编译源码包过程介绍。

采用源码包编译的方式安装软件包时，有哪些基本过程、各自的作用是什么？

参考答案：

tar 解包：解压、释放安装包内的文件

./configure 配置：针对当前系统环境指定安装目录、选择功能等设置。

make 编译：将源代码编译成二进制的可执行程序、库文件等数据。

make install：将编译好的程序文件、配置文档等复制到对应的安装目录

描述 Linux 运行级别 0-6 的各自含义

0：关机模式

1：单用户模式<==破解 root 密码

2：无网络支持的多用户模式

3：有网络支持的多用户模式（文本模式，工作中最常用的模式）

4：保留，未使用

5：有网络支持的 X-windows 支持多用户模式（桌面）

6: 重新引导系统，即重启

简述 MySQL 数据库访问的执行过程。

参考答案

1）客户端发出请求。

2）服务器端开辟线程响应客户端请求。

3）客户端发起 sql 语句查询数据库。

4）查询缓存：记录用户的 sql 查询语句，如果查询内容相同，直接从查询缓存回复。

5）如果缓存没有进入分析器。

6）分析器：分析用户命令语法是否正确，将用户的命令进行切片，一个词一个词用空格隔开，获得用户要查询的

表、内容、用户的权限等。

7）优化器：执行路径的选择，生成执行树。（每个 SQL 语句都有很多执行路径，优化的目的就是在这些执行路径

里选择最优的执行路径）。

8）存储引擎：用于管理存储的文件系统，不同的存储引擎有不同的功能和存储方式。

MySQL 主从复制原理？

答案：

分为同步复制和异步复制，实际复制架构中大部分为异步复制。

复制的基本过程如下：

1)、 Slave 上面的 IO 进程连接上 Master，并请求从指定日志文件的指定位置（或者从最开始的日志）之后的

日志内容；

2)、 Master 接收到来自 Slave 的 IO 进程的请求后，通过负责复制的 IO 进程根据请求信息读取制定日志指定位

置之后的日志信息，返回给 Slave 的 IO 进程。返回信息中除了日志所包含的信息之外，还包括本次返回的信息已

经到 Master 端的 bin-log 文件的名称以及 bin-log 的位置；

3)、 Slave 的 IO 进程接收到信息后，将接收到的日志内容依次添加到 Slave 端的 relay-log 文件的最末端，并

将读取到的 Master 端的 bin-log 的文件名和位置记录到 master-info 文件中，以便在下一次读取的时候能够清楚

的告诉 Master“我需要从某个 bin-log 的哪个位置开始往后的日志内容，请发给我”；

4)、 Slave 的 Sql 进程检测到 relay-log 中新增加了内容后，会马上解析 relay-log 的内容成为在 Master 端真

实执行时候的那些可执行的内容，并在自身执行。

Mysql 为了解决这个风险并提高复制的性能，将 Slave 端的复制改为两个进程来完成。提出这个改进方案的人

是 Yahoo!的一位工程师“Jeremy Zawodny”。这样既解决了性能问题，又缩短了异步的延时时间，同时也减少了可

能存在的数据丢失量。当然，即使是换成了现在这样两个线程处理以后，同样也还是存在 slave 数据延时以及数据

丢失的可能性的，毕竟这个复制是异步的。只要数据的更改不是在一个事物中，这些问题都是会存在的。如果要完

全避免这些问题，就只能用 mysql 的 cluster 来解决了。不过 mysql 的 cluster 是内存数据库的解决方案，需要将

所有数据都 load 到内存中，这样就对内存的要求就非常大了，对于一般的应用来说可实施性不是太大。

复制常用架构

Mysql 复制环境 90%以上都是一个 Master 带一个或者多个 Slave 的架构模式，主要用于读压力比较大的应用的

数据库端廉价扩展解决方案。因为只要 master 和 slave 的压力不是太大（尤其是 slave 端压力）的话，异步复制

的延时一般都很少很少。尤其是自 slave 端的复制方式改成两个进程处理之后，更是减小了 slave 端的延时。而带

来的效益是，对于数据实时性要求不是特别的敏感度的应用，只需要通过廉价的 pc server 来扩展 slave 的数量，

将读压力分散到多台 slave 的机器上面，即可解决数据库端的读压力瓶颈。这在很大程度上解决了目前很多中小型

网站的数据库压力瓶颈问题，甚至有些大型网站也在使用类似方案解决数据库瓶颈。{自己组织语言，没必要背这么多}；

Nginx 的优点?答案：  
（1） 它可以高并发连接：官方测试连接数（5 万） 实际生产环境中可支撑（2~4 万）  
（2） 内存消耗少  
（3） 成本低廉  
（4） 其他理由（配置文件非常简单， 支持 Rewrite 重写规则）  
（5） 内置的健康检查功能  
（6） 节省宽带（支持 GZIP 压缩，可以添加浏览器本地缓存的 Header 头）  
（7） 稳定性高  
（8） 支持热部署

如何动态访问日志或文件：

tailf 日志路径

cat 路径 | less

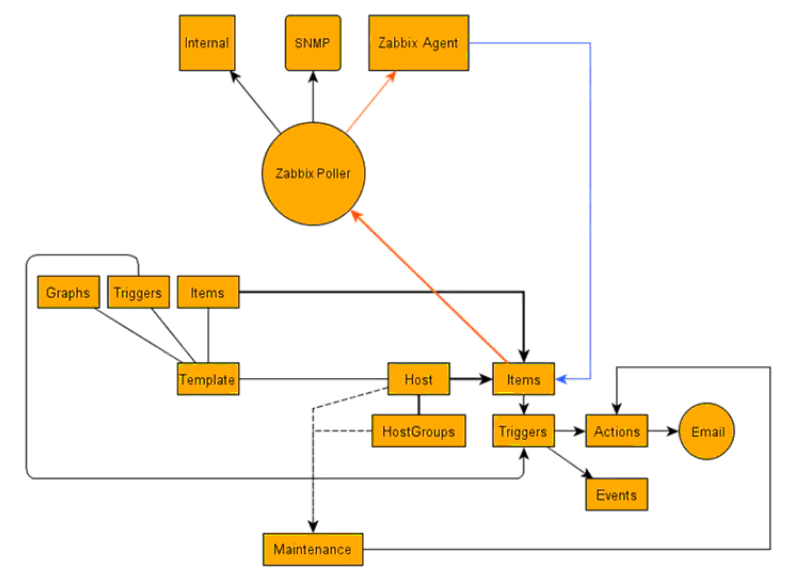
简单描述 **Zabbix** 具有哪些监控功能  
参考答案  
具备常见的商业监控软件所具备的功能  
主机性能监控、网络设备监控、数据库监控等  
支持多种报警机制  
支持自动发现网络设备和服务器  
可以通过配置自动发现服务器规则来实现  
支持分布式，能集中展示、管理分布式的监控点  
编写插件容易，可以自定义监控项  
具有实时绘图功能

Cacti和nagios的区别

Cacti：在监控方面绘图计较不错，在流量与图形展现比较存在优势

Nagios：在故障分析比较不错，报警机制相对来说比较好， 报警机制：邮箱、短信等，而且比cacti灵活；同事使用大量服务器以及服务器上面大批服务状态是否正常，重点不在图形化，而在状态故障的监控。

Zabbix的构架图



简单命令基础（无答案）

显示当前用户

Grep 和find命令的使用

Vim中显示行号、行首、删除、复制、粘贴、替换

查看隐藏文件

查看当前系统的的字符编码