记录中，所以只需记录所有变量相对当前函数当前函数活动记录的栈顶的相对位置。在本次实验中，我使用了 fp 寄存器来对栈内的变量寻址，在函数体开始时保存返回地址寄存器 ra,fp 寄存器，然后分配栈帧，为 fp 寄存器赋值。当记录当前函数体变量的位置时，第一个变量偏移量为 $-(8 + \text{size})$ (相对于 fp 寄存器)，因为 $-4(\$fp)$ 用来储存 ra 寄存器 $-8(\$fp)$ 用来储存旧 fp 寄存器，后续变量顺序递减偏移量（栈从高地址向低地址生长），最后一个变量的偏移量即位当前函数的栈帧大小。

对于函数传递参数部分，前四个参数存入 a0,a1,a2,a3 寄存器，其他参数依次压入栈中，函数栈帧示意图如下：

	\dots
	$param_6$
$fp \rightarrow$	$param_5$
	ra
	$old\ fp$
$sp \rightarrow$	\dots $local\ variable$ \dots

3 编译方法

在./Code 文件夹下运行 make 指令编译, 在./文件夹下生成可执行文件 parser

4 实验总结

通过本次实验, 我加深了对 mips32 代码生成的理解