1.操作系统：有效的管理硬件资源，向下封装硬件，向上提供接口。

2.文件系统：windows以盘符管理，linux以目录管理，都是以树状方式管理。Linux中FHS标准规定了Linux根目录各文件夹的名称和作用。

3./bin:放工具

/home:家目录，放用户自己的文件

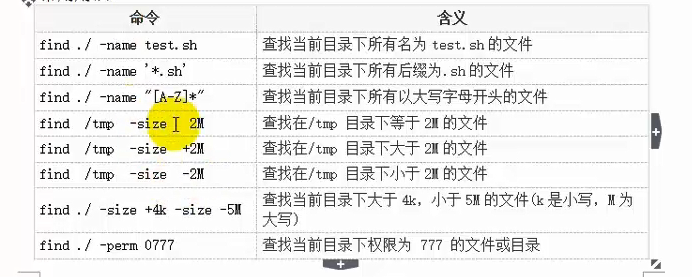
/lib:放库函数

/dev：放设备文件

4.Linux一切皆文件，分为普通文件（文件权限中以-开头），目录文件（以d开头），设备文件（以c(char)开头或以b(block)开头），链接文件（以l开头，类似快捷方式）。

1. 文件权限：三组：文件所有者，用户组，其他用户。（分别有读，写，执行）。
2. 帮助，ls --help或man ls
3. 查看历史命令：history.
4. ls命令：-l 列表详细信息 -a 全部文件（包含隐藏）-h 人性化方式（以K,M,G方式显示文件大小）
5. 输出重定向：ls > test.txt, ls >> test.txt(追加不覆盖)，本来显示到屏幕上的显示到文件里。
6. more:分屏显示
7. 管道:一段塞东西，另一段取东西。| ls | grep XX
8. cd, .为当前目录，~为用户主目录。
9. mkdir -p(递归创建) mkdir ./a/b/c -p
10. rmdir 不能删非空目录
11. rm -r(删目录) rm \* -rf (-f 忽略不存在的文件)
12. ln 链接文件 ln -s 软链接
13. etc 配置文件
14. 硬链接：
15. 给某个文件拷贝一份，这2个文件独立占用内存。
16. 改变其中一个文件，会影响另外一个文件。
17. 删除某个文件，不会影响另一个文件。

软链接：

1. 类似windows的快捷方式，软链接依赖于源文件存在
2. 如果源文件删除，软链接没有意义。
3. 文本搜索：grep 正则表达式：^a（以a开头） ke$(以ke结尾) [0-9]（包含0-9k中的一个数） .（通配符）
4. find 
5. cp -r(拷贝文件夹) -v显示进度
6. mv 可用来改名
7. file 看文件类型
8. 打包命令：tar -cvf aaaa.tar [文件名] c = creater v = 进度 f后名称
9. 解包命令：tar -xvf -C（指定路径） 目录
10. gzip xxx.tar 压缩 -r(需要指定压缩后的名字)
11. gzip -d 解压
12. 压缩打包: tar -czvf xxx.tar.gz 所需文件（加z调用gzip）
13. 解压解包:tar - xzvf xxx.tar.gz -C（指定路径） 目录
14. bzip:另一种压缩方法
15. 一步到位：tar -cvjf xxx.tar.bz2 所需文件
16. tar -xvjf xxx.tar.bz2
17. which 查看命令所在目录
18. 查看目录大小 du ./目录 –h
19. who查看用户
20. exit退出当前用户到上一个用户
21. 创建一个用户（系统默认创建一个用户用户组，和用户名字一样）
22. 查看用户组信息 cat /etc/group
23. 查看用户信息 cat /etc/passwd
24. usermod –g 用户组 用户名 将用户添加到某组
25. 文件夹权限说明：
    1. x: 没有此权限，不允许用户进入
    2. r: 没有此权限，无法查看文件夹的内容
    3. w:没有此权限，无法新建或删除文件。

38．ps -aux:查看进程

39. kill 进程号，杀进程

40. 后台程序：&,jobs,fg

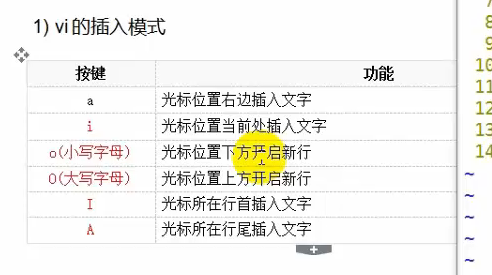
a) 用户可以将一个前台执行的程序调入后台执行，方法为：命令 &

b) 如果程序已经在执行，ctrl + z可以将程序调入后台

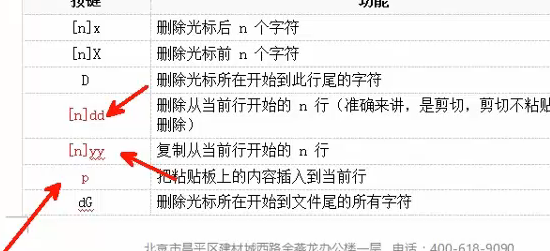
c) jobs查看后台运行程序

d) fg编号（编号为通过jobs查看的编号），将后台程序调到前台

41．vi插入模式

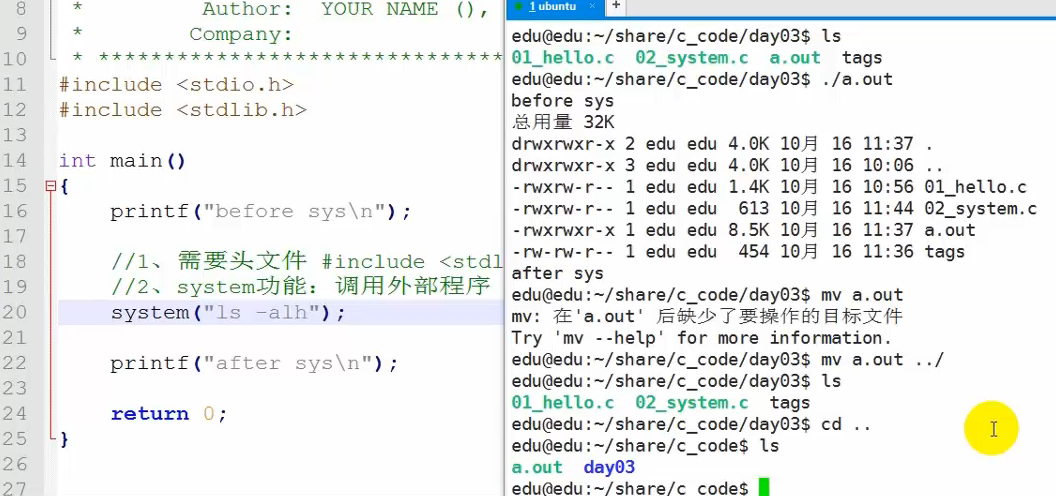


42. vi删除和修改



43. vi的行定位



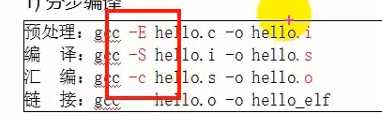
45.system:在已经运行的程序中执行另一个外部程序  
 

46. 字符编码：

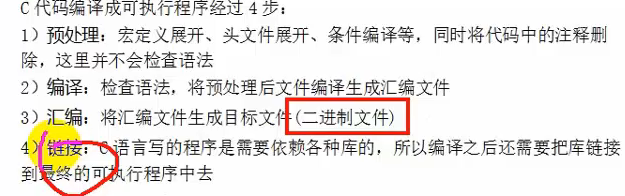
1)windows默认支持的中文编码为gbk,gb2312,ANSI

2)linux默认支持的中文编码为utf-8(Unicode)

47. gcc分步编译的过程



一步到位：gcc hello.c –o hello



48. ldd 查找可执行文件依赖的动态库

49.数据类型的作用：告诉编译器定义这么一个类型的变量需要分配多大的空间。

50.计算机为何用补码存储？

原码两个问题：1.两种0的表示

正数和负数相加，结果不正确

反码：正数的原码和反码一样

负数符号位不变，其他位取反

反码问题：0有两种存储方式

补码：正数的原码、反码、补码都一样，负数为其反码+1

51．char 无符号：0000 0000 ~1111 1111 0~255

char 有符号：-128~127

52. 数组名是常量，常量不能修改，不初始化{}，则内部为随机数。

53. C语言没有字符数组类型，用字符串代替。

54. strlen需要使用返回值，返回值就是字符串的长度，从首元素开始，到结束符为止的长度，结束符不算（遇到’\0’线束）。sizeof()测数据类型的长度，不会因为结束符提前结束。

55. strcpy从首元素开始，遇到\0结束，strncpy指定长度，不会遇到\0结束。

56. strncmp指定比较前n个字符。

57. atoi(字符串转为整数)，atof(字符串转为浮点数)