

= 第三阶段技术面试知识点 =

Shell

1 自定义 Shell 变量时，变量名有什么规则？

参考答案

可以包括数字、字母、下划线，不能以数字开头

变量名区分大小写

赋值时等号两边不要有空格

尽量不要使用关键字和特殊字符

给同一个变量多次赋值时，最后一次的赋值生效

2 简述预定义变量 \$\$、\$?、\$0、\$#、\$*、\$! 的作用。

参考答案

\$\$ 保存当前运行进程的（PID）号

\$? 保存命令执行结果的（返回状态）

\$0 保存当前运行的（进程名）或（脚本名）

\$# 保存位置变量的（个数）

\$* 保存所有位置变量的（值）

#! 保存后台（最后一个进程）的 PID 号

3 简述三种定界符在变量赋值操作中的特点。

参考答案

双引号 " "：允许扩展，以 \$ 引用其他变量

单引号 ' '：禁用扩展，即便 \$ 也视为普通字符

反撇号 ` `：将命令的执行输出作为变量值

4 列出常见的整数值比较操作，并说明各自作用。

参考答案

-eq 等于（Equal）

-ne 不等于（Not Equal）

-ge 大于或等于（Greater or Equal）

-le 小于或等于（Lesser or Equal）

-gt 大于（Greater Than）

-lt 小于（Lesser Than）

5 简述 Shell 环境常见的中断及退出控制指令。

参考答案

break: 跳出当前所在的循环体, 执行循环体后的语句。

continue: 跳过循环体内余下的语句, 重新判断条件以便执行下一次循环。

exit: 退出脚本, 默认返回值是 0。

return: 用在函数里 指定返回值。

shift: 用于迁移位置变量, 将 \$1~\$9 依次向左顺序移动。

6 正则表达式中的+、?、*分别表示什么含义?

参考答案

这三个字符用来限制关键词的匹配次数, 含义分别如下:

+: 最少匹配一次, 比如 a+可匹配 a、aa、aaa 等

? : 最多匹配一次, 比如 a?可匹配零个或一个 a

*: 匹配任意多次, 比如 a*可匹配零个或任意多个连续的 a

7 简述 awk 工具常用的内置变量、各自的作用。

参考答案

FS: 保存或设置字段分隔符

\$n: 即\$1、\$2、\$3……, 表示指定分隔的第几个字段

\$0: 保存当前读入的整行文本内容

NF: 记录当前处理行的字段个数(列数)

NR: 记录当前已读入行的数量(行数)

FNR: 保存当前处理行在原文本内的序号(行号)

FILENAME: 保存 awk 当前处理的(文件名)

ENVIRON: 调用 Shell 环境变量, 格式: ENVIRON["变量名"]

8, 什么是 shell? 如何查看当前系统支持的 Shell?

答案:

1) 实现某种功能的, 有执行权限的文件 2) cat /etc/shells

9, 如何切换当前使用的 shell

答案: bash

10、/bin/bash 特性有哪些

答案: 特性: 提供命令补全, 命令编辑和命令历史表等功能

11、用户配置文件是哪几个?有什么用?

答案:

1) `~/.bashrc` `~/.bash_profile`

可以在里面定义变量，用户每打开一个终端时加载的文件，只针对用户有效

12、系统配置文件是哪几个? 有什么用?

`/etc/profile` `/etc/bashrc`

可以在里面定义变量，用户每打开一个终端时加载的文件，针对所有用户生效

13、shell 变量类型有哪些? 列举出常用的 4 个预定义变量

1) 1、自定义变量 2、系统环境变量 3、预定义变量 4、位置变量

2) `$?` `$#` `$*` `$$`

14、自定义变量的命名规则?

1、可以用数字字母下划线，但不能用数字开头和纯数字

2、同一个变量赋两个值，最后一个值生效

3、给变量赋值时，两边不能有空格

4、区分大小写

15、变量的生命周期，和作用范围?

1) 在脚本里定义的变量，只在脚本执行过程中有效

2) 默认情况下只在当前 Shell 里可以使用，要想定义的 Shell 在所有的 Shell 里被使用，要把变量定义为全局变量

16、脚本的执行过程? 脚本的执行方式?

1) 从左到右 从上到下

2) `sh xx.sh` `bash xx.sh` `/xx/xx/sh ./xx.sh`

17、编写脚本的步骤？

- 1、明确脚本要实现的功能
- 2、需要使用哪些命令
- 3、需要使用哪些流程控制
- 4、数据是变化的，就要用变量来表示

18、Shell 条件判断类型？

- 1、文件状态
- 2、数值比较
- 3、字符比较
- 4、逻辑比较

19、eval 在 shell 中有什么作用？请举个例子说明？

作用：

```
#!/bin/bash
aa=h1
bb=h2
var=(aa bb)
for vas in ${var[@]}
do
    echo $vas
    eval var_temp=\${vas}
    echo $var_temp
done
```

20、dirname 有什么作用？请举个例子说明？

作用：获取当前脚本路径

例子：

```
#!/bin/bash
cd `dirname $0`
pwd;ls
```

21、ssh -o stricthostkeychecking=no -o ConnectTimeout=60 分别代表什么意思？

ssh 加接时不需要输入 key 检查，然后六十秒为超时时间

22、说一下-n -z -d -f -e -r -w -x -l !-z 在 shell 中分别代表什么意思？ -n 和-z 试举一个例子说明

```
-n 非空为真
-z 字符串为空为真
-d 是目录且存在为真
-f 是普通文件且存在为真
-e 存在为真
-r 可读为真
-w 可写为真
-x 可执行为真
-l 是链接为真
!-z 不为空
1) -n 例子:
JavaApp=$(ps aux | grep "java" | grep "/app/guaji")
if [ -n "${JavaApp[0]}" ];then
    echo -e "\033[1;31m 脚本退出 \033[0m"
exit 1
fi
2) -z 例子:
ssh=$(ssh -o StrictHostKeyChecking=no $2 "ls /")
if [[ -z "$ssh" ]];then
echo -e "\033[1;31m \033[5m 远程主机($2)无法连接, 脚本退出. \033[0m"
exit 1
fi
```

23、正则表达式的 "|", "\n", "\r", "\t"分别代表什么意思

|表示或的意思
\n 表示换行，将当前位移置下行开头
\r 回车，将当前位置移至本行开头

24、写出获取公网 IP 的两种方法

```
ifconfig | grep -Po "(?<=addr:).*?(?=Bcast)" | egrep -v
'^192.168|^172.1[6-9].|^172.2[0-9].|^172.3[0-1].|^10.|^127.'
/sbin/ifconfig | awk -F'[ :]+' '/inet addr:/ {print $4}' | egrep -v
'^192.168|^172.1[6-9].|^172.2[0-9].|^172.3[0-1].|^10.|^127.'
```

25、\$\$表示什么意思？

获取当前执行脚本的进程名

26、函数内的 local 是什么意思?举一脚本例子说明

脚本中局部变量，存在于脚本函数(function)中的变量称为局部变量，要以 local 方式进行声明，使之只在本函数作用域内有效，

防止变量在函中的命名与变量外部程序中变量重名造成程序异常，下面是一脚本例子：

```
#!/bin/sh
a() {
    local s=1
    echo $s
}
b() {
    s=2
    echo $s
}
s=0
a
echo $s
b
echo $s
```

27、\$@和\$*的区别

\$@将命令行每个参数视为单独的字符串，等同于"\$1"、"\$2"、"\$3"

\$*将所有的命令行所有参数视为单个字符串，等同于"\$1\$2\$3"

28、\$0、dirname 和 basename 有什么作用？具个例子说明？

作用：

dirname 获取脚本路径名

basename 获取脚本名

\$0 获取脚本路径和脚本名

例子：

```
#!/bin/bash
echo "#####"
echo "# this is '$0' test#"
echo "#####"
```

```

echo $0
echo
echo "#####"
echo "# this is 'dirname' test#"
echo "#####@@@#"
cd `dirname $0` && pwd
echo
echo "#####"
echo "# this is 'basename' test#"
echo "#####"
echo `basename $0`

```

29、取出 3306 端口号这个数字

`netstat -tlnp | grep 3306 | awk '[:]+ ' {print $5}' #[:]+` 以空格和点为分隔符，这种分隔符有一个或多个

30、shell 中单引号和双引号与不加与号分别代表什么意思

单引号：可以说是所见即所得，即将单号引号的内容原样输出

双引号：把双引号的内容输出出来，如果内容中有命令、变量等，会先把变量、命令解析出结果，然后再输出最终内容来

无引号：把内容输出出来，会将含有空格的字符串视为一个整体输出，如果内容中有命令、变量等，会先把变量、命令解析出结果，

然后再输出内容来，如果字符串中带有空格等特殊字符，则不能完整的输出，需要改加双引号，一般连续的字符串，数字

路径等可以不加任何引号，不过最好用双引号替代之

31、echo -n 是什么意思？举一脚本例子显示他的功能

```

-n 不换行
#!/bin/bash
. /etc/init.d/functions
echo "-----Please wait while we are checking-----"
echo -n "6 秒后执行该操作."
for ((i=0;i<6;i++))
do
    echo -n ".";sleep 2
    #[ $i -eq 3 ] && break
done
echo

```

```
action "test success" /bin/true
```

32、echo -e "\n" | nohup ping www.baidu.com > filename 2>&1 & 这里的-e 和 "\n" 是什么意思？

-e 是代表后面输入的内容包含特殊字符需要加的参数
\n 是代表换行

33、有 1-9 的数字，echo 输出 1-9 的数字时，屏蔽 0-5 的数字再输出

```
echo '1234567890' | sed 's/[0-5]//g'
```

34、用 nohup 把一个脚本放在后台运行

```
nohup sh test.sh > filename 2>&1 &
```

35、用 seq 打印 10 以空格分隔，用 seq 竖着打印 10 到 1

```
seq -s " " 10  
seq 10 -1 1
```

36、curl -I -s www.baidu.com | head -1 | wc -l 中的-I 和-s 代表什么意思

-s 沉默或安静模式。不显示进度表或错误消息。使卷曲静音
-I 只读取 HTTP 头！ HTTP 的服务器功能命令 HEAD 此用来获取
--connect-timeout 2 2 秒连接超时

37、wget -T 10 -c -q --spider www.baidu.com 这些参数分别代表什么意思？

-T 超时 10 秒
-c 断点续传
-q 为执行命令时屏蔽他打印输出的意思，执行命令不会任何内容出来
--spider wget 命令加 spider 参数不会下载任何东西，spider 的主要作用是测试下载链接。

tries

38、生产环境常用的监控 web 的方法是哪种

用 wget 或 curl 的方法作监控

39、生产环境写脚本报错就发邮件的命令是怎么样

```
mail -s "uname -n 's httpd status is on" 123456@qq.com < $logfile
```

40、给文件改名

```
mv $file llinux-`echo $file | cut -d "-" -f2`  
ls *.jpg | awk -F '_finished' '{print "mv " " $0" "$1".jpg"}' | bash
```

41、查看当前目录的所有文件夹

```
ls -F | grep /
```

42、awk 中的 NF 代表什么意思？

NF 代表结尾的意思

43、shell 上: 0、1>、2>、>、2>&1、&>/dev/null 分别代表什么意思

0 表示标准输入

1>表示标准输出

2>表示标准错误输出

> 默认为标准输出重定向，与 1> 相同

2>&1 意思是把 标准错误输出 重定向到 标准输出.

&>/dev/null 意思是把 标准输出 和 标准错误输出 都重定向到空

OPERATION

1 简述 Squid 的优缺点有哪些

参考答案

优势：

通过缓存增加访问速度；

提高网络的安全性；

Squid 历史悠久，技术积淀深厚，同时无需支付任何费用。

缺点：

性能不足，借助于 windows、linux 等通用操作系统才能实现上网加速等功能。

2 简述常见 Web 服务器软件有哪些

参考答案

Apache、Nginx、Lighttpd、Tomcat、IBM WebSphere、IIS。

3 简述什么是 LNMP

参考答案

LNMP 是网站运行平台。

L：操作系统（Linux）

N：网站服务软件（Nginx）

M：数据库服务软件（MySQL）

P：网站开发语言（PHP、perl、python）

4 地址重写的好处

参考答案

缩短 URL，隐藏实际路径提高安全性；

易于用户记忆和键入；

易于被搜索引擎收录。

5 简述什么是 memcached

参考答案

memcached 是高性能的分布式缓存服务器，是一个跨平台的、开源的实现分布式缓存服务的软件
用来集中缓存数据库查询结果，减少数据库访问次数，以提高动态 Web 应用的响应速度

memcached 支持许多平台：Linux、FreeBSD、Solaris（memcached 1.2.5 以上版本）、Mac OS X、Windows

2 简述 memcached 什么情况下会删除缓存数据

参考答案

当分配的存储空间存满时，有新数据需要存储时，删除最近最少使用的数据。腾出空间存储新数据；
重新启动 memcached 服务会删除缓存数据；
运行 memcached 服务的操作系统重启后会删除缓存数据。

6 部署实施 CDN 的优势是什么

参考答案

本地 Cache 加速，提升访问速度和稳定性
网站镜像，消除不同地区、运营商网络互连的问题
智能 DNS 解析，自动选择最快的 Cache 服务器
负载优化，减轻后端源 Web 服务器的压力负载
集群抗攻击，有效降低 DDOS 攻击的影响

6，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

7 Nginx 和 Apache 的综合对比？

Web 服务器	Nginx	Apache
反向代理	非常好	好
Rewrite 规则	非常好	好
FastCGI	好	差
热部署	支持	不支持
系统压力比较	很小	小
稳定性	非常好	好
安全性	一般	好
技术资料	很少	非常多
静态文件处理	非常好	一般
虚拟主机	支持	支持
内存消耗	非常小	很大

8，什么是 CDN？

答案：CDN 的全称是内容分发网络。其目的是通过在现有的 Internet 中增加一层新的网络架构，将网站的内容发布到最接近用户的网络“边缘”，使用户可以就近取得所需的内容，提高用户访问网站的响应速度。CDN 有别于镜像，因为它比镜像更智能，或者可以做这样一个比喻：CDN=更智能的镜像+缓存+流量导流。因而，CDN 可以明显提高 Internet 网络中信息流动的效率。从技术上全面解决由于网络带宽小、用户访问量大、网点分布不均等问题，提高用户访问网站的响应速度。

9，CDN 的类型特点？

答案：CDN 的实现分为三类：镜像、高速缓存、专线。

（1）**镜像站点（Mirror Site）**：是最常见的，它让内容直接发布，适用于静态和准动态的数据同步。但是购买和维护新服务器的费用较高还必须在各个地区设置镜像服务器，配备专业技术人员进行管理与维护。对于大型网站来说，更新所用的带宽成本也大大提高了。

（2）**高速缓存**：成本较低，适用于静态内容。Internet 的统计表明，超过 80%的用户经常访问的是 20%的网站的内容，在这个规律下，缓存服务器可以处理大部分客户的静态请求，而原始的服务器只需处理约 20%左右的非缓存请求和动态请求，于是大大加快了客户请求的响应时间，并降低了原始服务器的负载。

（3）**专线**：让用户直接访问数据源，可以实现数据的动态同步。

10 CDN 的工作原理？

答案：

若是忽略各种缓存，传统的网站访问过程为：

1. 用户在浏览器中输入要访问的域名；
2. 浏览器向域名解析服务器发出解析请求，获得此域名对应的 IP 地址；
3. 浏览器利用所得到的 IP 地址，向该 IP 对应的服务器发出访问请求；
4. 服务器对此响应，将数据回传至用户浏览器端显示出来。

与传统访问方式不同，CDN 网络则是在**用户和服务器之间增加 Cache 层**，将用户的访问请求引导到 Cache 节点而不是服务器源站点，要实现这一方式，主要是通过接管 DNS 实现。

使用 CDN 缓存后的网站访问过程演变为：

1. 用户在浏览器中输入要访问的域名；

2. 浏览器向域名解析服务器发出解析请求，由于 CDN 对域名解析过程进行了调整，所以用户端一般得到的是该域名**对应的 CNAME 记录**，此时浏览器需要再次对获得的 CNAME 域名进行解析才能得到缓存服务器实际的 IP 地址。在此过程中，全局负载均衡 DNS 解析服务器会根据用户端的源 IP 地址，如地理位置（深圳还是上海）、接入网类型（电信还是网通）将用户的访问请求定位到离用户路由最短、位置最近、负载最轻的 Cache 节点（缓存服务器）上，实现就近定位。定位优先原则可按位置、可按路由、也可按负载等等；
3. 再次解析后浏览器得到该域名 CDN 缓存服务器的实际 IP 地址，向缓存服务器发出访问请求；
4. 缓存服务器根据浏览器提供的域名，通过 **Cache 内部专用 DNS 解析得到此域名源服务器的真实 IP 地址，再由缓存服务器向此真实 IP 地址提交访问请求；**
5. 缓存服务器从真实 IP 地址得到内容后，一方面在本地进行保存，以备以后使用，同时把得到的数据发送到客户端浏览器，完成访问的响应过程；
6. 用户端得到由缓存服务器传回的数据后显示出来，至此完成整个域名访问过程；

不论是否使用 CDN 网络，用户客户端设置不需做任何改变，直接使用被加速网站原有域名访问即可。对于要加速的网站，只需修改整个访问过程中的域名解析部分，便能实现透明的网络加速服务。
