

怎么挂载 windows 的共享目录?

mount.cifs //IP/SHARE linux 的目录 --verbose -o user=username <--这个用户是 windows 下的用户--verbose 这个参数可以不加,它是显示过程的

例如 mount.cifs //10.1.1.246/gongxiang /mnt --verbose -o user=gao

或者是 mount -t cifs

umount /mnt 或 umount.cifs /mnt -l <--取消挂载

图形界面: smb://IP

A B 网络是通的,最少列出五种传输文件的服务

nfs,ftp,scp,rsync,samba,http://

1.假设 Apache 产生的日志文件名为 access_log,在 apache 正在运行时,执行命令 mv

access_log access_log.bak,执行完后,请问新的 apache 的日志会打印到哪里,为什么?

新的日志会打印在 access_log.bak 中,因为 apache 启动时会找到 access_log 文件,随时准备向文件中加入日志信息,

虽然此时文件被改名,但是由于服务正在运行,因为它的 inode 节点的位置没有变,程序打开的 fd 仍然会指向原来那个 inode,

不会因为文件名的改变而改变。apache 会继续向已改名的文件中追加日志,但是若重启 apache 服务,系统会检查 access_log

文件是否存在,若不存在则创建。

2.在 Shell 环境下,如何查看远程 Linux 系统运行了多少时间?

2、监控主机执行: ssh user@被监控主机 ip "uptime"

这样得到了被监控主机的 uptime

3.处理以下文件内容,将域名取出并进行计数排序,如处理:

http://www.baidu.com/index.html

http://www.baidu.com/1.html

http://post.baidu.com/index.html

http://mp3.baidu.com/index.html

http://www.baidu.com/3.html

http://post.baidu.com/2.html

得到如下结果:

域名的出现的次数 域名

3 www.baidu.com

2 post.baidu.com

1 mp3.baidu.com

可以使用 bash/perl/php/c 任意一种

3、[root@localhost shell]# cat file | sed -e 's/http://\\/' -e 's/\\/\\.*/' | sort | uniq -c | sort -rn

3 www.baidu.com

2 post.baidu.com

1 mp3.baidu.com

```
[root@codfei4 shell]# awk -F/ '{print $3}' file |sort -r|uniq -c|awk '{print $1"\t",$2}'
```

```
3      www.baidu.com
2      post.baidu.com
1      mp3.baidu.com
```

4.如果得到随机的字串,长度和字串中出现的字符表可定义,并将字串倒序显示,如把 0123456789 作为基准的字串字符表,产生一个 6 位的字串 642031,打印出的字串为

130246,可使用 bash/perl/php/c 任意一种.

```
4 、 [root@localhost ~]# awk -v count=6 'BEGIN
{ srand();str="0123456789";len=length(str);for(i=count;i>0;i--)
marry[i]=substr(str,int(rand()*len),1);for(i=count;i>0;i--)
printf("%c",marry[i]);printf("\n");for
```

```
(i=0;i<=count;i++) printf("%c",marry[i]);printf("\n"))}'
```

```
838705
```

```
507838
```

5.如何查看当前 Linux 系统的状态,如 CPU 使用,内存使用,负载情况等.

5、Linux 系统中“/proc”是个伪文件目录,不占用系统空间,及时的反应出内存现在使用的进程情况.....其中许多文件都保存系统运行状态和相关信息

对于“/proc”中文件可使用文件查看命令浏览其内容,文件中包含系统特定信息:

cpuinfo	主机 CPU 信息
filesystems	文件系统信息
meninfo	主机内存信息
version	Linux 内存版本信息
diskstatus	磁盘负载情况

另外 top 命令可以动态的显示当前系统进程用户的使用情况,而且是动态的显示出来,尤其是在该命令显示出来的对上方对系统的情况进行汇总.

free 命令呢可以查看真实使用的内存 一般用 free -m

使用 lsof 、 ps -aux 可以查看详细的每个进程的使用状况

dmesg 也是常用来查看系统性能的命令

```
#####
#####
#####
##
```

#题目: 有 10 台被监控主机、一台监控机,在监控机上编写脚本,一旦某台被监控机器/分区适用率大于 80%, 就发邮件报警放到 crontab 里面, 每 10 分钟检查一次

#测试机器: 虚拟机 Linux as 4

#1.首先建立服务器间的信任关系.拿两台机器做测试

本机 ip:192.168.1.6

```
[root@codfei ~]# ssh-keygen -t rsa
```

Generating public/private rsa key pair.

```

Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y (以为我是第 2 次建立关系所以此处覆盖原来的文件)
Enter passphrase (empty for no passphrase):(直接回车无须输入密钥)
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
04:37:13:2a:4b:10:af:c1:2b:03:3f:6b:27:ce:b9:62 root@codfei
[root@codfei ~]# cd .ssh/
[root@codfei .ssh]# ll
-rw----- 1 root root 883 Apr 25 17:51 id_rsa
-rw-r--r-- 1 root root 221 Apr 25 17:51 id_rsa.pub
-rw-r--r-- 1 root root 442 Apr 25 17:37 known_hosts
id_rsa 是密钥文件，id_rsa.pub 是公钥文件。
[root@codfei .ssh]# scp id_rsa.pub 192.168.1.4:/root/.ssh/192.168.1.6
root@192.168.1.4's password:
id_rsa.pub                                100% 221      0.2KB/s
00:00
这里把公钥文件取名为本机的 ip 地址就是为了以后和更多的机器建立信任关系
不发生混淆。
现在登陆到 192.168.1.4 机器
[root@codfei ~]# cd .ssh/
[root@codfei .ssh]# cat 192.168.1.6 >> authorized_keys
然后回到 192.168.1.6 机器直接
[root@codfei .ssh]# ssh 192.168.1.4
Last login: Wed Aug  8 12:14:42 2007 from 192.168.1.6
这样就可以了，里面偶尔涉及到权限问题。一般 .ssh 文件夹是 755
authorized_keys 为 600 或者 644

```

####脚本如下#####

```

#!/bin/bash
#SCRIPT:df_check.sh
#Writeen by codfei Mon Sep  3 07:25:28 CST 2007
#PURPOSE:This script is used to monitor for full filesystems.
#####Begining#####
#####
FSMAX="80"
remote_user='root'  #####完全可以不用 root
remote_ip=(192.168.1.5 192.168.1.6 192.168.1.7 192.168.1.8 192.168.1.9
192.168.1.10 192.168.1.11 192.168.1.12 192.168.1.13 192.168.1.14 )  ---->
这里填写你要监控的主机 ip

```

```

ip_num='0'
while [ "$ip_num" -le "$(expr ${#remote_ip[@]} - 1)" ]
do
    read_num='1'
    ssh "$remote_user"@"${remote_ip[$ip_num]}" df -h > /tmp/diskcheck_tmp
    grep '^/dev/*' /tmp/diskcheck_tmp|awk '{print $5}'|sed 's/\%//g' >
/tmp/diskcheck_num_tmp
        while [ "$read_num" -le $(wc -l < /tmp/diskcheck_num_tmp) ]
        do
            size=$(sed -n "$read_num"p' /tmp/diskcheck_num_tmp)
            if [ "$size" -gt "$FSMAX" ]
            then
                $(grep '^/dev/*' /tmp/diskcheck_tmp|sed -n $read_num'p' >
/tmp/disk_check_mail)
                $(echo ${remote_ip[$ip_num]} >> /tmp/disk_check_mail)
                $(mail -s "diskcheck_alert" admin < /tmp/disk_check_mail)
            fi
            read_num=$(expr $read_num + 1)
        done
    ip_num=$(expr $ip_num + 1)
done

```

#####over#####
#####让脚本每十分钟执行一次#####
在 cron 表中加入
0/10 * * * * /home/codfei/diskcheck.sh 2>&1

#####

比如， ext2 文件系统， 如果异常死机， 开机如何修复文件系统？

如果异常关机， 比如断电， 通知机房的人开机之后，

我们需要远程修复、 检查文件系统

除了/分区之外， 其他的分区：

umount /home

fsck -y /home

/ 分区需要开机之后， 由机房的人来扫描

随后我们再登录并扫描/home 等其他分区

如何查看一个进程所使用的文件句柄？

看这里面 /proc/进程号/fd/

的个数就行了

简单的比如如何查看 apache 进程数

[root@localhost fd]# ps -ef|grep httpd|wc -l

1

如何统计 apache 的每秒访问数?

```
tail access_log | awk '{print $1,$4}'
```

```
[root@localhost logs]# grep -c `date -d '3 second ago' +%T` access_log
```

0

```
#####  
#####  
#####  
#####  
#####  
#####  
#####
```

1、/proc/sys 子目录的作用

该子目录的作用是报告各种不同的内核参数，并让您能交互地更改其中的某些。与 /proc 中所有其他文件不同，该目录中的某些文件可以写入，不过这仅针对 root。

其中的目录以及文件的详细列表将占据过多的篇幅，而且该目录的内容是依赖于系统的，而大部分的文件也仅仅对某些特殊的应用程序有用。然而，以下是该子目录的两个最常见的用途：

允许路由：即便是 Mandrakelinux 默认的内核也是允许路由的，您必需显式允许它这么做。为此，您只要以 root 身份键入以下命令：

```
$ echo 1 >/proc/sys/net/ipv4/ip_forward
```

如果您要禁用路由，请将上述命令中的 1 改为 0。

阻止 IP 欺骗：IP 欺骗会让人认为某个来自于外部的某个数据包是来自于它到达的那个接口。这一技术常被骇客(cracker)所使用。您可以让内核阻止这种入侵。请键入：

```
$ echo 1 >/proc/sys/net/ipv4/conf/all/rp_filter
```

这样，这种攻击就不再可能了。

这些改变仅当系统运行时有效。在系统重新启动之后，它们会改会它们的默认值。要在启动时就改动这些值，您可以将您在 shell 提示符后键入的命令添加到 /etc/rc.d/rc.local 中以免每次都键入它们。另一个方法是修改

```
/etc/sysctl.conf
```

2、将一个文本的奇数行和偶数行合并，第 2 行和第 3 行合并

```
[root@localhost bin]# cat 1
```

```

48      Oct      3bc1997 lpas      68.00      lvx2a      138
484     Jan      380sdf1 usp      78.00      deiv      344
483     nov      7pl1998 usp      37.00      kvm9d      644
320     aug      der9393 psh      83.00      wiel      293
231     jul      sdf9dsf sdfs      99.00      werl      223
230     nov      19dfd9d abd      87.00      sdiv      230
219     sept     5ap1996 usp      65.00      lvx2c      189
216     Sept     3zl1998 usp      86.00      kvm9e      234
[root@localhost bin]# sed '$!N;s/\n/ /g' 1
48      Oct      3bc1997 lpas      68.00      lvx2a      138      484      Jan
380sdf1 usp      78.00      deiv      344
483     nov      7pl1998 usp      37.00      kvm9d      644      320      aug
der9393 psh      83.00      wiel      293
231     jul      sdf9dsf sdfs      99.00      werl      223      230      nov
19dfd9d abd      87.00      sdiv      230
219     sept     5ap1996 usp      65.00      lvx2c      189      216      Sept
3zl1998 usp      86.00      kvm9e      234
[root@localhost bin]# sed -n -e 2p -e 3p 1|sed '$!N;s/\n/ /'
484     Jan      380sdf1 usp      78.00      deiv      344      483      nov
7pl1998 usp      37.00      kvm9d      644

```

3、read 命令 5 秒后自动退出

```
[root@localhost bin]# read -t 5
```

4、自动 ftp 上传

```
#!/bin/sh
```

```
ftp -n<<END_FTP
```

```
open 192.168.1.4
```

```
user codfei duibuqi //用户名 codfei 密码 duibuqi
```

```
binary
```

```
prompt off //关闭提示
```

```
mput test //上传 test
```

```
close
```

```
bye
```

```
END_FTP
```

自动 ssh 登陆 从 A 到 B 然后再到 c

```
#!/usr/bin/expect -f
```

```
set timeout 30
```

```
spawn ssh codfei@B
```

```
expect "password:"
```

```
send "pppppp\r"
```

```
expect "]"
```

```
send "ssh codfei@C\r"
```

```
expect "password:"
```

```
send "pppppp\r"
```

```
interact
```

5、#打印第一个域

```
[root@localhost bin]# cat 3
eqeqedadasD
eqeqdadfdfDD
fdsfsfQWEDD
DSADASDSADSA
[root@localhost bin]#
[root@localhost bin]#
[root@localhost bin]# awk -F "" '{print $1}' 3
e
e
f
D
```

6、实现字符串翻转

```
[root@localhost bin]# cat 8
qweqewqedadaddas
[root@localhost bin]# rev 8
saddadadeqweqewq
```

#####第 2 次电面

7、sed awk grep 哪个最好

我答的是 哪个掌握的精通，都很好，但是还是问我哪个最好，我只能说 awk 了，对于行操作和列操作都可以操作的很好。

8、grep -E -P 是什么意思

我说的是-E, --extended-regexp 采用规则表示式去解释样式。 -P 不太清楚

9、还问了我对运维这个工作的理解，和应该具备的素质。

.....

腾讯系统工程师面试题

shell 脚本编程部分：

1. 从 a.log 文件中提取包含“WARNING”或“FATAL”,同时不包含“IGNOR”的行，然后，提取以“:”分割的第五个字段？

```
grep -E 'WARNING|FATAL' a.log | grep -v 'IGNOR' | awk -F : '{print $5}'
```

2. 添加一个新组为 class01,然后，添加属于这个组的30个用户，用户名的形式为 stdXX,其中，XX 从 01到 30？

```
#!/bin/bash
groupadd class01
for((i=1;i<=9;i++))
do
useradd std0$i -G class01
done

for((i=10;i<=30;i++))
do
useradd std$i -G class01
done
```

3. 在每个月的第一天备份并压缩/etc 目录下的所有内容，存放在/root/backup 目录里，且文件名为如下形式 yymmdd_etc.yy 为年，mm 为月，dd 为日。shell 程序 fileback 存放在 /usr/bin 目录下？

```
tar -zcf /root/backup/`date+%y%m%d`_etc /etc
```

4. 用 shell 编程，判断一文件是不是字符设备文件，如果是将其拷贝到/dev 目录下？

```
#!/bin/bash
if test $# -ne 1
then
echo "ERROR"
exit 1
fi
if test -c $1
then
/bin/cp -a $1 /dev/
exit 0
fi
exit 1
```

5. 某系统管理员需要每天做一定的重复工作，编制一个解决方案：

- (1).从下午 4：50 删除/abc 目录下的全部子目录和全部文件；
- (2).从早上 8：00～下午 6：00每小时读取/xyz 目录下 x1 文件中每行第一个域的全部数据加入到/backup 目录下的 back01.txt 文件内；
- (3).每逢周一下午 5：50 将/data 目录下的所有目录和文件归档并压缩为文件 backup.tar.gz;
- (4).在下午 5：55 将 IDE 接口的 CD—ROM 卸载（假设 CD—ROM 的设备名为 hdc）；
- (5).在早上 8：00 开机后启动。

使用计划任务来完成：

crontab-e

50 16 *** rm-rf/abc/*

00 08-18 *** awk '{print\$1 }' /xyz/x1 >> /backup/back01.txt

50 17 ** 01 tar-zcf/backup/backup.tar.gz /data

55 17 *** umount/dev/hdc

设定 BIOS 每日上午 08:00开机

1、简述 Apache 两种工作模式，以及它们之间的区别。

答案：最主要的两种模式是 prefork 模式与 worker 模式。prefork 每个子进程只有一个线程，效率高但消耗内存大，是 unix 下默认的模式；worker 模式每个子进程有多个线程，内存消耗低，但一个线程崩溃会牵连其它同子进程的线程。

2、用 iptables 添加一个规则允许 192.168.0.123 访问本机 3306 端口
iptables -I INPUT 1 -p tcp -m tcp --dport 3306 -s 192.168.0.123 -j ACCEPT

3、如何对一台 Linux 服务器进行系统性能调优，列举出参数。

4、DNS 服务器的工作原理。

5、修改第一块网卡的路径是什么。
/etc/sysconfig/network-scripts/ifcfg-eth0

7、使用 shell，建立 class1 用户组，再批量建立 stu1--stu30 的用户，并指定用户组为 class1。

```
vi autoaddusr
#!/usr/bin/php -q
<?php
exec("groupadd class1");
for($i=1; $i<=30; $i++){
    exec("useradd -G class1 stu".$i);
}
?>
chmod +x autoaddusr
./autoaddusr
```

8、个人对该工作的未来如何规划，需要加强哪些能力。

首先，我有一颗真诚的心，遇事沉着冷静，不急不躁；
其次，我有相应的专业知识和工作经验。一年多的系统管理经验锻炼了我在这个行业的业务能力，并对行业前景和发展动态有相应的了解；
最后，我会用踏实的作风在今后的工作中证明我自己的能力！

9、日常监控都需要监控哪些？

1)硬件：

CPU：/proc/cpuinfo

内存：/proc/meminfo

硬盘：fdisk -l

2)系统：

负载：/proc/loadavg

uptime 查看实时 load average、swap

虚拟内存：vmstat（参数-s；2 4）

SUID,用户,进程

系统日志：tail -f /var/log/messages

logwatch --print --range Today --service SSHD --service pam_unix

3)网络：Host_Alive,Ping,端口,连接

.如何将本地 80 端口的请求转发到 8080 端口,当前主机 IP 为 192.168.16.1,其中本地网卡 eth0:

答：

```
#iptables -t nat -A PREROUTING -d 192.168.16.1 -p tcp -dport 80 -j DNAT -to 192.168.16.1:8080
```

或者：

```
#iptables -t nat -A PREROUTING -i eth0 -d 192.168.16.1 -p tcp -m tcp -dport 80 -j REDIRECT -to-ports 8080
```

2.什么是 NAT,常见分为那几种，DNAT 与 SNAT 有什么不同，应用事例有那些？

SNAT，DNAT，MASQUERADE 都是 NAT。

MASQUERADE 是 SNAT 的一个特例。

SNAT 是指在数据包从网卡发送出去的时候，把数据包中的源地址部分替换为指定的 IP，这样，接收方就认为数据包的来源是被替换的那个 IP 的主机。

MASQUERADE 是用发送数据的网卡上的 IP 来替换源 IP，因此，对于那些 IP 不固定的场合，比如拨号网络或者通过 dhcp 分配 IP 的情况下，就得用 MASQUERADE。

DNAT，就是指数据包从网卡发送出去的时候，修改数据包中的目的 IP，表现为如果你想访问 A，可是因为网关做了 DNAT，把所有访问 A 的数据包的目的 IP 全部修改为 B，那么，你实际上访问的是 B

因为，路由是按照目的地址来选择的，因此，DNAT 是在 PREROUTING 链上来进行的，而 SNAT 是在数据包发送出去的时候才进行，因此是在 POSTROUTING 链上进行的。

3.包过滤防火墙与代理应用防火墙有什么区别，能列举几种相应的产品吗？

包过滤防火墙是根据包头进行的过滤，并且是处于网络层的，根据包的源 ip 地址，目标 ip 地址，协议类型，端口号，进行过滤；代理应用防火墙工作在应用层，他使用代理服务器技术，将内网对外网的访问，变为防火墙对外网的访问，可以对包的内容进行分辨，从而过滤。

代理应用防火墙：天融信 GFW4000

包过滤防火墙：华为 NE 16E

4.iptables 是否支持 time 时间控制用户行为，如有请写出具体操作步骤。

支持。需要增加相关支持的内核补丁，并且要重新编译内核。

或者使用 crontab 配合 iptables，首先：vi /deny.bat 输入/sbin/iptables -A OUTPUT -p tcp -s 192.168.1.0/24 --dport 80 -j DROP 保存退出

打开 crontab -e

输入：00 21 * * * /bin/sh /deny.bat

5.说出你知道的几种 linux/unix 发行版本。

Redhat、CentOS、Fedora、SuSE、Slackware、Gentoo、Debian、Ubuntu、FreeBSD、Solaris、SCO、AIX、HP...

6.列出 linux 常见打包工具并写相应解压缩参数(至少三种)

Tar gz bz

Tar -xvzf gzip -d bzip2 -d

7.计划每星期天早 8 点服务器定时重启，如何实现？

Crontab -e

00 08 * * 7 /sbin/init 6

8.列出作为完整邮件系统的软件,至少二类。

Sendmail,postfix,qmail

9，当用户在浏览器当中输入一个网站，说说计算机对 dns 解释经过那些流程？

注：本机跟本地 dns 还没有缓存。

答：

a.用户输入网址到浏览器；

b.浏览器发出 DNS 请求信息；

c.计算机首先查询本机 HOST 文件，看是否存在，存在直接返回结果，不存在，继续下一步；

d.计算机按照本地 DNS 的顺序，向合法 dns 服务器查询 IP 结果；

e.合法 dns 返回 dns 结果给本地 dns，本地 dns 并缓存本结果，直到 TTL 过期，才再次查询此结果；

f.返回 IP 结果给浏览器；

g.浏览器根据 IP 信息，获取页面；

10，我们都知道，dns 既采用了 tcp 协议，又采用了 udp 协议，什么时候采用 tcp 协议？什么时候采用 udp 协议？为什么要这么设计？

答：这个题需要理解的东西比较的多，分一下几个方面

a，从数据包大小上分：UDP 的最大包长度是 65507 个字节，响应 dns 查询的时候数据包长度超过 512 个字节，而返回的只要前 512 个字节，这时名字 解释器通常使用 TCP 从发原来的请求。

b，从协议本身来分：大部分的情况下使用 UDP 协议，大家都知道 UDP 协议是一种不可靠的协议，dns 不像其它的使用 UDP 的 Internet 应用（如：TFTP, BOOTP 和 SNMP 等），大部分集中在局域网，dns 查询和响应需要经过广域网，分组丢失和往返时间的不确定性在广域网比局域网上更大，这就要求 dns 客户端需要好的重传和超时算法，这时候使用 TCP。

11，一个 EXT3 的文件分区，当使用 touch test.file 命令创建一个新文件时报错，报错的信息是提示磁盘已满，但是采用 df -h 命令查看磁盘大小时，只使用了，60%的磁盘空间，为什么会出现这个情况，说说你的理由。

答：两种情况，一种是磁盘配额问题，另外一种就是 EXT3 文件系统的设计不适合很多小文件跟大文件的一种文件格式，出现很多小文件时，容易导致 inode 耗尽了。

12，我们都知道 FTP 协议有两种工作模式，说说它们的大概的一个工作流程？

FTP 两种工作模式：主动模式（Active FTP）和被动模式（Passive FTP）

在主动模式下，FTP 客户端随机开启一个大于 1024 的端口 N 向服务器的 21 号端口发起连接，然后开放 N+1 号端口进行监听，并向服务器发出 PORT N+1 命令。

服务器接收到命令后，会用其本地的 FTP 数据端口（通常是 20）来连接客户端指定的端口 N+1，进行数据传输。

在被动模式下，FTP 客户端随机开启一个大于 1024 的端口 N 向服务器的 21 号端口发起连接，同时会开启 N+1 号端口。然后向服务器发送 PASV 命令，通知服务器自己处于被动模式。服务器收到命令后，会开放一个大于 1024 的端口 P 进行监听，然后用 PORT P 命令通知客户端，自己的数据端口是 P。客户端收到命令后，会通过 N+1 号端口连接服务器的端口 P，然后在两个端口之间进行数据传输。

总的来说，主动模式的 FTP 是指服务器主动连接客户端的数据端口，被动模式的 FTP 是指服务器被动地等待客户端连接自己的数据端口。

被动模式的 FTP 通常用在处于防火墙之后的 FTP 客户访问外界 FTP 服务器的情况，因为在这种情况下，防火墙通常配置为不允许外界访问防火墙之后主机，而只允许由防火墙之后的主机发起的连接请求通过。

因此，在这种情况下不能使用主动模式的 FTP 传输，而被动模式的 FTP 可以良好的工作。

13.编写个 shell 脚本将当前目录下大于 10K 的文件转移到/tmp 目录下

Q：主要是考察 awk 这些的用法

```
#!/bin/sh
#Programm :
# Using for move currently directory to /tmp
for FileName in `ls -l |awk '$5>10240 {print $9}'`
do
mv $FileName /tmp
done
```

```
ls -la /tmp
```

```
echo "Done! "
```

14. apache 有几种工作模式，分别介绍下其特点，并说明什么情况下采用不同的工作模式？

apache 主要有两种工作模式：prefork(apache 的默认安装模式)和 worker(可以在编译的时候加参数-with-mpm- worker 选择工作模式)

prefork 的特点是：(预派生)

1. 这种模式可以不必在请求到来时再产生新的进程，从而减小了系统开销

2. 可以防止意外的内存泄漏

3. 在服务器负载下降的时候会自动减少子进程数

worker 的特点是：支持混合的多线程多进程的多路处理模块

如果对于一个高流量的 HTTP 服务器，worker MPM 是一个比较好的选择，因为 worker MPM 占用的内存要比 prefork 要小。

15. 名词解释 HDLC,VTP,OSPF,RIP,DDOS,system V,GNU,netscreen,ssh,smartd,apache,WAIT_TIME 等等。

HDLC:高级链路控制；VTP: vlan 传输协议；OSPF: 开放最短路径优先；RIP: 路由信息协议；DDOS: 分布式拒绝服务攻击；system V: UNIX 版本 V；GNU: 非 UNIX，开放源码软件工程；netscreen: 国际著名防火墙厂家之一，04 年后被 juniper 收购，变为其防火墙的一个系列；ssh: 安全外壳，防止中间人攻击的一种连接方式；smartd: 硬盘检测工具 s.m.a.r.t 的进程；Apache: web 服务器软件；WAIT_TIME: netstat 命令显示的参数，客户端正在等待。

16. 编写 shell 脚本获取本机的网络地址。比如：本机的 ip 地址是：

192.168.100.2/255.255.255.0，那么它的网络地址是

192.168.100.1/255.255.255.0

方法一：

```
#!/bin/bash
```

```
#This script print ip and network
```

```
file="/etc/sysconfig/network-scripts/ifcfg-eth0"
```

```
if [ -f $file ];then
```

```
IP=`grep "IPADDR" $file|awk -F"=" '{ print
```

```
$2 }`
```

```
MASK=`grep "NETMASK" $file|awk -F"="
```

```
{ print $2 }`
```

```
echo "$IP/$MASK"
```

```
exit 1
```

```
fi
```

方法二：

```
#!/bin/bash
```

```
#This program will printf ip/network
```

```
#
```

```
IP=`ifconfig eth0 |grep 'inet ' |sed 's/^.*.addr://g'|sed 's/ Bcast.*$//g`
```

```
NETMASK=`ifconfig eth0 |grep 'inet ' |sed 's/^.*.Mask://g`
```

```
echo "$IP/$NETMASK"
```

```
exit
```

17.在命令行下发一邮件，发件人：123@abc.com, 收信人：abc@xyz.com

Mail -s "hello"

二、简述题：

1.linux 下如何改 IP,主机名，DNS

Setup

2.linux 下如何添加路由

Route add -net 10.1.1.0/24 eth1

3.简述 linux 下编译内核的意义与步骤

编译内核的意义在于让硬件设备更稳定的发挥其应有的效能；

4.简述 Linux 启动过程

Bios 引导--》启动引导工具 grub--》核心初始化--》载入初始程序 init--》init 初始化--》从 inittab 中读取数据，决定启动级别--》系统运行

5.简述 DDOS 攻击的原理

黑客劫持大量傀儡主机，对目标服务器进行合理的资源请求，导致服务器资源耗尽而不能进行正常的服务。

6.简述 Tcp 三次握手的过程

第一次握手，建立连接，客户端发送 SYN 包到服务器，并进入 SYN_SEND 状态，等待服务器确认；

第二次握手，服务器收到 SYN，同时自己也发送一个 SYN 包和一个 ACK 包来确认客户端的 SYN，并进入 SYN_RECV；

第三次握手，客户端收到服务器发来的 SYN+ACK 后，回复服务器端一个 ACK 确认，发送完毕后，双方进入 ESTABLISHED 状态。

三次握手成功后，开始传输数据。

7.简述 VPN，常见有哪几种？

VPN 是指在公共的网络上建立专用网络的技术，但是两个节点间并没有物理上的专用的端到端链路，而是通过广域网或者运营商提供的网络平台之上的逻辑网络，用户数据在逻辑链路中传输，它可以有效的节省一般需要达到 DDN 专线所能达到的同样的目的，而且 VPN 采用身份验证和加密技术，充分保证了安全性。

常见的 VPN 有：ipsec vpn、PPTP vpn、L2TP vpn、SSL vpn