好的，这是根据提供的“零声教育 17代Linux C/C++开发成长体系课程大纲 v1.12”内容整理成的清晰目录结构：

​**零声教育 17代Linux C/C++开发成长体系课程大纲 (v1.12)​**​

​**一、 精进基石专栏**​  
\* ​**1.1 数据结构与算法**​  
\* 1.1.1 随处可见的红黑树 (应用场景、进程调度CFS、内存管理、数学证明、左旋右旋、添加实现与证明、删除实现与证明、线程安全、工程特点)  
\* 1.1.2 磁盘存储链式的B树与B+树 (多叉树应用、B树定义证明、插入分裂、删除借位合并、遍历查找实现、B+树定义实现、应用场景、实用特点、线程安全)  
\* 1.1.3 海量数据去重的Hash与BloomFilter, bitmap (Hash原理与实现、应用场景、分布式Hash、布隆过滤器原理、数学推导证明、bitmap)  
\* ​**1.2 设计模式**​  
\* 1.2.1 创建型设计模式 (单例模式、工厂方法模式、抽象工厂模式、原型模式)  
\* 1.2.2 结构型设计模式 (适配器模式、代理模式、桥接模式、组合模式)  
\* 1.2.3 行为型设计模式 (策略模式、观察者模式、责任链模式、状态模式)  
\* ​**1.3 C++新特性**​  
\* 1.3.1 STL容器, 智能指针, 正则表达式 (unordered\_map, shared\_ptr, unique\_ptr, basic\_regex, sub\_match)  
\* 1.3.2 新特性的线程, 协程, 原子操作, lambda表达式 (thread, thread\_local, condition\_variable, atomic, 内存序, coroutine, function, bind, 异常处理)  
\* ​**1.4 Linux工程管理**​  
\* 1.4.1 Makefile/cmake/configure (Makefile规则与原理、单文件/多文件编译、参数传递、递归编译、通配符/伪目标/文件搜索、操作函数/特殊语法、configure生成原则、cmake写法)  
\* 1.4.2 分布式版本控制git (工作流程、创建与基本操作、分支管理、查看历史、服务器搭建)  
\* 1.4.3 Linux系统运行时参数命令 (进程间通信状态 ipcs, 系统运行时长 uptime, CPU负载/磁盘活动 iostat, 系统活动监控 sar, 多处理器监控 mpstat, 进程内存监控 pmap/glances, 调优基准测量 nmon, 系统调用跟踪 strace, FTP信息 ftptop/ftp, 电源管理 powertop, MySQL监控 mytop, 系统分析 htop/top/atop, 网络统计 netstat, 网卡配置 ethtool, 抓包 tcpdump, 远程登录 telnet, 实时网络统计 iptraf, 带宽监控 iftop)  
\* 1.4.4 vscode gdb调试 (环境安装、调试准备、命令详解、多线程调试、附加进程调试、core文件调试)

​**二、 高性能网络设计专栏**​  
\* ​**2.1 网络编程 (异步网络库zvnet)​**​  
\* 2.1.1 网络IO与IO多路复用 (Socket与fd、select/poll/epoll、LT/ET区别代码实现)  
\* 2.1.2 事件驱动Reactor的原理与实现 (针对业务的优点、多核实现、跨平台封装、send\_cb/recv\_cb/accept\_cb封装、sendbuffer/recvbuffer封装、HTTP协议实现)  
\* 2.1.3 HTTP服务器的实现 (协议格式、有限状态机FSM解析、同步/异步处理差异)  
\* ​**2.2 网络原理**​  
\* 2.2.1 服务器百万并发实现 (实操: ulimit fd调优、sysctl.conf rmem/wmem调优、conntrack原理)  
\* 2.2.2 Posix API与网络协议栈 (connect/listen/accept与握手、listen backlog、SYN泛洪解决、close与挥手、11状态迁移、TIME\_WAIT/CLOSE\_WAIT解决、keepalive与心跳、拥塞控制与滑动窗口)  
\* 2.2.3 UDP的可靠传输协议QUIC (UDP优缺点、高并发设计、早期QQ选择原因、可靠传输原理、QUIC设计原理、开源方案quiche、KCP方案与算法)  
\* ​**2.3 自研框架：协程框架NtyCo的实现(已开源)​**​  
\* 2.3.1 协程设计原理与汇编实现 (存在原因、原语、切换方式setjmp/ucontext/汇编、寄存器、启动、栈空间定义、结构体定义)  
\* 2.3.2 协程调度器实现与性能测试 (调度器定义分析、超时/就绪/IO等待集合、执行流程、接口实现、异步流程、hook实现、MySQL请求示例、多核方案、性能测试)  
\* ​**2.4 自研框架：基于DPDK的用户态协议栈的实现(已开源)​**​  
\* 2.4.1 用户态协议栈设计实现 (存在场景与原理、netmap框架)  
\* 2.4.2 TCP的原理实现 (TCP 11状态实现、滑动窗口慢启动、定时器)  
\* 2.4.3 应用层POSIX API的具体实现 (协议实现: ETH/IP/UDP/ARP/ICMP、Socket API实现: socket/bind/listen/accept/recv/send)  
\* 2.4.4 手把手设计实现epoll (数据结构封装与线程安全、fd就绪回调、接口实现、LT/ET实现、线程池异步处理)  
\* ​**2.5 高性能异步IO机制**​  
\* 2.5.1 与epoll媲美的io\_uring (使用场景、系统调用io\_uring\_setup/register/enter、liburing关系、与epoll性能对比、共享内存机制、accept/connect/recv/send实现、网络/磁盘读写、Proactor实现)  
\* 2.5.3 Windows异步机制IOCP (工作机制、重叠IO精髓、连接维护与数据处理、多线程方案、开发调试环境安装)

​**三、 基础组件设计专栏**​  
\* ​**3.1 池式组件 (C++11/14/17)​**​  
\* 3.1.1 手写线程池与性能分析(项目) (使用场景、组成任务/执行队列、任务回调/条件等待、动态防缩、多生产者多消费者优化)  
\* 3.1.2 内存池的实现与场景分析(项目) (应用场景与性能分析、小块/大块分配管理、手写实现与API、避免泄漏方法、定位泄漏工具、Nginx内存池扩展)  
\* 3.1.3 MySQL连接池的实现(项目) (性能影响因素、请求归还策略、超时策略、同步/异步运行机理)  
\* ​**3.2 高性能组件 (C++11/14/17)​**​  
\* 3.2.1 原子操作CAS与锁实现(项目) (CPU缓存与MESI、C++11原子操作与内存序、lock-free/wait-free/blocking、volatile、内存屏障)  
\* 3.2.2 无锁消息队列实现(项目) (DPDK rte\_ring源码阅读)  
\* 3.2.3 网络缓冲区设计 (项目: RingBuffer, fixed buffer, ChainBuffer设计; redis/memcached/nginx网络组件)  
\* 3.2.4 定时器方案红黑树, 时间轮, 最小堆(项目) (使用场景、数据结构选择红黑树/最小堆/时间轮、触发机制、手撕实现)  
\* 3.2.5 手写死锁检测组件(项目) (死锁现象原理、pthread\_mutex、dlsym、有向图构建与DFS判环、原语操作lock\_before/after/unlock\_after、检测线程实现)  
\* 3.2.6 手写内存泄漏检测组件(项目) (内存泄漏现象、第三方/代码泄漏、malloc/free的dlsym实现、检测策略、场景测试)  
\* 3.2.7 手把手实现分布式锁(项目) (使用场景、异常分析、基于Redis实现、std::mutex底层)  
\* ​**3.3 开源组件**​  
\* 3.3.1 异步日志方案spdlog(项目) (日志库性能瓶颈、异步日志设计实现、批量写入与双缓冲、崩溃日志找回)  
\* 3.3.2 应用层协议设计ProtoBuf(项目) (IM/云平台/nginx/http/redis协议设计、消息完整性保证、手撕protobuf、序列化反序列化、编码原理)

​**四、 中间件开发专栏**​  
\* ​**4.1 Redis**​  
\* 4.1.1 Redis相关命令详解及其原理 (string/set/zset/list/hash命令、分布式锁、Lua脚本解决ACID、事务ACID分析)  
\* 4.1.2 Redis协议与异步方式 (协议解析、特殊操作、订阅发布、手撕异步协议)  
\* 4.1.3 存储原理与数据模型 (编码方式int/raw/embstr、list实现、hash实现/hash函数/冲突/rehash、跳表实现与论证、整数集合、压缩列表、对象类型与编码/字符串/列表/哈希/集合/有序集合、类型检测/命令多态、内存回收、对象共享、空转时长)  
\* 4.1.4 主从同步与对象模型 (集群方式主从/sentinel/cluster、4种持久化方案)  
\* ​**4.2 MySQL**​  
\* 4.2.1 SQL语句, 索引, 视图, 存储过程, 触发器 (体系结构、SQL执行流程、CURD/高级查询、视图/触发器/存储过程、权限管理)  
\* 4.2.2 MySQL索引原理以及SQL优化 (索引/约束与区别、B+树/聚集索引/辅助索引、最左匹配/覆盖索引、失效/优化原则、EXPLAIN执行计划)  
\* 4.2.3 MySQL事务原理分析 (ACID特性、并发问题脏读/不可重复读/幻读、隔离级别、锁类型/算法/对象S/X/IS/IX/记录/间隙/next-key/插入意向/自增锁、MVCC原理)  
\* 4.2.4 MySQL缓存策略 (读写分离/连接池局限、缓存策略问题、强一致性/最终一致性方案、2种同步方案从库+触发器/UDF、开源方案go-mysql-transfer/canal原理、缓存故障击穿/穿透/雪崩)  
\* ​**4.3 Kafka**​  
\* 4.3.1 Kafka使用场景与设计原理 (发布订阅、点对点消息、Brokers原理、Topics/Partitions)  
\* 4.3.2 Kafka存储机制 (Partition存储分布、Segment文件结构、offset查找、高效文件存储设计)  
\* ​**4.4 微服务之间通信基石 gRPC**​  
\* 4.4.1 gRPC的内部组件关联 (ClientSide/ServerSide/Channel/Service/Stub概念、异步实现、回调异步调用、Server/Client对RPC实现)  
\* 4.4.2 基于http2的gRPC通信协议 (基于HTTP构造、ABNF语法、Request-Headers、上下文传递)  
\* ​**4.5 Nginx**​  
\* 4.5.1 Nginx反向代理与系统参数配置conf原理 (静态文件配置、动态接口代理、MQTT转发、RTMP推拉流、Openresty代理Redis)  
\* 4.5.2 Nginx过滤器模块实现 (Filter模块原理、过滤链表顺序、模块开发数据结构ngx\_str\_t/ngx\_list\_t/ngx\_buf\_t/ngx\_chain\_t、error日志、ngx\_command\_t、ngx\_http\_module\_t执行流程)  
\* 4.5.3 Nginx Handler模块实现 (Handler模块原理、ngx\_module\_t/ngx\_http\_module\_t、ngx\_http\_top\_body/header\_filter原理、ngx\_rbtree\_t使用与自定义添加、性能测试、核心数据结构ngx\_cycle\_t/ngx\_event\_module\_t、请求11阶段、包体处理、响应发送、Upstream机制)

​**五、 开源框架专栏**​  
\* ​**5.1 游戏服务器开发 skynet(录播答疑)​**​  
\* 5.1.1 Skynet设计原理 (多核并发模型多线程/进程/CSP/actor、actor模型lua/c服务实现、消息队列、actor消息调度、网络模型封装reactor、socket/socketchannel封装、手撕高性能c服务)  
\* 5.1.2 skynet网络层封装以及lua/c接口编程 (Lua编程及lua/c接口、基础接口skynet.send/call/response)  
\* 5.1.3 skynet重要组件以及手撕游戏项目 (广播组件multicastd、数据共享sharedatad/datasheet、手撕万人同时在线游戏)  
\* ​**5.2 分布式API网关**​  
\* 5.2.1 高性能Web网关 Openresty (Nginx与lua模块、访问Redis/MySQL、Restful API开发、性能分析)  
\* 5.2.2 Kong (动态负载均衡与服务发现、nginx/openresty/Kong关系、动态负载均衡原理、服务发现原理、Serverless、系统日志、监控故障恢复、代理层缓存)  
\* ​**5.3 高性能计算 CUDA(录播答疑)​**​  
\* 5.3.1 GPU并行计算CUDA的开发流程 (计算机体系GPU、环境搭建、nvcc/srun、向量加法/矩阵乘法、MPI与CUDA)  
\* 5.3.2 音视频编解码中的并行计算 (CUDA的H264/MPEG编解码、FFmpeg的CUDA支持、CPU+GPU异构计算)  
\* ​**5.4 并行计算与异步网络引擎 workflow**​  
\* 5.4.1 workflow的应用场景 (应用场景)  
\* 5.4.2 workflow的组件实现 (编程范式与设计理念、mysql/redis/kafka/dns请求实现、parallel处理与任务组装、线程池实现DAG图任务、msgqueue实现、纯c jsonparser实现)  
\* ​**5.5 物联网通信协议 mqtt的实现框架mosquitto**​  
\* 5.5.1 MQTT的高效使用场景 (高效使用场景、发布订阅模式、低带宽传输、3种QoS、遗嘱机制、过滤器)  
\* 5.5.2 MQTT的broker (mosquitto的docker部署、日志实时监控、OAuth与JWT认证)  
\* ​**5.6 ZeroMQ**​  
\* 5.6.1 消息队列与ZeroMQ的应用 (消息模型REQ/REP, PUB/SUB, PUSH/PULL, Router/Dealer原理分析)  
\* 5.6.2 ZeroMQ源码分析：消息模型的实现 (消息模型、传递模式、分帧、中间层代理、消息丢失处理)  
\* 5.6.3 ZeroMQ源码分析：网络机制与性能分析 (零拷贝、消息高水位标记、无锁队列、可靠性设计、性能分析)

​**六、 云原生专栏**​  
\* ​**6.1 Docker**​  
\* 6.1.1 Docker风光下的内核功能(录播答疑) (namespaces进程/UTS/IPC/网络/文件系统、cgroup资源控制)  
\* 6.1.2 Docker容器管理与镜像操作(录播答疑) (镜像下载与运行、存储管理)  
\* 6.1.3 Docker网络管理(项目) (5种网络驱动、pipework跨主机通信、OvS划分vlan与隧道、GRE跨主机通信)  
\* 6.1.4 Docker云与容器编排(项目) (Dockerfile语法流程、编排Fig/Compose、Flynn体系、Docker改变)  
\* ​**6.2 Kubernetes**​  
\* 6.2.1 k8s环境搭建(录播答疑) (集群安全设置、网络设置、核心服务配置、kubectl工具)  
\* 6.2.2 Pod与Service的用法(录播答疑) (Pod管理配置/升级回滚、DNS服务、HTTP7层/TLS设置)  
\* 6.2.3 k8s集群管理的那些事儿(项目) (RESTful接口、API聚合/组、Node管理、集群日志/监控、namespace隔离、yaml语法)  
\* 6.2.4 k8s二次开发与k8s API(项目) (Go访问k8s API)

​**七、 性能分析专栏**​  
\* ​**7.1 性能与测试工具**​  
\* 7.1.1 测试框架gtest以及内存泄漏检测(录播答疑) (googletest/gmock文件、函数/类测试、test fixture、类型参数化、事件测试、内存泄漏检测、设置期望/参数/次数)  
\* 7.1.2 性能工具与性能分析(录播答疑) (测试工具mysqlslap/redis-benchmark/wrk/TCPBenchmarks、磁盘/内存/网络性能分析)  
\* 7.1.3 火焰图的生成原理与构建方式 (使用场景与原理、生成工具、nginx/MySQL/Redis火焰图)  
\* ​**7.2 观测技术BPF与eBPF**​  
\* 7.2.1 bpftrace的实现原理 (跟踪/嗅探/采样/可观测理解、动态hook kprobe/uprobe、静态hook tracepoint/USDT)  
\* 7.2.2 bpf对tcp请求的监控(项目) (CPU观测taskset、网络观测tcp\_connect/accept/close、五元组获取、流量统计)  
\* 7.2.3 bpf实现对goroutine的探测(项目) (Go webserver实现、uprobe挂载、ringbuffer原理、进程maps、libjvm.so)  
\* 7.2.4 bpf探测java的垃圾回收(项目) (bpf usdt使用、Java GC耗时探测)  
\* ​**7.3 内核源码机制**​  
\* 7.3.1 进程调度机制那些事儿 (进程调度CFS与其他调度类、task\_struct、RCU机制、内存优化屏障、QEMU调试)  
\* 7.3.2 内核内存管理运行机制 (虚拟内存地址布局、SMP/NUMA模型、页表与缓存、伙伴系统、块分配Slab/Slub/Slob、brk/kmalloc/vmalloc系统调用流程)  
\* 7.3.3 文件系统组件 (虚拟文件系统VFS、Proc文件系统、super\_block/inode、文件描述符与挂载)

​**八、 分布式架构专栏**​  
\* ​**8.1 分布式数据库**​  
\* 8.1.1 不一样的kv存储RocksDB的使用场景 (使用场景)  
\* 8.1.2 TiDB存储引擎的原理 (集群组件TiDB Server/PD Server/TiKV Server、Raft协议、OLTP/OLAP、TiKV引擎)  
\* 8.1.3 TiDB集群方案与Replication原理 (基于RBAC权限、数据加密)  
\* ​**8.2 分布式文件系统(录播答疑)​**​  
\* 8.2.1 内核级支持的分布式存储Ceph (集群部署monitor/OSD、5核心组件、集群监控、性能调优与benchmark)  
\* 8.2.2 分布式Ceph存储集群部署 (同步机制、线性扩容、高可用实现、负载均衡、blobstore与blob关系、4层结构设计)  
\* ​**8.3 分布式协同**​  
\* 8.3.1 注册服务中心Etcd (etcd配置/服务发现/集群监控/leader选举/分布式锁、体系结构gRPC/WAL/Snapshot/BoltDB/Raft、存储原理B树/B+树、读写机制与事务ACID)  
\* 8.3.2 协同事件 (同步机制)  
\* 8.3.3 快播核心技术揭秘 P2P框架的实现(录播答疑) (网关NAT表分析、NAT类型完全锥型/对称/端口限制锥型/IP限制锥型、NAT类型检测代码、网络穿透原理与3种情况)  
\* ​**8.4 分布式监控 Prometheus**​  
\* 8.4.1 Prometheus监控系统 (数据模型Metric/Time Series/Sample/Label、指标类型Counter/Gauge/Histogram/Summary、采集/警告规则、PromQL)  
\* 8.4.2 Prometheus对MySQL/Redis的监控 (Exporter安装配置、PromQL计算性能指标)  
\* 8.4.3 Prometheus数据采集 (6种采集点配置、动态重写目标、自定义应用指标Go实现、远程存储)  
\* 8.4.4 Prometheus构建企微与邮件的状态告警 (邮件/企微接入、Alertmanager配置容器化、警告分组、Grafana仪表盘)

​**九、 上线产品项目(基础架构, 从零构建)​**​  
\* ​**9.1 dkvstore实现(基础架构开发)​**​  
\* 9.1.1 kv存储的架构设计 (存储节点定义、TCP server/client、hash/list/skiptable/rbtree数据存储)  
\* 9.1.2 网络同步与事务序列化 (序列化格式、事务建立释放、线程安全处理)  
\* 9.1.3 内存池的使用与LRU的实现 (大块/小块分配策略、内存回收机制、吞吐量测试、数据持久化)  
\* 9.1.4 KV存储的性能测试 (网络测试TPS、Go/Lua/Java支持、hash/list/skiptable/rbtree测试)  
\* ​**9.2 zrpc的实现(基础架构开发)​**​  
\* 9.2.1 rpc使用场景与架构设计 (分布式调度、网络业务分离、网络选型reactor/协程/io\_uring)  
\* 9.2.2 rpc的协议与序列化 (json定义函数、数据序列化、rpc处理流程、接口统一化、float/double差异、第三方语言支持)  
\* 9.2.3 rpc代码自动生成器的实现 (性能测试上线、内存池特性)  
\* ​**9.3 DPDK的数据包测试仪(网络开发)​**​  
\* 9.3.1 测试仪架构设计与使用场景 (使用场景与测试方案、多核多线程亲缘性、命令行参数)  
\* 9.3.2 协议流组织 (ARP数据组织、UDP包、TCP SYN/RST/FIN/ACK包、ICMP协议组织)  
\* 9.3.3 异常情况与测试标准 (异常命令行参数处理、异常网络包处理、CPU/内存不足处理)  
\* ​**9.4 openresty的zv\_waf的实现(网关防火墙开发)​**​  
\* 9.4.1 WAF基础架构与IP/URL防御 (OpenResty与WAF架构、IP黑白名单、URL基础过滤、日志监控)  
\* 9.4.2 WAF请求头与基础参数防御 (User-Agent识别、Referer验证、Cookie过滤、HTTP方法限制)  
\* 9.4.3 WAF CC攻击防护与请求频率控制 (CC攻击原理防御、人机验证、IP动态封禁放行、CC防护)  
\* 9.4.4 WAF深度参数过滤与SQL注入/XSS防御 (Query参数过滤、POST数据过滤、SQL注入过滤、XSS防御)  
\* ​**9.5 qemu的virtio/vhost虚拟化实现(虚拟化开发)​**​  
\* 9.5.1 virtio与vhost的工作原理 (vhost与qemu通信协议、get features、recvmsg作用、setmem与vring设置、gpa/gva/hva/hpa关系)  
\* 9.5.2 virtio设备模型实现 (ifreq创建tap/tun、vring物理/虚拟内存映射)  
\* 9.5.3 vhost的性能优化与多队列网卡 (vhost/virtio总结、多队列网卡实现、vhost用户空间后端开发)

​**十、 上线产品项目(开源项目, 框架迭代)​**​  
\* ​**10.1 SPDK助力MySQL数据落盘,让性能腾飞 (存储方向)​**​  
\* 10.1.1 SPDK文件系统设计与实现 (用户态文件系统fuse、NVMe与PCIe原理、NVMe Controller与bdev rpc)  
\* 10.1.2 文件系统的posix api实现 (VFS、SPDK异步改造posix、同步API open/write/read/close实现)  
\* 10.1.3 文件系统的性能测试与承接mysql业务 (iodepth讲解、随机/顺序读写测试、承接MySQL业务)  
\* ​**10.2 魔兽世界后端TrinityCore(游戏方向)​**​  
\* 10.2.1 TrinityCore开发调试环境安装 (环境安装)  
\* 10.2.2 TrinityCore中高性能基础组件应用 (高性能定时器方案、线程模型锁/无锁队列、数据库连接池、日志库)  
\* 10.2.3 TrinityCore高性能网络模块设计 (Boost.Asio跨平台网络库、核心接口、在TC中的封装、应用实践)  
\* 10.2.4 TrinityCore游戏技能模块设计 (技能配置与数据库表设计、组成部分、生成/处理/生命周期管理、暴风雪/背刺等实现)  
\* 10.2.5 游戏AI模块设计 (AI实现层次、三种细分方向、固定AI设计与幼狼案例、脚本AI设计与Boss案例)  
\* 10.2.6 游戏副本模块设计 (副本管理/匹配/积分结算、抽象层次实现、战歌峡谷/阿拉希盆地实现)  
\* ​**10.3 网络数据包过滤firewall(安全方向)​**​  
\* 10.3.1 firewall内核空间netfilter架构设计 (内核空间设计分析、数据包传递参数宏、端口/IP规则封装、检测规则/IP/协议函数设计)  
\* 10.3.2 firewall内核空间架构实现 (权限及规则检查、数据包出入站过滤器、读写规则/匹配实现、规则删除/模块加载卸载)  
\* 10.3.3 firewall用户空间架构设计 (用户空间设计分析、Hook入口函数设计、IP/掩码/端口函数、协议类型/解析/删除规则)  
\* 10.3.4 firewall应用程序调试与上线 (用户层访问与bug处理、测试调试技巧)  
\* ​**10.4 BPF网络数据包探测器(网关开发)​**​  
\* 10.4.1 监控探测仪的场景与架构设计 (应用场景与架构)  
\* 10.4.2 tc数据流记录存储 (bpf.c实现t