|  |
| --- |
| 1559012095(1)  服务器核心架构设计项目  软件需求规格说明书 |
| **项目名称：\_服务器核心架构设计（C++实现）**  **项目编号： 00001**  **编写人员： （12组）郭乐晨\_\_\_\_**  **编写日期： 2019.6.25 \_**  **审批人员：**  **审批日期：**  修订历史记录   |  |  |  |  | | --- | --- | --- | --- | | **日期** | **版本** | **说明** | **作者** | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |
|  |

**目 录**

[1. 引言 1](#_Toc479181408)

[1.1 目的 1](#_Toc479181409)

[1.2 文档格式 1](#_Toc479181410)

[1.3 预期的读者和阅读建议 1](#_Toc479181411)

[1.4 术语[可选] 2](#_Toc479181412)

[2. 系统概述 2](#_Toc479181413)

[2.1 概述 2](#_Toc479181414)

[2.2 功能 2](#_Toc479181415)

[2.3 运行环境 2](#_Toc479181416)

[3. 需求描述 3](#_Toc479181417)

[3.1 用户角色 3](#_Toc479181418)

[3.2 服务器架构 4](#_Toc479181419)

[3.3 技术需求 5](#_Toc479181419)

[3.4 功能需求 5](#_Toc479181419)

[4. 非功能性需求 7](#_Toc479181424)

[4.1 性能需求 7](#_Toc479181425)

[4.2 安全性需求 7](#_Toc479181426)

[4.3 可用性需求 7](#_Toc479181427)

[4.4 用户文档 7](#_Toc479181428)

[4.5 其它需求 8](#_Toc479181429)

[5. 外部接口需求 8](#_Toc479181430)

[5.1 用户接口 8](#_Toc479181431)

[5.2 硬件接口 8](#_Toc479181432)

[5.3 软件接口 8](#_Toc479181433)

[5.4 通信接口 8](#_Toc479181434)

# 1. 引言

## 1.1 目的

该文档首先给出了整个服务器的整体功能结构的概貌，试图从总体架构上给出服务器的轮廓，然后又对功能需求、性能需求和其它非功能性需求进行了详细的描述。本文档能详细准确地描述用户的需求，同时也为用户更容易地理解这些需求的描述创造了条件。

该文档详尽说明了这一软件产品的需求和规格，这些规格说明是进行设计的基础，也是编写测试用例和进行系统测试的主要依据。同时，该文档也是用户确定软件功能需求的主要依据。

## 1.2 文档格式

本文档按以下要求和约定进行书写：

（1）页面的左边距为2.5cm，右边距为2.0cm，装订线靠左，行距为最小值20磅。

（2）标题最多分三级，分别为黑体小三、黑体四号、黑体小四，标题均加粗。

（3）正文字体为宋体小四号，无特殊情况下，字体颜色均采用黑色。

（4）出现序号的段落不采用自动编号功能而采用人工编号，各级别的序号依次为（1）、1）、a)等，特殊情况另作规定。

## 1.3 预期的读者和阅读建议

本文档的主要内容共分4部分：综合描述、系统特性、和非功能性需求和外部接口描述。综合描述部分主要对系统的整体结构进行了大致的介绍；系统特性部分对系统的功能需求进行了详细描述，是本文的主要部分；非功能性需求部分对非功能需求进行了详细的描述；外部接口需求部分对用户界面、软件接口、硬件接口和通讯接口等进行了描述。

本文档面向多种读者对象：

（1）设计员：对需求进行分析，并设计出服务器。

（2）程序员：配合《设计报告》，了解服务器功能。

（3）用户：了解预期产品的功能，并与分析人员一起对整个需求进行讨论和协商。

（4）其他人员：如部门领导、公司领导等可以据此了解产品的功能和性能。

在阅读本文档时，首先要了解产品的功能概貌，然后可以根据自身的需要对每一功能进行适当的了解。

## 1.4 术语

Makefile:一个工程中的源文件不计其数，其按类型、功能、模块分别放在若干个目录中，makefile定义了一系列的规则来指定，哪些文件需要先编译，哪些文件需要后编译，哪些文件需要重新编译，还可以进行更复杂的功能操作

Epoll: Linux内核为处理大批量文件描述符而作了改进的poll，是Linux下多路复用IO接口select/poll的增强版本，它能显著提高程序在大量并发连接中只有少量活跃的情况下的系统CPU利用率

线程池：一种多线程处理形式，处理过程中将任务添加到队列，然后在创建线程后自动启动这些任务

线程同步：当有一个线程在对内存进行操作时，其他线程都不可以对这个内存地址进行操作，直到该线程完成操作， 其他线程才能对该内存地址进行操作

连接池：创建和管理一个连接的缓冲池的技术，这些连接准备好被任何需要它们的线程使用。

# 2. 系统概述

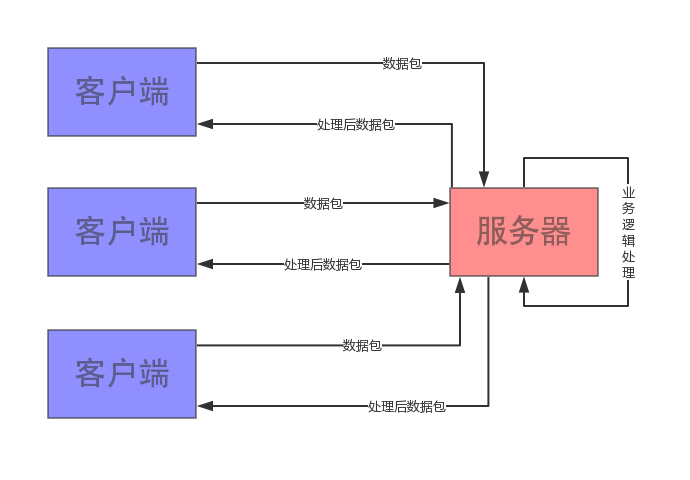
## 2.1 概述

无论是浏览器、应用程序、游戏，任何客户端都需要有相应的服务器接受请求处理相应数据并返回结果，服务器是必不可少的存在。此项目旨在实现一个多线程高并发企业级服务器的核心框架，日后做任何软件服务器都可以基于此框架进行拓展。

**2.2 功能**

1. 该服务器可在高并发环境下顺畅运行。
2. 按照包头包体格式正确接收客户端发送的数据包，解决数据粘包问题。
3. 根据收到数据包的不同执行不同的业务处理逻辑。
4. 把业务处理产生的结果正确返回客户端。

图形描述如下：

****

## 2.3 运行环境

以下是系统的软件环境。

（1）客户端

操作系统：Windows10

使用测试工具：Visual Studio 2017

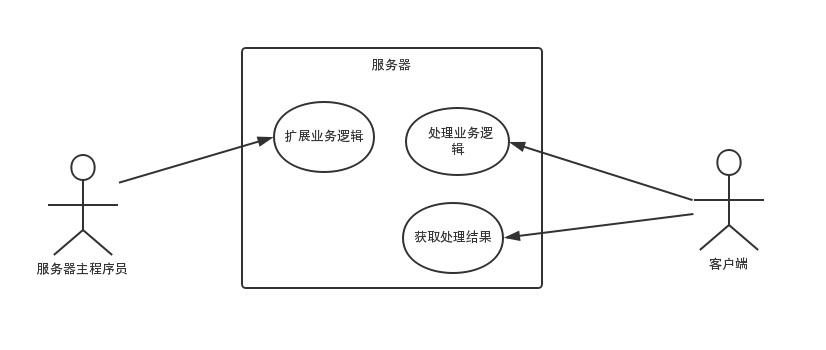
（2）服务器端

操作系统：Linux ubuntu 18

# 3. 需求描述

## 3.1 用户角色

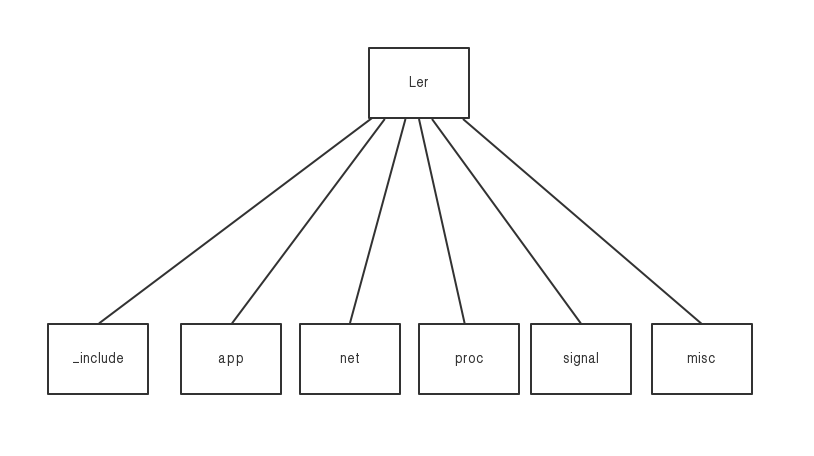
本项目的目标用户是客户端与服务器程序员：



## 3.2 服务器架构

使用makefile编译项目生成可执行文件

目录结构描述：



其中

\_include目录：专门存放各种头文件。

app目录：放主应用程序以及一些比较核心的文件。

net目录：专门存放和网络处理相关的文件。

proc目录：专门存放和进程处理有关的文件。

signal目录：专门用于存放和信号处理有关的文件。

misc目录：专门存放各种杂合性的不好归类的文件。

# **3.3 技术需求**

# 1）使用epoll高并发通讯技术。

# **2）实现一套线程池处理业务逻辑，调用适当的业务逻辑处理函数处理业务并返给客户端处理结果。**

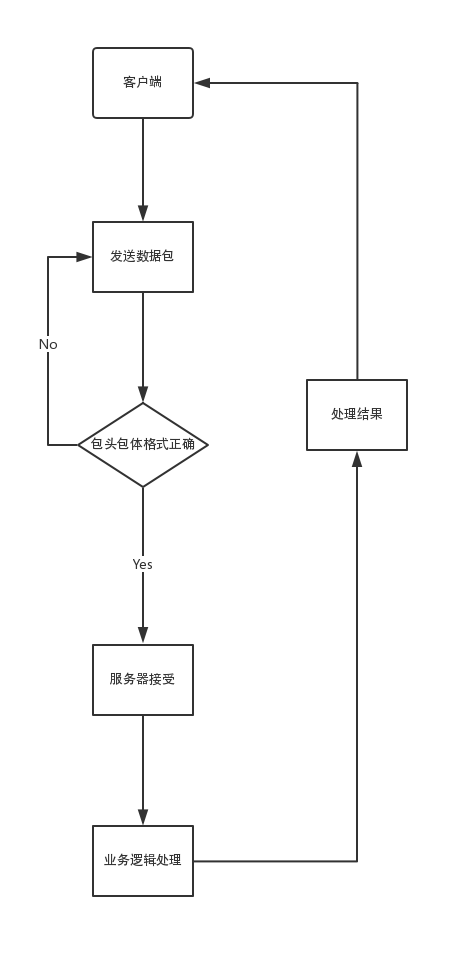
# **3）线程同步技术。**

**4）连接池中的延迟回收技术，需极大程度消除诸多导致服务器不稳定的因素。**

# **5）日志打印，创建守护进程。**

# 3.4 功能性需求

如下流程图：



# 4. 非功能性需求

## 4.1 性能需求

（1）服务器至少支持100000客户端同时连接

（2）由于不同机器测试数据不同，无法提供准确数据，但吞吐量与事件处理效率应至少与商业服务器（libevent、Nginx)持平。

### 4.2 安全性需求

（1）记录日志

本服务器应该能够记录服务器运行时所发生的所有错误，包括本机错误和网络错误。这些错误记录便于查找错误的原因。日志同时记录用户的关键性操作信息。

### 4.3 可用性需求

（1）方便操作，操作流程合理

（2）容错能力

服务器应具有一定的容错和抗干扰能力，在非硬件故障或非通讯故障 时，系统能够保证正常运行。

（3）用户可自定义

用户（服务器主程序员）可对服务器进行扩展与完善。

### 4.4 用户文档

同本软件一起发行的用户文档包括：

（1）安装手册：Word格式文件。

（2）用户手册：Word格式文件。

（3）在线帮助：HTML Help格式文件，联机式。

### 4.5 其它需求

服务器安装使用方便，易于维护。

# 5. 外部接口需求

## 5.1 用户接口

本服务器运行在Linux环境中

## 5.2 硬件接口

无特殊需求

## 5.3 软件接口

无特殊需求。

## 5.4 通信接口

无特殊需求。