## 为了将当前目录下的归档文件myftp. tgz解压缩到/tmp目录下，用户可以使用命令。

tar xvzf myftp. tgz –C /tmp 考察-C参数的作用，指定目录解压。

## Linux下的进程有哪三种状态？

Linux的进程三种状态分别是：

运行状态：这个不用解释了吧，就是正在运行

就绪状态：这个状态等待CPU 时间片的状态，一切准备就绪，随时可以执行，等 CPU 切换到该进程，该进程就会由就绪状态变为运行状态

等待状态：其实就是未就绪状态，还有做一些准备工作或者等待资源

## 在unix系统下执行chmod(“/usr/test/sample”,0753)之后该文件sample的访问权限为

拥有者可读写执行，同组用户可读可执行，其他用户可写可执行

## 在RHEL5系统中使用vi编辑文件report.txt时，要自下而上查找字符串“2006”，应该在命令模式下使用

/从上到下查找

?从下到上查找

## 在退出unix系统账户之后还需要继续运行某个进程，那么可用

nohup命令：如果你正在运行一个进程，而且你觉得在退出帐户时该进程还不会结束，那么可以使用nohup命令。该命令可以在你退出帐户/关闭终端之后继续运行相应的进程。nohup就是不挂起的意思

## 终止一个前台进程可能用到的命令和操作是？

前台是^c，后台用kill

## Linux 文件权限一共 10 位长度，从前数第 5-7 位表示的内容是

第一位表示是文件或者目录，2~4是用户权限， 5~7是组权限 ，8~10是其他用户权限

## Linux下通过哪个命令怎么查看中断？

/proc/interrupts 显示使用的中断

/proc/ioports 当前使用的I/O端口

/proc/kcore 系统物理内存映像。与物理内存大小完全一样，但不实际占用这么多的内存。

/proc/kmsg 内核输出的消息，也被送到syslog

## 若基于Linux操作系统所开发的ARM应用程序源文件名为test.c，那么要生成该程序代码的调试信息，编译时使用的GCC命令正确的是

arm-linux-gcc -g -o test test.c

-o参数用于指定输出的文本，如果不指定输出文件则默认输出.out文件

-c参数将对源程序test.c进行预处理、编译、汇编操作，生成test.0文件

-S参数将对源程序test.c进行预处理、编译，生成test.s文件

-g选项，加入GDB能够使用的调试信息,使用GDB调试时比较方便

## 在OSI模型中，HTTP协议工作在第()层，交换机工作在第()层。

1 物理层： RJ45 、 CLOCK 、 IEEE802.3 （中继器，集线器，网关） -

2 数据链路： PPP 、 FR 、 HDLC 、 VLAN 、 MAC （网桥，交换机） -

3 网络层： IP 、 ICMP 、 ARP 、 RARP 、 OSPF 、 IPX 、 RIP 、 IGRP 、 （路由器） -

4 传输层： TCP 、 UDP 、 SPX -

5 会话层： NFS 、 SQL 、 NETBIOS 、 RPC -

6 表示层： JPEG 、 MPEG 、 ASII -

7 应用层： FTP 、 DNS 、 Telnet 、 SMTP 、 HTTP 、 WWW 、 NFS

## 内存空间被分成不同区域,其中用函数

void\* malloc(unsigned int size);

申请的内存在()

C 编程语言中的 malloc 、 calloc 函数和 C++ 的 new 运算符都是在动态存储区（ heap ）上申请内存空间

## 在重新启动Linux系统的同时把内存中的信息写入硬盘，应使用（#shutdown -r now）命令实现

在linux命令中reboot是重新启动，shutdown -r now是立即停止然后重新启动，都说他们两个是一样的，其实是有一定的区别的。

shutdown命令可以安全地关闭或重启Linux系统，它在系统关闭之前给系统上的所有登录用户提示一条警告信息。该命令还允许用户指定一个时间参数，可以是一个精确的时间，也可以是从现在开始的一个时间段。

精确时间的格式是hh:mm，表示小时和分钟，时间段由+ 和分钟数表示。系统执行该命令后会自动进行数据同步的工作。

该命令的一般格式: shutdown [选项] [时间] [警告信息]

命令中各选项的含义为:

- k 并不真正关机而只是发出警告信息给所有用户

- r 关机后立即重新启动

- h 关机后不重新启动

- f 快速关机重启动时跳过fsck

- n 快速关机不经过init 程序

- c 取消一个已经运行的shutdown

需要特别说明的是该命令只能由超级用户使用。

例1，系统在十分钟后关机并且马上重新启动: # shutdown –r +10

例2，系统马上关机并且不重新启动:# shutdown –h now

halt是最简单的关机命令，其实际上是调用shutdown -h命令。halt执行时，杀死应用进程，文件系统写操作完成后就会停止内核。

halt命令的部分参数如下：

[-f] 没有调用shutdown而强制关机或重启

[-i] 关机或重新启动之前，关掉所有的网络接口

[-p] 关机时调用poweroff，此选项为缺省选项

reboot的工作过程与halt类似，其作用是重新启动，而halt是关机。其参数也与halt类似。reboot命令重启动系统时是删除所有的进程，而不是平稳地终止它们。因此，使用reboot命令可以快速地关闭系统，但如果还有其它用户在该系统上工作时，就会引起数据的丢失。所以使用reboot命令的场合主要是在单用户模式。

init是所有进程的祖先，其进程号始终为1。init用于切换系统的运行级别，切换的工作是立即完成的。init 0命令用于立即将系统运行级别切换为0，即关机；init 6命令用于将系统运行级别切换为6，即重新启动

## 在Linux 系统中，在运行一个程序时，程序中未初始化的全局变量会被加载到以下哪个内存段中? BSS

BSS（Block Started by Symbol）通常是指用来存放程序中未初始化的全局变量和静态变量的一块内存区域。特点是:可读写的，在程序执行之前BSS段会自动清0。所以，未初始的全局变量在程序执行之前已经成0了。

数据段：数据段（data segment）通常是指用来存放程序中已初始化的全局变量的一块内存区域。数据段属于静态内存分配。

代码段：代码段（code segment/text segment）通常是指用来存放程序执行代码的一块内存区域。这部分区域的大小在程序运行前就已经确定，并且内存区域通常属于只读, 某些架构也允许代码段为可写，即允许修改程序。在代码段中，也有可能包含一些只读的常数变量，例如字符串常量等。

堆（heap）：堆是用于存放进程运行中被动态分配的内存段，它的大小并不固定，可动态扩张或缩减。当进程调用malloc等函数分配内存时，新分配的内存就被动态添加到堆上（堆被扩张）；当利用free等函数释放内存时，被释放的内存从堆中被剔除（堆被缩减）

栈(stack)：栈又称堆栈， 是用户存放程序临时创建的局部变量，也就是说我们函数括弧“{}”中定义的变量（但不包括static声明的变量，static意味着在数据段中存放变量）。除此以外，在函数被调用时，其参数也会被压入发起调用的进程栈中，并且待到调用结束后，函数的返回值也会被存放回栈中。由于栈的先进后出特点，所以栈特别方便用来保存/恢复调用现场。从这个意义上讲，我们可以把堆栈看成一个寄存、交换临时数据的内存区

## 下列提法中，不属于ifconfig命令作用范围的是

ifconfig

1.作用：ifconfig用于查看和更改网络接口的地址和参数，包括IP地址、网络掩码、广播地址，使用权限是超级用户。

2.格式 ifconfig -interface [options] address

3.主要参数如下：

-interface：指定的网络接口名，如eth0和eth1。

up：激活指定的网络接口卡。

down：关闭指定的网络接口。

broadcast address：设置接口的广播地址。

pointopoint：启用点对点方式。

address：设置指定接口设备的IP地址。

netmask address：设置接口的子网掩码。

## Linux下多少个"-"将被打印：

int main(void){

int i;

for(i=0;i<4;i++){

fork();

printf("-\n");

}

return 0;

}

i=0时，主进程和其创建的子进程分别打印'-', 打印2个

i=1时，之前两个进程打印'-', 每个进程又创建新的子进程， 共打印4个'-'

i=2时，之前的四个进程分别打印'-', 并创建新的子进程, 故共打印8个'-'

i=3时，之前的8个进程分别打印'-', 并创建新的子进程,故共打印16个'-'

综上所述, 共打印2+4+8+16=30个

## 下列有关Nginx配置文件nginx.conf的叙述正确的是

A. 进程数设置为CPU总核心数最佳。 (正确)

B. 配置虚拟主机，多个域名时，用 空格 分隔。 (错误)

C. 设定http服务器中 sendfile on; #开启高效文件传输模式

普通应用设置为on；

用于进行下载等应用磁盘IO重负载应用，设置为off，以平衡磁盘与网络I/O处理速度，降低系统的负载。(错误)

D. 工作模式与连接数上限 ：（最大连接数 = 连接数 \* 进程数）(正确)

## 下面哪些属于，Fork后子进程保留了父进程的什么？

使用fork函数得到的子进程从父进程的继承了整个进程的地址空间，包括：进程上下文、进程堆栈、内存信息、打开的文件描述符、信号控制设置、进程优先级、进程组号、当前工作目录、根目录、资源限制、控制终端等。

子进程与父进程的区别在于：

1、父进程设置的锁，子进程不继承（因为如果是排它锁，被继承的话，矛盾了）

2、各自的进程ID和父进程ID不同

3、子进程的未决告警被清除；

4、子进程的未决信号集设置为空集。