

1. 正文第 11 页:

- 例 1.10 输入代码处, 第 6 行 (即“nested comment \*/”) 去除最右端的“\*/”。

2. 正文第 31 页:

- 1.2.11 节内 YY\_USER\_ACTION 代码处, 第 1 行前需添加 1 行“%locations”与 1 行“...”, 第 6 行 (即“#define YY\_USER\_ACTION”) 最后添加“\”。修改后的完整代码如下 (可直接替换原先代码):

```
1 %locations
2 ...
3 %{
4  /* 此处省略#include部分 */
5  int yycolumn = 1;
6  #define YY_USER_ACTION \
7    yyloc.first_line = yyloc.last_line = yylineno; \
8    yyloc.first_column = yycolumn; \
9    yyloc.last_column = yycolumn + yyleng - 1; \
10   yycolumn += yyleng;
11 %}
```

3. 正文第 39 页:

- 2.1.1 节错误类型 7 处, “或数组 (或结构体) 变量与数组 (或结构体) 结构体变量相加减”修改为“或数组 (或结构体) 变量与数组 (或结构体) 变量相加减”。

4. 正文第 39 页:

- 2.1.1 节当页底部, “注意两点”修改为“注意三点”。

5. 正文第 40 页:

- 2.1.1 节当页顶部, “已经填满)。”修改为“已经填满); 三是对于结构体类型等价

的判定，每个匿名的结构体类型我们认为均具有一个独有的隐藏名字，以此进行名等价判定。”。

6. 正文第 41 页：

- 2.1.4 节第一段处，“程序将在如下环境中被编译运行”修改为“程序将在如下环境中被编译运行（同实验一）”。

7. 正文第 64 页：

- 假设 7 处，“输入文件中不包含任何词法、语法或语义错误”修改为“输入文件中不包含任何词法、语法或语义错误（函数也必有 return 语句）”。

8. 正文第 65 页：

- 第 10) 点处，末尾添加如下语句：“注意：当函数参数是结构体或数组时，ARG 语句的参数为结构体或数组的地址（即以传引用的方式实现函数参数传递）。”。

9. 正文第 67 页：

- 3.1.4 节第一段处，“程序将在如下环境中被编译运行”修改为“程序将在如下环境中被编译运行（同实验一）”。

10. 正文第 68 页：

- 3.1.6 节例 3.1 处，“然后计算并输出符号函数 `sgn(x)`”修改为“然后计算并输出符号函数 `sgn(n)`”。
- 3.1.6 节例 3.1 处，“使用的变量遵循着字母 `t` 后跟一个数字”修改为“使用的变量遵循着字母后跟一个数字”。

11. 正文第 85 页：

- 表 3-5 处，“`if (function.name == "write") return code1 + [WRITE arg_list[1]]`”之后添加“`+ [place := #0]`”。

12. 正文第 89 页：

- 假设 1 处，“输入文件中不包含任何词法、语法或语义错误”修改为“输入文件中不包含任何词法、语法或语义错误（函数也必有 return 语句）”。
- 假设 3 处，“你不必考虑如果某个常数”修改为“你不必考虑如果某个整型常数”。
- 假设 5 处，“所有的变量均不重名”修改为“没有全局变量的使用，并且所有变量均不重名”。
- 4.1.2 节第一段处，“输出文件名为 out.s”修改为“输出文件名为 out1.s”。
- 4.1.2 节第一段处，“那么在 Linux 命令行下运行 ./cc test.cmm out.s 即可将输出结果写入当前目录下名为 out.s 的文件中”修改为“那么在 Linux 命令行下运行 ./cc test1.cmm out1.s 即可将输出结果写入当前目录下名为 out1.s 的文件中”。

13. 正文第 90 页：

- 4.1.4 节第一段处，“程序将在如下环境中被编译运行”修改为“程序将在如下环境中被编译运行（同实验一）”。

14. 正文第 98 页：

- 表 4-2（即常用的伪指令表）处，第一条伪指令的“描述”与“对应的 MIPS32 指令”部分，以及第二条伪指令的“伪指令”和“对应的 MIPS32 指令”部分修改为如下内容（阴影部分）：

伪指令	描述	对应的 MIPS32 指令
li Rdest, imm	把立即数 imm（小于等于 0xffff）加载到寄存器 Rdest 中。	ori Rdest, \$0, imm
	把立即数 imm（大于 0xffff）加载到寄存器 Rdest 中。	lui Rdest, upper(imm) ori Rdest, Rdest, lower(imm)
la Rdest, addr	把地址（而非其中的内容）加载到寄存器 Rdest 中。	lui Rdest, upper(addr) ori Rdest, Rdest, lower(addr)

- 同时为表格中的 upper(imm)添加脚注：“表中包含的 upper 和 lower 指令并非真

实的 MIPS32 指令, `upper(num)`表示取一个 32 位整数 `num` 的第 16–31 位, `lower(num)`表示取一个 32 位整数 `num` 的第 0–15 位。”。

15. 正文第 111 页:

- 中间代码处, 第 4 行和第 12 行中的“`n - 5`”修改为“`n - 4`”。

16. 正文第 118 页:

- A.2.2 第 3)点之 a)处, “例如 `int global1, global2.`”修改为“例如“`int global1, global2;`”。”。
- A.2.2 第 3)点之 b)处, “例如 `struct {...}`。”修改为“例如“`struct {...};`”。”。

17. 正文第 120 页:

- A.2.6 第 1)点处, “它就是一串像 `int a; float b, c; int d[10];` 这样的变量定义”修改为“它就是一串像 `int a; float b, c; int d[10];`这样的变量定义”。
- A.2.6 第 2)点处, “它包括一个类型描述符 `Speicifier` 以及一个 `DecList`”修改为“它包括一个类型描述符 `Specifier` 以及一个 `DecList`”, “例如 `int a = 5.`”修改为“例如 `int a = 5;`”。