**C++基础**

1. **常量指针**(底层const)：const在\*号前。**指针常量**(顶层const)：const在\*号后(**指针是常量**)
2. **constexpr**：
3. **const成员变量**：需在构造函数初始化列表中初始化。**const成员函数**：不能修改类的任何非静态成员。在设计类的时候，一个原则就是对于不改变数据成员的成员函数都要在后面加 const(可以修改**mutable**修饰的成员变量)。
4. **volatile**：会从内存中重新装载内容，而不是直接从寄存器拷贝内容。
5. **重载前置++和后置++**：前置没有参数 &operator++()，后置有参数operator++(int)**(举例)**
6. **访问权限**：protected:只允许本类及其子类的成员函数访问。private: 只运行本类成员函数访问。
7. **重载**：类中同名函数，参数不同。**覆盖**：子类重新定义虚函数，**参数相同**、返回类型相同。
8. **虚继承**：解决多继承时的命名冲突和冗余数据问题 class B:public virtual A{ };
9. **Shared\_ptr**：管理动态创建对象的销毁。shared\_ptr<int> p1 = make\_shared<int>();
10. **C++类型转换**：static\_cast:子转父安全。 dynamic\_cast:父转子亦安全。
11. **const\_cast**：常量指针转换为非常量指针，常量引用转非常量引用。
12. **内存泄漏**：使用Valgrind、mtrace进行内存泄漏检测。防止泄漏：将内存的分配封装在类中，构造函数分配内存，析构函数释放内存；使用智能指针（共享指针、独占指针、弱指针）。
13. 函数参数为数组名，即数组首元素的地址，sizeof()为指针大小。
14. **memcpy**：未考虑内存重叠情况。**memove**：考虑内存重叠情况(if dst>src(左小右大) 末尾开始)。
15. **进程地址空间分布**：命令行参数&环境变量、栈、文件映射区、堆、BSS段、数据段、代码段。
16. **静态存储区域**：内存在程序编译的时候就已经分配好、它主要存放静态数据、全局数据和常量
17. **信号量**：binary\_semaphore,一次只能被一个线程所持有。counting\_semaphore,一次可以被多个线程所持有。
18. **feature库**：
19. **创建类有关的几个函数**：
20. 构造函数：复制(拷贝)构造函数：移动构造函数： **拷贝&移动**
21. 拷贝赋值运算符：移动赋值运算符：
22. **C++模板**：
23. **函数调用栈图**：

**C++ STL**

1. **STL**：提供六大组件，容器、算法、迭代器、仿函数、适配器(stack，queue、priority\_queue)、空间配置器(allocator)。
2. **allocator**：alloctor调用operator new()，deallocte调用operator delete()。

**C++ 11新特性**

1. **左值**：可以取地址、有名字。**右值**：不能取地址、没有名字。 ++i、--i是左值，i++、i—是右值。”abcd”是左值。
2. **C++右值引用**：const int &c = 10;//常量左值引用、c不能修改。const int&& c = 10;//常量右值引用、c不能修改。int &&c = 10;//右值引用，c可以修改。用于实现***C++完美转发***、
3. **nullptr**：之前NULL可能是0或((void\*)0),重载有歧义。使用nullptr避免。
4. **C++几种初始化方式**：内置类型习惯等号初始化。类类型习惯用构造函数圆括号显式初始化。vector、map、set习惯列表初始化。
5. **lambda表达式**：变量捕获[]、[&]、[=]、[=, &a]、[&, a]、[a]、[this].
6. **并发thread**：①可被joinable的thread对象必须在他们销毁之前被主线程join或者将其设置为detached.②detach可以让线程对象与线程函数脱离关系，即线程对象被销毁也不影响函数运行
7. **lock\_guard**：创建自动加锁、析构自动解锁。
8. **unique\_lock**：比lock\_guard灵活、有上锁和解锁控制

**操作系统**

1. **进程切换比线程切换慢**：进程(非父子进程)切换需要切换页表，TLB失效。线程不需要切换页表。
2. **僵尸进程**：子进程结束后，父进程读取前这一阶段子进程为僵尸进程。捕获子进程：wait() (阻塞)、waitpid(非阻塞)。
3. **多级队列**：=时间片轮转+优先级调度

**进程通信IPC**

1. **管道(pipe)**：四种特殊情况：缺陷：一个管道只能实现单向通信、只能用于父子或兄弟进程。
2. **命名管道(FIFO)**：命名管道存于硬盘上(属于文件)、用于任意两个进程之间的通信。创建FIFO：mkfifo(path, …)
3. **文件权限**：0666->即用户，组用户和其它用户具有读写权限。
4. **存储映射**：mmap(…,fd,…)，任意进程通信. **匿名映射**：mmap(…),用于父子进程通信
5. **消息队列(msg)**：key -> 标识符 ->消息类型。Key值可人为指定，或ftok()获得。创建消息队列：int msgget(key\_t key, int msgflg) 将新消息添加到消息队列：int msgsnd(int msqid, …) 从消息队列中读取一个消息：ssize\_t msgrcv(int msqid,…) 修改属性、删除：int msgctl(int msqid)
6. **信号**：常用信号：Ctrl+C终止进程(SIGINT). Ctrl+\终止程序(SIGQUIT). (SIGSEGV)无效内存访问(段错误)、产生core文件. SIGKILL、SIGSTOP不允许忽略和捕捉、阻塞。
7. **阻塞信号集**：需要屏蔽的信号加入集合，对他们设置屏蔽。
8. **未决信号集**：信号产生，为决信号集中描述该信号的位立刻翻转为1，处理时翻转为0
9. **Kill命令**：kill(pid, sig). pid(>0、=0、=-1、<-1). Sig(信号编号)。
10. **定时器**：unsigned int alarm(unsigned int seconds). setitimer()：
11. **signal**：
12. **sigaction**：int sigaction(int signum, const struct sigaction \*act,…)检查或修改指定信号的设置。
13. **守护进程模型**：
14. **线程**：
15. **共享内存**：
16. **信号量**：用于多任务同步，一个线程完成某一动作就通过信号量告诉其他线程，其他线程再进行动作。
17. ***锁和信号量***：
18. **Liux top进程状态**：R(可执行)、S(可中断睡眠态)、D(不可中断睡眠态)、T(暂停态和跟踪态)、Z(僵尸态)、X(退出态，将被销毁)
19. **系统调用过程**:
20. **进程、线程、协程**：

**存储系统**

1. **页面置换算法**：**最近最久未使用的置换算法(LRU)**. 根据页面未被访问时长用升序列表将页面排列、每次将最久未被使用页面置换出去
2. **分段和分页**：
3. **虚拟内存**：使进程认为自己拥有连续的可用内存空间，但实际上是被分隔的多个物理内存页、以及部分暂时存储在磁盘上的交换分区所构成。**优点**：
4. **TLB**：不通过访问页，表直接将虚拟地址映射到物理地址。称快表。在**内存管理单元(MMU)**中。
5. **用户态和内核态**：
6. **上下文切换**：
7. **系统启动流程分析**：
8. **DMA**: 直接内存访问

**互斥与同步**

1. **互斥锁(mutex)**：lock -> 访问 -> unlock
2. **死锁**：你等我、我等他、他等你。
3. **死锁检测与死锁恢复**：判断有向图是否存在环，存在则检测到死锁。恢复：a.利用抢占恢复 b.利用回滚恢复 c.杀死进程恢复
4. **死锁预防**：
5. **银行家算法**：
6. **读写锁**：允许多读一写，读时不能写，写时不能读。
7. ***条件变量***：
8. **信号量**：用于进程或线程的同步和互斥，本质为一个非负整数计数器。用于对公共资源的控制。值>0可访问、值=0不可访问。
9. **PV原语**：P(if(>0) --) 、V(++)
10. **管程**：简化客户端信号量控制代码

**文件系统**

1. **Linux7种文件类型**：普通文件、目录文件、设备文件(块设备文件、字符设备文件)、链接文件、管道文件、套接字文件。
2. **虚拟文件系统**：

**外设管理**

1. **键盘工作流程**：
2. **显示器工作流程**：

**计算机网络**

1. **三次握手**：client 🡪 SYN(seq = c\_num) 🡪 server.

server 🡪 SYN+ACK(ACKnum = c\_num+1, seq = s\_num) 🡪 client.

client 🡪 ACK(ACK\_num = s\_num + 1).

1. **三次握手优点**：同步双方初始序列号、避免资源浪费、阻止重复历史连接初始化(RST报文)。
2. **四次挥手**：client 🡪 FIN 🡪 server

server 🡪 ACK 🡪 client

server 🡪 FIN 🡪 client

client 🡪ACK 🡪server

1. **MSL**：报文最大生成时间。
2. **TIME\_WAIT**：防止旧链接的数据包
3. **HTTP**：文档传递协议
4. **SSL/TSL**：
5. **HTTP/HTTPS**：HTTP端口号80、HTTPS端口号443.
6. **HTTPS加密细节**：
7. **DNS**：迭代查询www.server.com：先找根，根说：你找com。再找com，com说：你找server.com。
8. **HTTP请求报文**：请求行(请求方法、URL、协议版本号) 请求头(包含请求的附加信息，由key：value组成) 请求体(承载多个请求参数的数据、含回车、换行和请求数据)
9. **HTTP响应报文**：响应行(协议版本、状态码、状态码描述) 响应头() 响应主体
10. **常见响应码**：
11. **cookie(本地)和session(服务器)**：
12. **TCP/UDP**：TCP拥有确认应答和重传机制、无重复、不丢失、按序到达；UDP尽可能交付、不保证可靠性。 TCP拥有流量控制、拥塞控制、保证传输安全性等；UDP在网络拥堵情况下不会降低发送速率。
13. **SYN攻击**：
14. **TCP拥塞控制**：1.**慢开始**：窗口1、2、4、8增大。2.**拥塞避免**：一个往返时间RTT就把窗口+1. 不论是在慢开始期间还是拥塞避免期间，只要判断网络发生了拥塞ssthresh就设置为当前发送窗口大小的一半，然后重新开始执行慢开始算法。3.**快速重传**：快速重传算法规定，发送方只要一连收到三个重复确认就应当立即重传对方还没有接收到的报文段，而不必继续等待设置的重传计时器时间到期。4.**快速恢复**：将窗口设置为ssssthresh减半的值、然后执行拥塞避免算法。
15. **TCP流量控制(滑动窗口)**：让发送方发送速率不要太快，要让接受方来得及接受。
16. **发送窗口**：窗口大小在**拥塞窗口(拥塞控制中发送方的窗口)**和**接受窗口**中取小者。
17. **tcpdump和Wireshark**
18. **IP和MAC**：源IP地址和目标IP地址在传输过程中是不会变的、只有源MAC地址和目标MAC一直在变化。
19. **CIDR无分类地址**：10.100.122.2/24 🡪 10.100.122.2/255.255.255.0 网络号：10.100.122.0
20. **子网划分**：将主机地址再分为子网网络地址和子网主机地址。192.168.1.100/26 🡪 192.168.1.100/255.255.255.192 ：子网网络地址：192.168.1.64
21. ***IP最长匹配***：匹配到主网络号、继续匹配到子网络号。
22. **RARP**：已知MAC求IP
23. **DHCP**：Client(UDP:68) 🡪 DHCP DISCOVER 🡪 Server(UDP:67)

Server 🡪 DHCP OFFER 🡪 Client

Client 🡪 DHCP REQUEST 🡪 Server

Server 🡪 DHCP ACK 🡪 Client

1. **DHCP中继代理**：实现不同网段的IP地址分配由一个DHCP服务器统一管理
2. **NAT转换**：ip1: 192.168.1.10:1025, ip2: 192.168.1.11:1025 🡪 公网ip:1025,公网ip:1026
3. **NAT缺点**：外部无法主动与NAT内部服务器建立连接。
4. **NAT穿透技术**：
5. **ICMP**：查询报文类型、差错报文类型
6. **浏览器输入网址后**：
7. **MTU**：一个网络包的最大长度、一般1500字节。
8. **OSI参考模型**：物理层、数据链路层、网络层、传输层、会话层、表示层、应用层
9. **TCP/IP模型**：网络接口层、网络层、传输层、应用层
10. **RPC远端过程调用**:

**数据库**

**鱼群计数与测量项目**

1. **卡尔曼滤波**：
2. **伪彩处理**：伪彩色处理(pseudocoloring)是指根据一定准则给灰度值赋予彩色值的处理。

**Linux C++ WebServer项目**