档案管理

创建文件

touch 目录1/文件1 ：在目录1下创建文件1

浏览文件

cat 文件名 ： 查看文件

cat -n 文件名 : 查看文件，同时文件的内容显示行号

cat -b 文件名 ：查看文件，同时文件的内容显示行号，但是空格行没有行号

less 文件 ： 查看文件内容

回车 ： 往下换行 空格 ：往下翻页 paup ： 往上翻页

/关键字 ： 在文件中查找关键字，按n切换关键字

more 文件 ： 查看文件内容

b ： 往上翻页

head -n ： 只显示文件的前n行

tail -n ： 只显示文件的后n行

tail -f ： 动态显示日志更新内容

创建目录

mkdir 目录1 ：创建目录1

mkdir -p 目录1/目录2 ：同时创建目录1/目录2

mkdir -m 777 目录1 ：创建目录1，并设置目录1的权限是777

命令实例：

同时创建多个目录？ mkdir 目录1 目录2 目录3

mkdir -m 777 -p /tmp/1/2/3/4 :只有目录4的权限是777,123都是默认权限

ls ：列出当前目录所有非隐藏文件

ls -a ：列出当前目录所有文件，包括隐藏文件，隐藏文件是文件名以 . 开头的文件

ls -l ：列出档案同时列出档案的属性，如，iNode，大小，权限等信息

ls -h ：档案大小单位用K，M，G，T表示

ls -ld ：【显示目录属性，如 ls/tmp,显示目录tmp的属性，而不是tmp里面各个档案的属性】

ls -t ：按档案最后修改时间排序

ls -S ：按档案大小排序

ls -sh ：列出档案时同时列出档案的大小

ls -i ：【输出档案的i结点】

绿色代表可执行文件

红色代表tar包文件

蓝色代表目录文件

红代表图象文件

青色代表链接文件

黄色代表设备文件

切换目录

cd 目录 ：切换到指定的目录

cd .. ：切换到当前目录的上一级目录

cd ~ ：切换到当前用户家目录

cd / ：切换到根目录

cd - ：切换到 进入当前目录前的目录

pwd ：显示当前目录

复制

cp 文件 目的目录1 ：把文件复制到目录1中

cp 文件 目的目录/新文件名： 把文件复制到目录1中同时重命名成新文件名

cp -r 目录1 目的目录2 ：把目录1复制到 目录2中

cp -r 目录1 目录2 文件1 文件2 目录3 目录4 ：把目录1 目录2 文件1 文件2 目录3 同时复制到目录4中

cp -p 或 --preserve ： 保留源文件或目录的属性，包括所有者、所属组、权限与【时间】

cp -f ：强制复制，直接覆盖原有文件或者目录，不必询问

剪切和重命名

mv 目录1/文件 ：目录2 把文件剪切到目录2

mv 目录1/文件1 目录2/文件2 ： 把文件1剪切到目录2，并重命名为文件2

mv -f ： 强制覆盖，不必询问

【在同一目录中执行mv操作，就是重命名】

mv 目录1 目录2

mv 文件1 文件2

重名为目录1 和 文件1

删除

rm 文件 ：删除文件

rm -r 目录 ：删除目录

rm -f ：强制删除，不必询问

软/硬链接

ln 文件1 目录2 ： 生成文件1/目录1的硬链接放到目录2下

ln –s 文件1/目录1 目录2 ：生成文件1/目录1的软链接放到目录2下

档案的硬链接与档案本身完全相同，相当于 cp –p + 同步更新 ，二者属性也相同，二者之一发生变化，另外一个保持同步更新。

硬链接不能跨分区，目录不能生成硬链接，硬链接和源档案的inode相同

档案的软链接相当于档案的快捷方式

chmod chown chgrp

修改文件的权限：chmod 567 file||目录 r4 w2 x1

chmod 567 目录1 目录1 的权限是567 但是目录1下的权限不应是567

chmod –p 567 目录1 目录1 和 目录1下的所有目录权限全是567

r w x 对文件和目录的意义：



对于目录 有r权限就有x权限

删除一个文件，不由文件权限决定，由对该文件所在目录是否有w权限决定

怎样查看一个新文件和新目录的默认权限：umask –S

改变文件和目录的所有者 chown 档案 所有者

改变文件和目录的所属组 chgrp 档案 所属组

只有root才能改变档案的所属组和所有者

文件搜索命令

尽量不用搜索命令，因为搜索命名什么占用cpu

零碎知识点

cat 1.txt 2.txt 3.txt > 4.txt ：把1.txt 2.txt 3.txt三个文本的内容合并成一个文本4，

文本4如果不是一个新文件，会把文本4原来的内容清空后填入 1,2,3的内容

cat 1.txt 2.txt 3.txt >> 0.txt 把一个或多个已存在的文件内容，追加到一个已存在的文件中

ctrl + l ：清屏

软件包管理

1，源码包，脚本安装包，rmp包（二进制包）区分

源码包是高级语言安装包，能看到源代码，安装自主选择性强，安装前需要编译，安装很慢，安装步骤繁琐，安装过程一旦出错，难以解决；

脚本安装包是带图形安装界面的源码包

rmp包是源码包经过编译后的二进制包，安装前不需要再行编译，安装很快，但是不能看到源代码，自主选择性也很低，安装包的依赖性也特别多。

linux软件安装包名字的格式：

软件名-版本号-发布次数-适用的linux-硬件平台-扩展名

2，rmp包的依赖性介绍

依赖性：安装A，需要先安装B……

依赖的分类：

树形依赖：先安装终端结点软件，上溯，最后安装根结点软件

环形依赖：A->B->C->A 解决方法：一条命令同时安装ABC

模块依赖：安装A要先安装B（扩展名.so.num），但是B不是一个独立的软件，它含在软件包C中，所以要安装C，怎样确定B在哪个安装包？[www.rmpfind.net](http://www.rmpfind.net)查询。

linux怎样挂载U盘？怎样挂载光盘?

fdisk –l 插入U盘 fdisk –l 确定U盘插上连接稳定，并记下U盘的设备名

cd /mnt mkdir /usb mount –t vfat /dev/sdb /mnt/usb

cd /mnt/usb 查看U盘内容

卸载U盘 删除U盘的挂载目录

rpm包全名：操作的包是没有安装的包时使用包全名.要注意路径

rpm包名：操作的包时已经安装的包时使用包名。是搜索/var/lib/rpm的数据库

rpm安装： rpm –ivm 包全名 i istall v ver

mount检查光盘是否已经挂载 cd /mnt/cdrom/Packages/进入rpm包所在光盘目录 ls 列出所有的rpm包 rpm –ivm 包全名

升级rpm包： rpm –Uvh 包全名 U upgrade升级

卸载 ： rpm –e 包名

gcc

条件编译：

----------------------------------------------

#include<stdio.h>

#define R 68000

int main() {

#if R == 8086

printf("哈哈\n");

#elif R == 68000

printf("嘿嘿\n");

#else

printf("呵呵\n");

#endif

}

#if 0

代码段

#endif

代码段被注释掉，不会被编译，把0改成1，会编译

-------------------------------------------------------------

#ifdef  标识符  
        程序段1  
    #else  
        程序段2  
    #endif

如果标志符被宏定义过，编译程序段1，否则编译程序段2

#include<stdio.h>

#define R

int main() {

#ifdef R

printf("哈哈\n");

#else

printf("嘿嘿\n");

#endif

}

---------------------------------------------------------------------------

#ifndef 标识符  
        程序段1   
    #else   
        程序段2   
    #endif

一般用于防止一个源程序多次包含头文件

----------------------------------------------------------------------------

#include<a.h>

#include"A.h"

<>到系统预设头文件目录寻找文件

“”到源文件的当前目录寻找头文件，找不到就到“”的指定目录下寻找头文件

main.c的头文件在../src目录下，编译main.c时的命令如下，会在src中寻找头文件，否则，编译出错。

gcc main.c -o main -I ../src

----------------------------------------------------------------------------

#define PI 3.1415926 符号常量

const double PI = 3.1415926

符号常量不占空间，常变量占空间

带参数的函数式宏定义

1. 求圆的面积 #define Area(x) (PI\*(x)\*(x)) Area(4)
2. max #define Max(x,y) ((x) > (y) ? (x) : (y)) Max(2,5)

宏定义最好用内联函数代替，内敛函数既消除了函数调用的开销，又能提供宏不能进行类型检查的缺点以及避免宏的各种副作用。带参数的宏务必要为参数加( ).

-------------------------------------------------------------------------------

运算符 # 和 ##

#define HELLO(x) cout <<

gcc编译器把C/C++及其汇编文件编译链接成可执行文件。

预处理：将头文件副本拷贝到源文件(#include)，替换宏（#define），条件编译。.i 经过预处理后的C语言文件

编译：把C/C++编译成汇编二进制文件（目标文件），但不是可执行文件 后缀是.o

链接：把多个目标文件链接成单个可执行文件。

main.c权限不是x所以不是可执行文件。可不可以执行ls -l.

-------------------------------------------------------------------------------

gcc main.c -o main //生成main文件，没有后缀，是可执行文件

./main

-o 确定可执行文件的名称，不-o，默认可执行文件名是a.out

gcc -c main.c //生成后缀是.o的目标文件，不可执行

gcc main.o -o main //生成可执行文件

./main

------------------------------------------------------------------------------

gcc -g main.c -o main //利用gdb调试，生成的可执行文件包含调试信息，比不加-g文件略大

l

gdb main

n 10

r

s main

p ret

gcc main.c -I /home/Ryan/desktop -o main

gcc -c mian.c 只编译不链接

gcc main.o -o main

./main

gcc -E mian.c>main.e//预编译文件，展开头文件和宏

gcc -S main.c //生成main.s 只编译不汇编

gcc -g main.c -o mian\_d //debug版本

gcc 多文件编译

静态编译：把所有引用的头文件全部弄到源文件中

gdb

视频链接 http://video.tudou.com/v/XMjIyNDg0NTQwMA==.html

安装gdb： yum install gdb

查看是否安装了gdb：gdb

退出：q

载入待调试的程序：

gcc -g main.c -o main

gdb启动时载入: gdb main

gdb启动后载入：

gdb

file main

l

l 列出10行；

l 3,5 列出3-5行

l 6 列出第6行的上下5行

r

运行程序

设置断点

b 9 //程序运行到第9行停止

r

回车键，重复执行断点

continue继续接着往下执行

b 6 if i > 9 //程序运行到第9行或i>9时停止

b 函数名，运行到该函数停止

删除断点

d 断点编号

b 1

b 2

b 4

b 6

info breakpoints //查看当前所有断点

d 1-5 //删除1-5号断点

删除指定行的断点

clear 8 //删除第8行的断点

断点失效

disable 1 //使编号为1的断点失效

enable 1 //使编号为1的断点恢复有效

设置观察点

watch 表达式或变量

用来监测某个变量或表达式是否变化，变化了暂停程序。

用法：

设置断点 r

watch 变量

发生变化暂停，打印旧值和新值

continue 下一次发生变化的旧值和新值

rwatch //变量被读时暂停

awatch //变量被读或被写时暂停

info watchpoints //显示所有观察点

显示数据与修改变量值

显示数据用于查看数据在程序运行过程中的值，改变变量值则是在程序调试过程中可以修改变量的值以便发现错误。

用法：

设置断点r,p 指定变量，会打印此时该变量的值。

查看数组a的前6个元素的值

p \*a@6

自动显示变量或表达式的值

display 变量

查看变量的数据类型

what is 变量

修改变量的值 set i = 100

单步执行 s进入函数内部，不断按s，就是单步执行，当调用函数时，会打印形参起始行号等信息。还会给出将要执行的语句和行号。

return结束被调用的函数

执行到指定行 until 行号

跳转到指定行 jump 行号，被跳过的代码不执行

makefile

将所有源文件放到一个目录中，执行make命令，或自动把所有文件生成目标文件，并把所有目标文件生成一个可执行文件。

make 的好处：

简化编译时的指令。

若在编译完成后，修改了某个源码文件，如果使用make命令，会针对被修改的文件重新编译，效率高

最后更新执行文件。

软件安装

源码包 // 后缀是.c，需要编译，安装慢，编译过程如果出错，无法处理，功能自由定制，

二进制包//（后缀.rpm）安装快，依赖性，不能看源码

rpm包命名格式：软件包名，软件版本，软件发布次数，适合的linux平台，硬件平台，后缀名

依赖性：安装A软件，需要事先安装B软件

环形依赖，同时安装所有就行了

模块依赖：A软件依赖B模块，B模块只是一个软件C的模块，无法单独安装，需要事先安装C，但是不知道C是谁。有专门网站可查询。

rpm包管理--yum在线管理，yum可以不用管依赖性，一次性安装。简单便捷。

挂在U盘

mkdir mnt/cdrom

mount /dev/sr0 /mnt/cdrom

cd /mnt/cdrom

安装wps的过程：

（1）百度wps的rpm包，下载到windows桌面

（2）利用Xftp发送到linux虚拟机

（3）切换到wps安装包所在目录

（4）rpm -ivh wps的rpm包全名

（5）出现依赖性，安装被依赖的包

（6）查询是否安装了wps rpm -qa | grep wps

（7）查询软件的安装位置 rpm -ql 包名

（8）升级wps 下载升级包，rpm -Uvh 升级包名

（9）卸载wps rpm -e 包名

临时配置虚拟机的IP地址 ifconfig ens33 IP

长期配置IP地址

cd /etc/sysconfig/network-scripts

vim ens-33

重启网络服务 service network restart