

1. 计算机基础

1.1 进程切换

当一个进程在执行时,CPU的所有寄存器中的值、进程的状态以及堆栈中的内容被称为该进程的上下文。当内核需要切换到另一个进程时,它需要保存当前进程的所有状态,即保存当前进程的上下文,以便在再次执行该进程时,能够必得到切换时的状态执行下去。在Linux中,当前进程上下文均保存在进程的任务数据结构中。在发生中断时,内核就在被中断进程的上下文中,在内核态下执行中断服务例程。但同时会保留所有需要用到的资源,以便中继服务结束时能恢复被中断进程的执行。

1.2 特性

- D-Bus 是 desktop-bus 的简称,是一个低延迟、低开销、高可用性的进程间通信机制。它越来越多地用于应用程序之间通信,也用于应用程序和操作系统内核之间的通信。很多现代的服务进程都使用D-Bus 取代套接字作为进程间通信机制,对外提供服务。

2. 常用目录

- **usr**: **unix/user system resources** - 早起的Unix版本中, /usr目录是作为用户的家目录而存在的, 主要存放的是一些非系统启动必要的程序和数据
 - /usr目录存放 **共享的**, **只读** 的数据。这意味着/usr应该是在兼容FHS标准的主机间共享的, 并且其中的数据是不应该被修改的。那些与主机属性相关或者经常变动的数据不应该存放在次目录
- **/etc/fstab** - 挂载点配置文件
- **etc**: etc不是什么缩写, 是and so on的意思 来源于 法语的 **et cetera** 翻译成中文就是 等等 的意思. 至于为什么在/etc下面存放配置文件, 按照原始的UNIX的说法(Linux文件结构参考UNIX的教学实现MINIX) 这下面放的都是一堆零零碎碎的东西, 就叫etc, 这其实是个历史遗留。

3. 环境变量

3.1 临时变量

- **export** key=value

3.2 永久变量

- 修改profile文件
 - 修改/etc/profile (对所有用户都是有效的)
- 修改.bashrc文件
 - 修改~/.bashrc 文件。 (每个用户目录下都有, ls -all, 单独用户有效)
 - **source** 命令使变量立即生效

3.3 详解

更多解释请[参考](#)

4. 命令

4.1 文件

4.1.1 find

- `find /dir -name filename`
- `-perm` 按照文件权限来查找文件（permission）
 - `find . -perm 755 -print` 在当前目录下查找文件权限位为755的文件
- `-prune` 使用这一选项可以使find命令不在当前指定的目录中查找，如果同时使用 `-depth` 选项，那么 `-prune` 将被find命令忽略
 - `find /apps -path "/apps/bin" -prune -o -print` 在/apps目录下查找文件，但不希望在/apps/bin目录下查找
- `-depth`：在查找文件时，首先查找当前目录中的文件，然后再在其子目录中查找
 - `find / -name "filename" -depth -print` 它将首先匹配所有的文件然后再进入子目录中查找
- `-user` 按照文件属主来查找文件，类似的属性筛选：`-group`
 - `find . -user userName -print` 在\$HOME目录中查找文件属主为userName的文件
- `-nogroup` 查找无有效所属组的文件，即该文件所属的组在/etc/groups中不存在
- `-nouser` 查找无有效属主的文件，即该文件的属主在/etc/passwd中不存在
- `-mtime -n +n` 按照文件的更改时间来查找文件
 - `-n` 表示文件更改时间距现在n天以内
 - `+n` 表示文件更改时间距现在n天以前
- `-newer file1 ! file2` 查找更改时间比文件file1新但比文件file2旧的文件
- `-type` 查找某一类型的文件
 - `b` - 块设备文件
 - `d` - 目录，eg: `find /etc -type d -print` 在/etc目录下查找所有的目录
 - `c` - 字符设备文件
 - `p` - 管道文件
 - `l` - 符号链接文件
 - `f` - 普通文件
- `-size n[c]` 查找文件长度为n块的文件，带有c时表示文件长度以字节计
 - `find . -size +1000000c -print` 在当前目录下查找文件长度大于1 M（1M应该是1024 * 1024，本人没有考证此命令中的1000000的正确性）字节的文件
- `-mount` 在查找文件时不跨越文件系统mount点
 - `find . -name "*.XC" -mount -print` 从当前目录开始查找位于本文件系统中文件名以XC结尾的文件（不进入其他文件系统）
- `-follow` 如果find命令遇到符号链接文件，就跟踪至链接所指向的文件
- `-exec`，find命令对匹配的文件执行该参数所给出的shell命令。相应命令的形式为`command'`，注意{}和;之间的空格

```
$ find ./ -size 0 -exec rm {} \; 删除文件大小为零的文件
$ rm -i `find ./ -size 0`
$ find ./ -size 0 | xargs rm -f &
```

为了用ls -l命令列出所匹配到的文件，可以把ls -l命令放在find命令的-exec选项中：

```
$ find . -type f -exec ls -l {} \;
在/logs目录中查找更改时间在5日以前的文件并删除它们：
find /logs -type f -mtime +5 -exec rm {} \;
```
- `-ok`，和-exec的作用相同，只不过以一种更为安全的模式来执行该参数所给出的shell命令，在执行每一个命令之前，都会给出提示，让用户来确定是否执行

```
find . -name "*.conf" -mtime +5 -ok rm { } \;
```

在当前目录中查找所有文件名以.LOG结尾、更改时间在5日以上的文件，并删除它们，只不过在删除之前先给出提示

4.1.2 whereis

和find相比，whereis查找的速度非常快，这是因为linux系统会将系统内的所有文件都记录在一个数据库文件中，当使用whereis（locate也是）时，会从数据库中查找数据，而不是像find命令那样，通过遍历硬盘来查找，效率自然会很高：

但是该数据库文件并不是实时更新，默认情况下时一星期更新一次

- 命令格式：whereis [-bmsu][BMS 目录名 -f] 文件名
- 参数
 - -b 定位可执行文件
 - -m 定位帮助文件
 - -s 定位源代码文件
 - -u 搜索默认路径下除可执行文件、源代码文件、帮助文件以外的其它文件
 - -B 指定搜索可执行文件的路径
 - -M 指定搜索帮助文件的路径
 - -S 指定搜索源代码文件的路径

4.1.3 文件编辑

- vim 设置行号：:set nu!

4.2 用户

4.2.1 adduser

会自动为用户指定主目录、系统shell版本，会在创建时输入用户密码

- adduser hongxing
 - 添加用户hongxing
 - 可以再通过 passwd hongxing 为用户设置密码

4.2.2 useradd

需要使用参数选项指定 adduser 默认的基本设置，如果不使用任何参数，则创建的用户无密码、无主目录、没有指定shell版本

4.2.3 userdel

5. 趣味linux

5.1 有意思的命名

- ~/.bashrc 中的 rc 意思 - run command

5.2 linux/unix/BSD关系

哎,老有人问这个,这么说吧,最早的unix是开放的,很多组织对unix都有修改,期中比较有名的就属白克利大学的修改版本,叫做bsd,是unix的分支,由于bsd的协议允许你直接使用/修改它的代码,并且可以作为商业用途,所以很多公司的unix都是从bsd衍生过来的,比如hp-unix,ibm的aix等。

linux呢,是一个完全没有参照unix代码的一个仿照unix架构,试图与unix相兼容的开源系统.换句话说,bsd是unix的分支,而linux是类unix系统或者说是仿unix系统。