

Linux 简答题总结

一、概念性知识：

1、Linux 的优点：

- 1) 稳定的系统：Linux 本来就是基于 Unix 概念而发展出来的操作系统，因此，Linux 具有与 Unix 系统相似的程序接口跟操作方式，当然也继承了 Unix 稳定并且有效率的特点。
- 2) 免费或少许费用：由于 Linux 是基于 GPL 的基础下的产物，因此任何人皆可以自由取得 Linux，至于一些『安装套件』的发行者，他们发行的安装光盘也仅需要些许费用即可获得！
- 3) 安全性、漏洞的快速修补：
- 4) 多任务、多使用者：与 Windows 系统不同的，Linux 主机上可以同时允许多人上线来工作，并且资源的分配较为公平，比起 Windows 的单人假多任务系统要稳定的多
- 5) 使用者与群组的规划：在 Linux 的机器中，档案的属性可以分为『可读、可写、可执行』等参数来定义一个档案的适用性。
- 6) 相对比较不耗资源的系统：
- 7) 适合需要小核心程序的嵌入式系统：由于 Linux 只要几百 K 不到的程序代码就可以完成一个完整的操作系统，因此相

当适合于目前家电或者是小电子用品的操作系统—嵌入式系统。

2、什么是 GNU 的 GPL ?

- 1) 任何个人或公司均可释出自由软件 (free software);
- 2) 任何释出自由软件的个人或公司, 均可由自己的服务来收取适当的费用;
- 3) 该软件的原始码 (Source Code) 需要随软件附上, 并且是可公开发表的;
- 4) 任何人均可透过任何正常管道取得此一自由软件, 且均可取得此一授权模式。

3、何谓多人 (Multi-user) 多任务 (Multitask)?

Multiuser 指的是 Linux 允许多人同时连上主机之外, 每个使用者皆有其各人的使用环境, 并且可以同时使用系统的资源!

Multitask 指的是多任务环境, 在 Linux 系统下, CPU 与其它例如网络资源可以同时进行多项工作, Linux 最大的特色之一即在于其多任务时, 资源分配较为平均!

4、简单说明 GNU General Public License (GPL) 与 Open Source 的精神:

- 1) GPL 的授权之软件, 乃为自由软件 (Free software), 任何人皆可拥有他;

- 2) 开发 GPL 的团体(或商业企业)可以经由该软件的服务来取得服务的费用;
- 3) 经过 GPL 授权的软件, 其属于 Open source 的情况, 所以应该公布其原始码;
- 4) 任何人皆可修改经由 GPL 授权过的软件, 使符合自己的需求;
- 5) 经过修改过后 Open source 应该回馈给 Linux 社群。

5、什么是 POSIX ?为何说 Linux 使用 POSIX 对于发展有很好的影响?

POSIX 是一种标准规范, 主要针对在 Unix 操作系统上面跑的程序来进行规范。若您的操作系统符合 POSIX, 则符合 POSIX 的程序就可以在您的操作系统上面运作。Linux 由于支持 POSIX, 因此很多 Unix 上的程序可以直接在 Linux 上运作, 因此程序的移植相当简易! 也让大家容易转换平台, 提升 Linux 的使用率。

6、Linux 的应用

- 1) 网络服务器:
- 2) 工作站计算机:
- 3) 桌上型计算机:
- 4) 嵌入式系统:

7、文件系统

文件系统是[操作系统](#)用于明确[磁盘](#)或分区上的文件的方法和[数据结构](#)；即在[磁盘](#)上组织文件的方法。也指用于存储文件的[磁盘](#)或分区，或文件系统种类。

8、GRUB:

- 1) GRUB 是一种引导装入器（类似在嵌入式中的 bootloader），它负责装入内核并引导 Linux 系统，位于硬盘的起始部分。
- 2) 因此 GRUB 在开机启动的流程中扮演重要角色。

9、BIOS: 是写入主板的一个软件，会识别第一个可开机设备。会根据 CMOS 的设置取得第一个启动设备。并在到该硬盘里读取第一个扇区的 MBR 位置。

10、MBR(主引导分区): 内含引导加载程序（gurb 或者嵌入式中的 Bootloader）

- 1) 引导加载程序：作用是读取内核文件
- 2) 内核文件：开始操作系统

11、Linux 文件类型

- Linux 中的文件类型与 Windows 有显著的区别，其中最显著的区别在于 Linux 对目录和设备都当作文件来进行处理。
- Linux 中主要的文件类型分为 4 种：普通文件、目录文件、链接文件和设备文件。

(1) 普通文件。

普通文件同 Windows 中的文件一样，是用户日常使用最多的文件。它包括文本文件、shell 脚本、二进制的可执行程序和各种类型的数据。

(2) 目录文件。

在 Linux 中，目录也是文件，它们包含文件名和子目录名以及指向那些文件和子目录的指针。目录文件是 Linux 中存储文件名的惟一地方，当把文件和目录相对应起来时，也就是用指针将其链接起来之后，就构成了目录文件。

(3) 链接文件。

链接文件有些类似于 Windows 中的“快捷方式”，但是它的功能更为强大。它可以实现对不同的目录、文件系统甚至是不同的机器上的文件直接访问，并且不必重新占用磁盘空间。

(4) 设备文件。

Linux 把设备都当作文件一样来进行操作，这样就大大方便了用户的使用。在 Linux 下与设备相关的文件一般都在/dev 目录下，它包括两种，一种是块设备文件，另一种是字符设备文件。

二、Linux 常用命令

1、目录命令

- (1) ls 命令：显示目录下文件
- (2) cd 命令：切换目录
- (3) pwd 命令：显示当前目录的命令

示例：ls -a 列出当前目录下的所有文件，包括以.头的隐含文件

ls -l 列出当前目录下文件的详细信息

cd /home 进入 '/home' 目录'

2、常用命令：

- 1) 询问当前日期和时间的 date 命令 例如\$ date 。
- 2) 询问当前用户的 who 命令
- 3) who 命令可列出当前每一个处在系统中的用户的登录名、终端名和登录进入时间
- 4) 显示当前目录路径名的 pwd 命令
- 5) 切换用户命令 su
- 6) 显示日历 cal
- 7) 简单计算器 bc
- 8) 关机 shutdown、halt、 poweroff
- 9) 重启 Reboot

- 10) 账户管理: useradd 添加用户 passwd 修改密码 userdel 删除用户

3、如何改变文件权限

chgrp : 改变档案所属群组

chown : 改变档案所属人

chmod : 改变档案的属性、 SUID 等等的特性

- **chmod 755 mydir (myfile)** 修改目录 mydir(文件 myfile) 的权限为 755, 即 **rwX r-X r-X**, 即目录 (文件) 所有者的权限是可读可写可执行, 同组用户和其他用户都为可读可执行。

4、目录的创建和删除

- (1) **mkdir** 命令 : 功能是建立目录。其语法格式为

```
mkdir [-ep][-m mode] dirname...
```

mkdir dir1 创建一个叫做 'dir1' 的目录'

mkdir -p /tmp/dir1/dir2 创建一个目录树

- (2) **rmdir** 命令 : 功能是删除目录。其语法格式为

```
rmdir [-p][-s] dirname...
```

rmdir dir1 删除一个叫做 'dir1' 的目录'

5、其他常用指令

cp 拷贝档案或目录

rm 删除档案或目录

mv 移动档案或目录

示例: `rm -f file1` 删除一个叫做 'file1' 的文件'

`rm -rf dir1` 删除一个叫做 'dir1' 的目录并同时删除其内容

`mv dir1 new_dir` 重命名/移动 一个目录

`cp file1 file2` 复制文件 file1 到 file2

`cp dir/* .` 复制一个目录下的所有文件到当前工作目录

`cp -a /tmp/dir1 .` 复制一个目录到当前工作目录

`cp -a dir1 dir2` 复制一个目录

6、观看档案内容: `cat`, `more`, `less`,

`cat` 由第一行开始显示档案内容

`more` 一页一页的显示档案内容

`less` 可以前后翻页

修改档案时间与建置新档: `touch`

示例: `touch file1` 在当前目录下新建文件 file1

档案类型: `file`

示例: `file ~/.bashrc` 查看 home 目录下 .bashrc 文件的类型

7、档案的搜寻: `which`、`whereis`、`locate` 和 `find`

`find / -name file1` 从 '/' 开始进入根文件系统搜索文件和目录

`find /home/user1 -name *.bin` 在目录 '/home/user1' 中搜索带有'.bin' 结尾的文件

`locate *.ps` 寻找以 '.ps' 结尾的文件 - 先运行 'updatedb' 命令

`whereis halt` 显示一个二进制文件、源码或 man 的位置

`which halt` 显示一个二进制文件或可执行文件的完整路径

■ 范例：找出档名为 `passwd` 这个档案

■ # `find / -name passwd`

■ 范例：找出系统中，大于 1MB 的档案

■ # `find / -size +1000k`

8、显示帮助

■ `man` [选项] 命令名称

9、文件压缩和解压缩

命令：`gzip` 压缩文件 `tar` 文件打包

`gzip file1` 压缩一个叫做 'file1' 的文件

`tar -cvf archive.tar file1` 创建一个非压缩的 tarball

`tar -cvf archive.tar file1 file2 dir1` 创建一个包含了 'file1', 'file2' 以及 'dir1' 的档案文件

`tar -tf archive.tar` 显示一个包中的内容

`tar -xvf archive.tar` 释放一个包

`tar -xvf archive.tar -C /tmp` 将压缩包释放到 `/tmp` 目录下

`tar -cvfz archive.tar.gz dir1` 创建一个 `gzip` 格式的压缩包

`tar -xvfz archive.tar.gz` 解压一个 `gzip` 格式的压缩包

10、硬盘的分区、格式化、检验与挂载

`df -h` 显示已经挂载的分区列表

磁盘分割: `fdisk`

范例一: 将刚刚建立的 `/dev/hdb5` 挂载到 `/mnt/hdb5` 上面!

```
[root@linux ~]# mkdir /mnt/hdb5
```

```
[root@linux ~]# mount -t ext3 /dev/hdb5 /mnt/hdb5
```

```
[root@linux ~]# df
```

范例二: 挂载光盘!

```
[root@linux ~]# mount -t iso9660 /dev/cdrom /mnt/cdrom
```

```
[root@linux ~]# mount /dev/cdrom
```

范例三: 挂载 USB

```
[root@linux ~]# mkdir /mnt/usb
```

```
[root@linux ~]# mount -t vfat /dev/sda1 /mnt/usb
```

11、文件属性：

	连接数	文件所属用户组	文件最新修改时间	
	↑↑	↑↑	↑↑	
-rwxrwxrwx	1	root	293	Oct 19 21:24 test
文件属性	文件拥有者	文件大小		文件名

三、VIM 使用

- 默认为普通模式
- 输入 i/a 进入编辑模式
- Escape 返回普通模式
- 输入:+命令，执行命令
 - :q! 不保存退出
 - :wq! 保存退出

1、光标移动

单词词尾或后一个单词词尾：e

后一个单词词首：w

单词词首或前一个单词词首：b

跳转到特定行：:行号（:5：跳转到第5行）

跳到文件头: gg

跳到文件尾: G

2、搜索

向后模式搜索: / + 搜索模式

向前模式搜索: ?+ 搜索模式

注意: 结合 n 和 N 向前向后查找

3、追加或插入

光标右侧追加: a

光标行尾追加: A

光标左侧追加: i

光标行首追加: I 『i 的大写』

光标下方插入行: o 【字母 o】

光标上方插入行: O 【字母 o 的大写】

4、删除

删除光标: x

删除光标左侧: X

删除光标所在的单词: dw

删除当前行: dd

删除光标至行尾: `d$` 或 `D`

删除当前行及以下 `n-1` 行: `ndd` 『`n` 是数字, 是要删除的行数』

删除 `n` 到 `m` 行: `:n,m d` 『删除 1 到 3 行 `:1,3 d`』 当 `m` 用 `$` 时, 表示到文档尾

5、复制粘贴

复制光标所在的单词: `yw`

复制光标至行尾: `y$` 或 `Y`

复制当前行: `yy`

复制行 `n` 次: `yyn`

复制当前行及以下 `n-1` 行: `nyy` 『`n` 是数字, 是要复制的行数』

复制 `n` 到 `m` 行, 粘贴到 `k+1` 行: `:n,m co k` 『复制 1 到 3 行, 粘贴到第 5 行 `:1,3 co 4`』

粘贴到光标下: `p`

粘贴到光标上: `P`

6、撤销回复

撤销: `u`

恢复: `ctrl + r`

7、替换

替换光标所在的字符: r

替换一个单词: cw

替换光标所在的前一个字符: cb

替换光标至行尾的所有字符: c\$

替换当前行: cc

8、其他

查看当前设置: :set all

显示行号: :set nu

隐藏行号: :set nonu

不保存退出: :q

不保存强制退出: :q!

保存编辑: :w

保存并退出: :wq

执行 shell 命令: :! shell 例如: 查看当前时间: :! date

读取 shell 执行的结果, 并把结果放在光标的后面: :r!shell 例如:

列出目录中的文件: :r!ls

简单加密文件: :X

2、gcc 编译过程

- 预处理-cpp
- 编译-cc1
- 汇编-as
- 链接-ld

例如: `$gcc -E hello.c -o hello.i`

说明: 将 `hello.c` 做预处理, 输出 `hello.i`, 它的本质是一个编译单元。

例如: `$gcc -S hello.i -o hello.s`

说明: 使用 GCC 的 `-S` 选项, 该选项将编译单元翻译成汇编语言, 它只进行编译而不进行汇编, 生成的 `hello.s` 是汇编代码。

例: `$gcc -c hello.s -o hello.o`

说明: 使用 GCC 的 `-c` 选项, 将汇编代码翻译成二进制机器码, 只翻译, 不进行连接。

例如: `$gcc hello.o -o hello.exe`

说明: 进行连接, 生成可执行文件 `hello.exe`

- 最简单用法: `gcc hello.c -o hello`