--------------------

涵盖方向：php、nginx、mysql、linux

要有题目和标准答案

主要针对外包和日常基础一面

每人最少五道题

2018.09.06下班前完成

辛苦各位哈

php基础

**1.在        上填写函数返回值**

$a = "";

$b = 0;

$c = null;

empty($a) :                  true

empty($b) :                  true

empty($c) :                  true

isset($a) :                  true

isset($b) :                  true

isset($c) :                  false

**2.php支持多线程吗？**

答：不支持

**3.== 和 === 有什么区别？**

答：===比较类型和值，==只比较值。

mysql

**1.电商系统中有两个表，用户user表(user\_id,user\_name,sex)和用户订单order表(user\_id,order\_id, status)，**

**a.写出导出 所有男用户及他们的订单 的sql,并按照用户的id进行排序**

**b.写出统计不同订单状态下订单数量的sql**

 select user.user\_name order.order\_id from user left join order on user.user\_id=order.user\_id where user.sex='man' order by user\_id;

select count(1) from order group by status;

2.**一次mysql查询能用几个索引？**

答：一次查询只能用一个索引。

redis

**1.redis有哪些数据类型**

string(字符串),hash(哈希),list(列表),set(集合)及zset(sorted set:有序集合)

nginx

**1.nginx 有哪些负载均衡的算法**

轮询（默认):每个请求按时间顺序逐一分配到不同的后端服务，如果后端某台服务器死机，自动剔除故障系统，使用户访问不受影响。

weight（轮询权值）:weight的值越大分配到的访问概率越高，主要用于后端每台服务器性能不均衡的情况下。或者仅仅为在主从的情况下设置不同的权值，达到合理有效的地利用主机资源。

ip\_hash: 每个请求按访问IP的哈希结果分配，使来自同一个IP的访客固定访问一台后端服务器，并且可以有效解决动态网页存在的session共享问题。

fair: 比 weight、ip\_hash更加智能的负载均衡算法，fair算法可以根据页面大小和加载时间长短智能地进行负载均衡，也就是根据后端服务器的响应时间 来分配请求，响应时间短的优先分配。Nginx本身不支持fair，如果需要这种调度算法，则必须安装upstream\_fair模块。

url\_hash:按访问的URL的哈希结果来分配请求，使每个URL定向到一台后端服务器，可以进一步提高后端缓存服务器的效率。Nginx本身不支持url\_hash，如果需要这种调度算法，则必须安装Nginx的hash软件包

**2.break和last的区别**

last：停止当前请求，根据rewrite匹配的规则重新发起一个请求。

break：停止当前请求的rewrite阶段，并继续执行本请求的其他阶段

location /break/ {
rewrite ^/break/(.\*) /test/$1 break;
echo "break page";
}
location /last/ {
rewrite ^/last/(.\*) /test/$1 last;
echo "last page";
}
location /test/ {
echo "test page";
}

请求:http://xxx.com/break/xxx

输出: break page

请求:http://xxx.com/last/xxx

输出: test page

网络

1.http返回的不同状态码都表示什么信息?

1XX:临时/信息响应

2XX:成功

3XX:重定向

4XX:客户端/请求错误

5XX:服务器错误

|  |  |  |
| --- | --- | --- |
| 分类 | 问题 | 答案 |
| PHP | 如何进入php命令行模式 | php -a |
| PHP | php是否支持多继承 | php只支持单继承 |
| PHP | final类或者final方法有什么作用 | final修饰符的类不能被继承，方法不能被重写 |
| NGINX | nginx是什么 | nginx是一个HTTP, HTTPS, SMTP, POP3 and IMAP反向代理服务器 |
| NGINX | 日常怎么用nginx | 1、使用nginx提供生成动态内容的http server，使用wsgi、fastcgi处理脚本。  2、作为7层负载均衡使用。 |
| LINUX | 如何查看当前目录下各个文件大小 | 1、ls -hl  2、du -sh \* |
| LINUX | 输出access\_log第一列的值，并统计每个值得数量 | cut -d' ' -f 1 access\_log  | sort | uniq -c |
| MYSQL | mysql是什么 | 是一个多线程、多用户的关系型数据库 |
| MYSQL | mysql的默认端口是多少 | 3306 |

一、http协议，get和post的区别

GET后退按钮/刷新无害，POST数据会被重新提交（浏览器应该告知用户数据会被重新提交）。

GET书签可收藏，POST为书签不可收藏。

GET能被缓存，POST不能缓存 。

GET编码类型application/x-www-form-url，POST支持多重编码。

GET历史参数保留在浏览器历史中。POST参数不会保存在浏览器历史中。

GET对数据长度有限制，当发送数据时，GET 方法向 URL 添加数据；URL 的长度是受限制的（URL 的最大长度是 2048 个字符）。POST无限制。

GET只允许 ASCII 字符。POST没有限制。也允许二进制数据。

与 POST 相比，GET 的安全性较差，因为所发送的数据是 URL 的一部分。在发送密码或其他敏感信息时绝不要使用 GET ！POST 比 GET 更安全，因为参数不会被保存在浏览器历史或 web 服务器日志中。

GET的数据在 URL 中对所有人都是可见的。POST的数据不会显示在 URL 中。

二、常用linux习题

日志类型：第一列是请求IP

|  |  |
| --- | --- |
| 1 | 172.24.146.134 - - [24/Aug/2018:16:22:22 +0800] "bjyz-ps-201607-f62-nwise59.bjyz.baidu.com:8080" "GET /dapp/api/lovecard/witness/ranklist?csrf\_token=3858bd101c8cfdf5a40b375ef323aaed HTTP/1.1" 200 4295 "-" "[http://bjyz-ps-201607-f62-nwise59.bjyz.baidu.com:8080/dapp/lov](http://bjyz-ps-201607-f62-nwise59.bjyz.baidu.com:8080/dapp/lov%C2%A0%C2%A0%C2%A0) echain" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_11\_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36" "-" “0.037" |

文件名称： access\_log.log

写一条语句, 统计日志中访问次数最多的十个请求IP

|  |  |
| --- | --- |
| 1 | cat access\_log.log | awk '{print $1}' | uniq -c | sort -n | tail -10 |

三、缓存 Memcache与Redis的区别都有哪些？

存储方式

Memecache把数据全部存在内存之中，断电后会挂掉，数据不能超过内存大小。

Redis有部份存在硬盘上，这样能保证数据的持久性。

数据支持类型

Memcache对数据类型支持相对简单。

Redis有复杂的数据类型。

使用底层模型不同

它们之间底层实现方式 以及与客户端之间通信的应用协议不一样。

Redis直接自己构建了VM 机制 ，因为一般的系统调用系统函数的话，会浪费一定的时间去移动和请求。

value大小

redis最大可以达到1GB，而memcache只有1MB

四、php问题

鸟哥是谁？

惠新宸

五、 NGINX

error\_page 指令用来做什么的？ 怎么写？

用来收敛upstream返回的错误code的

error\_page 400 403 404 500 501 502 503 504 505 location地址

--------------------zhangwei07 ---------------------

1 有序数组arr，里面的元素都为整数。给定待查找的值 search=?, 查找该值在数组的最小下标。如果不存在，返回-1. 要求时间复杂度尽可能低。

eg      $arr = {1,2,3,3,4,4,5};

          $search = 3;   $ans = 2;

          $search = 7;  $ans = -1;

 面试过程中注意引导，以及熟悉面试者的编码风格，风格不好的直接挂掉。风格oK,但是思路不灵活的最多给到T3.

|  |
| --- |
| <?php  /\*\*   \*@param pos 默认取最左下标, pos=right取最右下标   \*@param arr 原始有序整数数组   \*@param search 待查找的值   \*@return 返回最${pos}下标，如果不存在返回-1   \*/  **function** binSearch($arr, $search, $pos='left'){      $len = count($arr);      $low = 0;      $high = $len -1;    **while**($low <  $high -1){          $mid = $low + round(($high - $low)/2);  **if**( $arr[$mid] < $search ||  ($arr[$mid]==$search && $pos=='right')){                  $low = $mid;          }**else**{                  $high = $mid;          }      }      $leftIndex = -1;      $rightIndex = -1;      //如果右节点数据和search相等  **if**($arr[$high] == $search){          $leftIndex = $rightIndex = $high;      }        //如果左边也相等，则最左下标取low,最右下标优先取high  **if**($arr[$low] == $search){          $leftIndex =  $low;          $rightIndex =  ($rightIndex == -1)? $low : $rightIndex;      }  **if**($pos == 'left'){  **return** $leftIndex;      }**else**{  **return** $rightIndex;      }  }  $arr = **array**(1,2,3,3,3,4,4,5);  echo 'searcharr ';  **foreach**($arr **as** $k => $v){      echo "[$k]$v ";  }  echo PHP\_EOL;  $search = $argv[1];  echo "search $search" .PHP\_EOL;  echo binSearch($arr,$search).PHP\_EOL;  echo binSearch($arr,$search,'right').PHP\_EOL; |

2 php的函数 count的时间复杂度

     o(1)

    如果答对进一步引导看是否对数组的底层数据结构有了解。

3 php中 isset 和 array\_key\_exists的区别

对于数组值的判断不同，对于值为null或''或false,isset返回false，array\_key\_exists返回true；

执行效率不同，isset是内建运算符，array\_key\_exists是php内置函数，isset要快一些。请参考：[PHP 函数实现原理及性能分析](http://wiki.skype.tom.of/index.php?doc-innerlink-PHP%E5%87%BD%E6%95%B0%E5%AE%9E%E7%8E%B0%E5%8E%9F%E7%90%86%E5%8F%8A%E6%80%A7%E8%83%BD%E5%88%86%E6%9E%90)

当用isset访问一个不存在索引数组值时，不会引起一个E\_NOTICE的php错误消息；

array\_key\_exists 会调用get\_defined\_vars判断数组变量是否存在，isset不用；

4 shell,给定两个文件filea,fileb，文件每一行都是一个整数id, 通过shell命令获取两个文件中相同的id。

        考察点： sort uniq 以及细心程度。 题目并未告知文件中id是否会存在重复，观察面试者是否会提问这个点，if,then加分。

        sort -u filea > filea.uniq

        sort -u fileb > fileb.uniq

        cat filea.uniq fileb.uniq |sort|uniq -d >both

        考察外包同学知识面。如果答不上来，不建议给到T4.

5 https和http的区别，为什么会使用https.

        考察外包同学知识面。如果答不上来，不建议给到T4.

--------------------zhangwei07 ---------------------

--------------------彦滔---------------------

1、LINUX

     （1）基础命令使用 ： 统计指定目录的文件个数

              find . -type f | wc -l

     （2）AWK基础： 给定日志文件，格式： IP\tTIME\t\UserAgent\... 统计出访问量最大的5个IP及访问次数

              cat  file\_name | awk '{print $1}' | uniq -c | sort -rn | head -n 5

2、PHP

     （1）PHP魔术方法 了解多少，各自功能，及应用场景

              \_\_construct()，类的构造函数

              \_\_destruct()，类的析构函数

              \_\_call()，在对象中调用一个不可访问方法时调用

              \_\_callStatic()，用静态方式中调用一个不可访问方法时调用

              \_\_get()，获得一个类的成员变量时调用

              \_\_set()，设置一个类的成员变量时调用

              \_\_isset()，当对不可访问属性调用isset()或empty()时调用

              \_\_unset()，当对不可访问属性调用unset()时被调用。

              \_\_sleep()，执行serialize()时，先会调用这个函数

              \_\_wakeup()，执行unserialize()时，先会调用这个函数

              \_\_toString()，类被当成字符串时的回应方法

              \_\_invoke()，调用函数的方式调用一个对象时的回应方法

              \_\_set\_state()，调用var\_export()导出类时，此静态方法会被调用。

              \_\_clone()，当对象复制完成时调用

              \_\_autoload()，尝试加载未定义的类

              \_\_debugInfo()，打印所需调试信息

3、NGINX NGINX负载均衡策略举例

|  |  |
| --- | --- |
| 调度策略 | 含义 |
| 轮询 | 按照时间顺序，逐一分配到不同的后端服务器 |
| 加权轮询 | weight值越大，分配到的访问几率越高 |
| ip\_hash | 每个请求按访问IP的hash结果分配，这样来自同一个IP的请求固定访问一个后端服务器，可以解决分布式session问题，但不是最优的解决办法，另一个即集中式session存储校验，将session放到redis集群当中。 |
| url\_hash | 按照访问的URL的hash结果来分配请求，使一个URL始终定向到同一个后端服务器 |
| less\_conn | 最少连接数，哪个机器连接数少，就分发 |
| hash关键数值 | hash自定义的key |

4、MYSQL  drop,delete与truncate的区别

      （1）drop直接删掉表 truncate删除表中数据，再插入时自增长id又从1开始 delete删除表中数据，可以加where字句。

      （2） DELETE语句执行删除的过程是每次从表中删除一行，并且同时将该行的删除操作作为事务记录在日志中保存以便进行进行回滚操作。TRUNCATE TABLE 则一次性地从表中删除所有的数据并不把单独的删除操作记录记入日志保存，删除行是不能恢复的。并且在删除的过程中不会激活与表有关的删除触发器。执行速度快。

      （3） 表和索引所占空间。当表被TRUNCATE 后，这个表和索引所占用的空间会恢复到初始大小，而DELETE操作不会减少表或索引所占用的空间。drop语句将表所占用的空间全释放掉。

      （4） 应用范围。TRUNCATE 只能对TABLE；DELETE可以是table和view

      （5） TRUNCATE 和DELETE只删除数据，而DROP则删除整个表（结构和数据）。

5、编码  列出目录下所有文件的全路径，保存到1维数组中

|  |
| --- |
| **class** A {      /\*\*       \*@param mix       \*@return  mix       \*\*/  **static** **public** function searchFile($dir) {          $ret = array ();  **if** ((! is\_dir ( $dir )) && (! is\_file ( $dir ))) {  **return** $ret;          } **else** **if** (is\_file ( $dir )) {              $ret [] = $dir;          } **else** {              $children = scandir ( $dir );              foreach ( $children as $child ) {  **if** ($child !== '.' && $child !== '..') {  **if** (empty ( $ret )) {                          $ret = self::searchFile ( $dir . DIRECTORY\_SEPARATOR . $child );                      } **else** {                          $ret = array\_merge ( $ret, self::searchFile ( $dir . DIRECTORY\_SEPARATOR . $child ) );                      }                  }              }          }  **return** $ret;      }  } |