

- 包诗峰 学号：2022211656 班级：2022211301

风格小作业

- 问题一

- 代码来源：fifo_queue.c, line 5-15
- 问题代码：

```
struct fifo_queue_item
{
    int item_value;
    struct fifo_queue_item* next_item;
};

struct fifo_queue
{
    struct fifo_queue_item* queue_head;
    struct fifo_queue_item* queue_tail;
};
```

- 存在问题：

item_value、next_item、queue_head、queue_tail 命名重复啰嗦，更好地命名方式应为 value、next、head、tail

- 修改结果：

```
struct fifo_queue_item
{
    int value;
    struct fifo_queue_item* next;
};

struct fifo_queue
{
    struct fifo_queue_item* head;
    struct fifo_queue_item* tail;
};
```

- 问题二：

- 代码来源：fifo_queue.c, line 33-51
- 问题代码：

```
// pop a item_value from queue front
int fifo_queue_pop_front(struct fifo_queue* queue, int* value)
{
    if(queue->queue_head != NULL)
    {
        struct fifo_queue_item* item = queue->queue_head;
        if(item->next_item == NULL)
        {
```

```

        queue->queue_head = NULL;
        queue->queue_tail = NULL;
    }
    else
        queue->queue_head = item->next_item;
    *value = item->item_value;
    free(item);
    return 1;
}
else
    return 0;
}

```

- 存在问题：
if语句过于复杂，降低可读性，可以化简，改成if-else语句
- 修改结果：

```

int fifo_queue_pop_front(struct fifo_queue* queue, int* value)
{
    if(queue->queue_head == NULL)
        return 0;
    struct fifo_queue_item* item = queue->queue_head;
    if(item->next_item != NULL)
        queue->queue_head = item->next_item;
    else
    {
        queue->queue_head = NULL;
        queue->queue_tail = NULL;
    }
    *value = item->item_value;
    free(item);
    return 1;
}

```

- 问题三：
 - 代码来源：fifo_queue.c, line 59、64
 - 问题代码：

```
for(i = 0; i < 10; i++)
```

```
for(i = 0; i < 11; i++)
```

- 存在问题：
10和11是magic number（神秘数），可以通过补充宏定义的方式修改
- 修改结果：

```
#define LOOP_SIZE 10
```

```
for(i = 0; i < LOOP_SIZE; i++)
```

```
for(i = 0; i < LOOP_SIZE + 1; i++)
```

- 问题四:

- 代码来源: fifo_queue.c, line 67-71

- 问题代码:

```
int success = fifo_queue_pop_front(&queue, &value);
if(success)
    printf("pop from front: %d\n", value);
else
    printf("pop from front fail\n");
```

- 存在问题:

success多余, 可直接将 `fifo_queue_pop_front(&queue, &value)` 作为判断

- 修改结果:

```
if(fifo_queue_pop_front(&queue, &value))
    printf("pop from front: %d\n", value);
else
    printf("pop from front fail\n");
```

- 问题五:

- 代码来源: serialize_json.cpp, line 22

- 问题代码:

```
if (!('\x00' <= *c && *c <= '\x1f'))
```

- 存在问题:

if判断较为复杂, 反直觉

- 修改结果:

```
if (c < '\x00' || *c > '\x1f')
```

- 问题六:

- 代码来源: serialize_json.cpp, line 21-25

- 问题代码:

```
default:
    if (!('\x00' <= *c && *c <= '\x1f'))
        o << *c;
    else
        o << "\\u" << hex << setw(4) << setfill('0') << (int)*c;
```

- 存在问题:

default分支没有加break语句

- 修改结果:

```
default:
    if (!('\x00' <= *c && *c <= '\x1f'))
        o << *c;
    else
        o << "\\u" << hex << setw(4) << setfill('0') << (int)*c;
    break;
```

- 问题七:

- 代码来源: serialize_json.cpp, line 37
- 问题代码:

```
bool not_first = false;
for(i = m.begin(); i != m.end(); i++){
    if(not_first)
        o << ",";
    else
        not_first = true;
    o << "\"" << escape_json(i->first) << "":"\" << escape_json(i->second)
    << "\"";
}
```

- 存在问题:

判断变量 `not_first` 存在否定之否定, 不符合阅读习惯, 应该改成 `is_first` 符合直觉。

- 修改结果:

```
bool is_first = true;
for(i = m.begin(); i != m.end(); i++){
    if(is_first)
        is_first = false;
    else
        o << ",";
    o << "\"" << escape_json(i->first) << "":"\" << escape_json(i->second)
    << "\"";
}
```

- 问题八:

- 代码来源: split_str.cpp, line 8
- 问题代码:

```
int process_str(char* argv[], const char* buffer)
```

- 存在问题:

函数名功能不明确, 应当修改成能表达函数功能的名字, 如 `split_str`

- 修改结果:

```
int split_str(char* argv[], const char* buffer)
```

- 问题九:

- 代码来源: split_str.cpp, line 47-50
- 问题代码:

```
char* argv[30];
int i;
for(i = 0; i < 30; i++)
    argv[i] = new char[32];
```

- 存在问题:
30和32还是magic number, 可以通过补充宏定义的方式修改
- 修改结果:

```
#define STR_NUMBER 30
#define STR_LENGTH 32
```

```
char* argv[STR_NUMBER];
int i;
for(i = 0; i < STR_NUMBER; i++)
    argv[i] = new char[STR_LENGTH];
```

- 问题十:
 - 代码来源: split_str.cpp, line 49-53
 - 问题代码:

```
for(i = 0; i < 30; i++)
    argv[i] = new char[32];
int argc = process_str(argv, " kjsf ks ks dhf ksdjh ksdjfh skdjf skdf
skdjf sdkjf kjsdhf ");
for(i = 0; i < argc; i++)
    printf("argv[%d] = %s\n", i, argv[i]);
```

- 存在问题:
argv在new了char数组之后没有delete,需要在使用结束之后delete。
- 修改结果:

```
for(i = 0; i < 30; i++)
    argv[i] = new char[32];
int argc = process_str(argv, " kjsf ks ks dhf ksdjh ksdjfh skdjf skdf
skdjf sdkjf kjsdhf ");
for(i = 0; i < argc; i++)
    printf("argv[%d] = %s\n", i, argv[i]);
for(i = 0; i < STRING_NUMBER; i++)
    delete [] argv[i];
```