

CS323 Project4

Target Code Generation

Team Members: Li Yuanzhao(11812420), Xu Xinyu(11811536), Jiang Yuchen(11812419)

Contribution: A, A, A

I. Overview

In this project, we are required to implement the generation of target code of given intermediate representation(IR) code on SPL language. We suppose that IR codes are correct so that we can directly output mips32 instruction files. Our files can be run successfully with GCC version 7.4.0, GNU Flex version 2.6.4 and GNU Bison version 3.0.4 .

II. Design and Implementation

There is no further design on project files. The main task is to finish the incomplete methods in `mips32.cpp` . We first change them to `.cpp` form in order to use the data structure designed only in `.cpp` . Then we construct a map which stores the pair of variable name and its register name. In `struct RegDesc` , we take use of `dirty` to represent which the register is using or not and it's 0 by default.

After that, we try to implement the incomplete methods given in `mips32.cpp` . We search the register list to find the unused register and load word to it for other operations. Specially, if we want to get a register for write purpose, there is no need to load word since the content is unknown. For spilling register, we will spill the used register and store its value so that we can reuse the register safely for other operations. When facing function, `varMap` , `offsetCnt` and `arg_num` are reset for function block. We will store registers and other status information before calling function. The rest parts are easy to translate according to mips32 instructions.

```
void spill_register(Register reg){
    /* COMPLETE the register spilling */

    if(varMap.count(_Keyval: regs[reg].var)==0){
        struct VarDesc newVar;
        newVar.offset = offsetCnt*4;
        offsetCnt++;
        varMap[regs[reg].var] = newVar;

        _mips_iprintf( fmt: "sw %s, -%d($fp)", _reg_name(reg), newVar.offset);
        _mips_iprintf( fmt: "addi $sp, $fp, -%d", newVar.offset);
    }else{
        _mips_iprintf( fmt: "sw %s, -%d($fp)", _reg_name(reg), varMap[regs[reg].var].offset);
    }

    regs[reg].dirty = FALSE;
}
```

Figure.1 spill_register()

III. Test Cases

For five test cases, we generate the target mips32 code for them and all of them can be executed successfully and correctly.

```
charlotte@ubuntu:~/cs323-proj4$ bin/splc test/test_4_r01.ir > out.s
charlotte@ubuntu:~/cs323-proj4$ spim -file out.s
SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: /usr/lib/spim/exceptions.s
10003
10002
30002
10003
20001
20003
10003
```

Figure.2 test_01

```
charlotte@ubuntu:~/cs323-proj4$ bin/splc test/test_4_r02.ir > out.s
charlotte@ubuntu:~/cs323-proj4$ spim -file out.s
SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: /usr/lib/spim/exceptions.s
Enter an integer: 3
1
charlotte@ubuntu:~/cs323-proj4$ spim -file out.s
SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: /usr/lib/spim/exceptions.s
Enter an integer: -2
-1
charlotte@ubuntu:~/cs323-proj4$ spim -file out.s
SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: /usr/lib/spim/exceptions.s
Enter an integer: 0
0
```

Figure.3 test_02

```
charlotte@ubuntu:~/cs323-proj4$ bin/splc test/test_4_r03.ir > out.s
charlotte@ubuntu:~/cs323-proj4$ spim -file out.s
SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: /usr/lib/spim/exceptions.s
19
```

Figure.4 test_03

```
charlotte@ubuntu:~/cs323-proj4$ bin/sp1c sample/test_4_fact.ir > out.s
charlotte@ubuntu:~/cs323-proj4$ spim -file out.s
SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: /usr/lib/spim/exceptions.s
Enter an integer: 3
6
charlotte@ubuntu:~/cs323-proj4$ spim -file out.s
SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: /usr/lib/spim/exceptions.s
Enter an integer: 5
120
charlotte@ubuntu:~/cs323-proj4$ spim -file out.s
SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: /usr/lib/spim/exceptions.s
Enter an integer: 8
40320
```

Figure.5 test_04

IV. Instructions

Change directory to the root path and using `make sp1c` to create `sp1c` in `./bin` root for spl codes' parsing. Then using `bin/sp1c <test_root>/<test_file_name>` to create mips32 code result. And you can use `make clean` to delete all created files.