BOSC 手册模版

A Manual Template for BOSC

BOSC Team 编著
BOSC Team 审校

Note: This is a note

文档状态：草稿/已审校/已冻结/已发布
最近修改时间：2024/01/19

北京

BEIJING

第一版序

这是一个用于 BOSC 手册的 LaTeX 模版

目　录

第一版序 ii

第一部分　基础使用 1

第一章 封面设置 2

1.1 环境配置 2
1.2 文档结构 2
1.3 封面布局 2
1.4 序言 2
参考文献 2

第二章 插入图片、表格和代码 3

2.1 插入图片 3
2.2 插图表格 3
2.3 插入代码 3
参考文献 4

附录 5

插图目录

表格目录

第一部分

基础使用

第一章

封面设置

1.1 环境配置

本文档需要 XeLaTex 编译器

1.2 文档结构

1.3 封面布局

1.4 序言

成功引用了吗 [**?**]

参考文献

[1] S. V. Adve and K. Gharachorloo, “Shared memory consistency models: A tutorial,” *computer*, vol. 29, no. 12, pp. 66–76, 1996.

第二章

插入图片、表格和代码

2.1 插入图片

2.2 插图表格

- 小表格，使用基本三线表格
- 大表格

2.3 插入代码

- 大块代码插入

代码 2.1: **test.cpp**

```
1
2  #include <iostream>
3  #include <boost/asio.hpp>
4
5  using namespace boost::asio;
6  using ip::tcp;
7
8  class TCPServer {
9  public:
10     TCPServer(io_service& io_service, short port)
11         : acceptor_(io_service, tcp::endpoint(tcp::v4(), port)),
12           socket_(io_service) {
13         startAccept();
14     }
15
16 private:
17     void startAccept() {
18         acceptor_.async_accept(socket_,
19             [this](boost::system::error_code ec) {
20                 if (!ec) {
21                     std::cout << "Client connected." << std::endl;
22                     handleRequest();
23                 }
24                 startAccept();
25             });
26     }
27
28     void handleRequest() {
29         async_read_until(socket_, buffer_, '\n',
```

```
            [this](boost::system::error_code ec, std::size_t /*length*/) {
                if (!ec) {
                    std::istream is(&buffer_);
                    std::string message;
                    std::getline(is, message);
                    std::cout << "Received: " << message << std::endl;

                    // Handle the message as needed

                    handleRequest(); // Continue listening for the next request
                }
            });
    }

    tcp::acceptor acceptor_;
    tcp::socket socket_;
    streambuf buffer_;
};

int main() {
    try {
        boost::asio::io_service io_service;
        TCPServer server(io_service, 1234);
        io_service.run();
    } catch (std::exception& e) {
        std::cerr << "Exception: " << e.what() << std::endl;
    }

    return 0;
}
```

- 少量代码插入

```
    def factorial(n):
    """Calculate the factorial of a number."""
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

print("Factorial of 5:", factorial(5))
```

参考文献

附录