bd mindmaster

# Week04

每日日之不清，知本者之可证

## 文件的查找

- find
  - find 起始目录 查找条件 操作
  - 查找所有以起始目录为根夫的目录文件
  - 按名字精确查找：find .name [文件名]
  - 按名字模糊查找：
    - 通配符：
      - *匹配：0~多个任意字符
      - ? 匹配：仅能匹配一个字符
    - [ ] 范围匹配：需要快捷ASCII码的范围写
  - find . -gid 按组查找
  - find . -uid 按用户查找
  - find . -empty 查找空文件
  - find . -prem xxx 按权限查找
  - find . -size n[bckw] （n后面的[ ]表示单位）默认单位为block
  - find . -type 按类型查找 d 目录 f 普通文件
  - find . [acm]+mins/time （+/-）（min 分钟，time 天）
    - a：访问
    - c：状态
    - m：内容
    - min后无符号，表示某时刻
    - min- 从现在到min分钟之内
    - min+ min分钟之前a
  - find .name "dir*" -exec 命令\; find的组合操作
    - 将find命令的结果作为参数传递exec的命令中
    - 其中的{}表示find的每一行
  - 将前一个命令的输出作为后面的输入，一次接受一个单词
  - 普通
    - -xargs 多行命令执成一个单词
    - eg.统计当前有多少文件：ls|wc -l
- 磁盘命令：repo repot file system disk space usage
  - df命令：
    - 显示磁盘空间：注意Mounted on 为挂载
    - 在Linux中，外部硬件会直接导入根目录？
    - 当在 Linux 系统中使用外部硬件时，
    - 只有将Linux本身的文件目录与硬件设备的文件目录合二为一，硬件设备才能为我们所用，合二为一的过程称为“挂载”
  - du：estimate file space usage
    - du命令：深度查找
    - - 显示当前目录占多少空间
    - 可以通过加参数 --max-depth=1 调整参数，显示当前深的大小
    - ...
- cat：Concatenate files and print on the standard output
  - cat
    - 打印
    - cat -b ：标记所有非空行
    - cat -n：标记所有行
    - cat -s：将连续空行合并显示
    - cat -E：对每一行的结束显示$符

## 标准输入和输出

- 杂七杂八
  - uniq：删除连续重复信息
  - sort：排序
  - 结合sort &&uniq可以连续去重
  - 文件字数统计：wc
    - wc 无参数时显示所有
    - wc -l：统计行数
    - wc -w：统计单词
    - wc -c：统计字节
  - 编码：
    - 由于Linux下用的UTF-8标准（3字节），会出现windows（2字节）和Linux编码不同的情况
    - 用conv转换
    - iconv -f gb2321 -t utf-8 文件名 > 新文件名
    - 需要重定向得到转换之后的结果
- 重定向
  - man sh：查看相关帮助
  - 重定向符号：箭头的指向就说数据的流向
    - < ：输出重定向（把前面输出的东西输入到后边的文件中，会清除文件原有的内容）
    - > ：输入重定向（输入重定向用于改变命令的输入，后面指定输入内容，后面跟文件名）
    - >> ：追加输出重定向（把前面输出的东西追加到后边的文件尾部，不会清除文件原有的内容）
  - 文件描述符
    - -0：stdin（标准输入）：使用<或<<，数据流从右向左
    - 1：stdout（标准正常输出）：使用>或>>，数据流从左向右
    - -2：stderr（标准错误输出）：使用2>或2>>，数据流从左向右
- grep：指定格式打印，将文件中符合格式的打印出来
  - -F：每个模式作为固定的字符串对待
  - -c：只显示匹配行的数量。（只显示总数）
  - -i：比较式不区分大小写。
  - 选项
  - 正则表达式：字符串的生成规则
    - 基本单位
      - . 各种[ ]
      - 匹配任意单个字符，但不匹配换行符
      - [] 用来指定一个字符集。使用连字符来指定字符集的范围。
      - -a-z-0-1-A-Z：匹配字符
    - 前缀&&后缀（补充在基本单位的前面后面）
      - *：基础单位*（基本单位出现0次或一次）
      - ? ：基础单位?（基本单位出现0次或一次）
      - ^：基本单位^（基本单位出现在行首）
      - $：基本单位$（基本单位出现在行尾）
      - 需要匹配任意字符：.*
  - 先用find找文件，在用grep找内容 grep和find的联合使用
- 管道：pipe line
  - 原理
    - 多个控制命令通过隔隔，
    - 将上一个命令的标准输出作为下一个命令的标准输入
    - 将管道内的文件分成单行的输出
  - xargs
    - 将每一行的结果作为cmd2的最后一个参数
    - 先运行cmd2，再将结果作为cmd1的参数
  - 命令替换

## 命令

- 打包和压缩
  - tar [主选项+辅助选项] 目标文件 源文件/目录
    - 主选项
      - -c：创建
      - -r：追加
    - 辅助选项
      - -f：使用档文件
      - -v：显示文件
      - -z：用gzip解压压缩
      - -d：解压
      - -v：显示文件
- scp命令
  - 上传
    - scp 本地文件名 用户名@ip:路径
  - 下载
    - scp 用户名@ip 本地路径
  - 参数
    - -n：行号
    - -E：拓展规则
    - -F：字面通匹配
    - -c：显示匹配总数
    - -i：不区分大小写
  - 正则表达式
    - 匹配的基本单位：
      - 单个字符
      - 括号内部的内容
      - [ ]：或
      - \的转字符
      - 转义字符
      - -：往意字符
      - [ ]：范围内字符
      - {}：字符长度
    - 基本单位的附加情况
      - *：出现在行首
      - $：出现在行尾
      - +：重复0~多次
      - ?：重复0或1次
      - +（x，y）：x~y次
- 文件内容查找
  - grep
- vim
  - 输入/编辑模式
    - 转换到命令模式：esc
    - 剩余的没看
    - 转换编辑模式：a/A/i/I
  - 普通模式
    - h|kl 对应上下左右
    - 删除单位于剪切 删除
    - p：粘贴
    - yy：复制一行（用p粘贴）
    - nyy：复制n行
    - u：撤销
    - ctrl+r 撤销撤销
    - ctrl+d：向下
    - ctrl+u：向上
    - ctrl+f：向前翻页
    - ctrl+b：向后翻页
    - gg：到页面开头部
    - w：往后移动一个光标
    - 查找和替换
      - 几得找目标/得找某个词（）
      - 当/替换目标/(/替换内容)：只对单文件生效
        - s/(替换目标)/(替换内容)/g：当行全部替换
        - %s/(替换目标)/(替换内容)/：忽略大小写
        - -xy s/(替换目标)/(替换内容)/g：修改x~y行的数据
        - -%s/(替换目标)/(替换内容)/g：全文件全部替换
        - 这里的修改目标和修改内容均支持正则表达式
    - 按v为可视化模式，按ctrl+v可以使用数选模式
      - 数选模式之后，按+//+esc
    - %:vxd：将文件转换成二进制
    - w：保存
    - 切换ctrl+w+w
- 批量替换文件
  - shell上批量替换
    - sed "s/(替换目标)/(替换内容)/g" > 文件名
    - 只显示修改结果，不修改实际文件
    - sed "s/(替换目标)/(替换内容)/g" *.c
  - 批量替换：
    - find . -name "*.c" |xargs sed "s/(替换目标)/(替换内容)/g" 利用find命令和管道上部的命令区别多个目标文件
    - 保存修改：sed "s/(替换目标)/(替换内容)/g" > 文件名 -i
    - 文件的对比
      - vimdiff [文件名1] [文件名2]
      - 尽可能的多匹配公共部分
- 编译工具：gcc工具链
  - 程序的编译过程 编译工具：gcc工具链
    - 预处理：将所有的#include头文件以及宏定义替换成其真正的内容
    - 处理之后的是代码文件，后缀为.i
    - 编译：将经过预处理之后的程序转换成特定汇编代码的过程
    - 处理后为汇编代码，后缀为.s文件
    - 汇编：汇编过程将上一步的汇编代码转换成机器码
    - 处理之后为二进制文件，后缀为.o
    - 链接：链接过程将多个目标文以及所需的库文件(.so等)链接成最终的可执行文件。
    - 确定.o文件的地址
    - 函数首先分配栈帧
    - 变量那转换成地址
    - 函数调用是使用其他函数的代码
    - 最终是用跳转实现的
    - 最汇编离要注意：
    - 找自己看得懂的
    - 分块看 找变量的地址
    - 链接的时候将库文件打包到程序里面
  - 库
    - 静态库
      - 生成静态库
      - 1.- gcc -c [文件名] 一此时生成一个文件名.O的文件
      - 2.- ar crsv lib[文件名].a 文件名1.O 打包，文件名为库的名字，生成一个.a文件
      - 3.- 将静态库复制到库路径中：cp lib[文件名].a /usr/lib
      - 4.- gcc -o main main.c -l[库名称]：链接，在编译的时候加上名称
      - 链接的时候将库的文件的位置，在运行的时候再加载到程序里面
    - 动态库
      - 定义并链接动态库
      - -gcc -c [库名].c -fpic
      - -gcc -shared -o lib[库名].so add.o
      - 程序编译时需要加上-g操作
      - 使用要求
      - GDB调试器
- makefile
  - 编译的概念
  - 执行模式
    - 广义：现代预处理+编译+汇编
    - 狭义：现代预处理之后的编译
    - 增量编译
    - make：编译程序
    - 执行和修改时间有关
  - 规则
    - 目标文件：可以是 Object File，也可以是可执行文件
    - 依赖文件：生成目标文件所需要的文件或者目标
    - 命令：- make需要执行的命令 (任意的shell命令)，Makefile中的命令必须以 [tab] 开头。
    - 目标：依赖文件+[tab]+命令
    - 命令是否执行取决于目标与依赖文件的时间关系
    - 命令可以写多个
  - 伪目标
    - rebuild：不像真正的目标那样会生成一个目标文件。
  - 变量
    - A+b和A+=b的区别
    - A + b：运行时替换
    - A := b：定义时替换
    - $(A)自定义变量
    - 变量的作用：进行字符串的替换
    - 预定义变量：可能会随着设备的变化而发生改变
    - 写法和普通变量相同
    - $@：目标文件
    - 自动变量（某单独规则内，不会随着设备的变化而发生改变）
    - $<：第一个依赖文件
    - $^：所有依赖文件
    - $?：新文件（修改时间）
    - $(@D)：目标文件的目录部分if
    - $(@F)：目标文件的文件部分
  - 通配符：模式匹配
    - *：表示任意一个或多个字符
    - ?：表示任意一个字符
    - [...]：ex [abcd] 表示a,b,c,d中任意一个字符，[!abcd]表示除a,b,c,d以外的字符。
    - [0-9]表示 0~9中任意一个数字
    - ~：表示用户的home目录
    - 匹配字符：%
  - 内置函数
    - $(函数名 参数1，参数2)
    - wildcard：对文件系统使用通配符
    - subst：替换函数，用from的值替换to，来源为text
    - $(patsubst %.c ,%.o,text) patsubst
  - 循环结构
    - 使用非避免打印代码

## Linux学习指南

- linux环境
  - 历史
  - 结构
  - 登陆和其他
- linux命令
- 操作系统
  - 如何配置用户
  - git
- 文件系统
  - 概况
  - 目录
  - 文件的细微改
- 文件权限
  - 查找文件：按文件名&&按文件内容
  - 压缩解压
- 远程转交
  - SCP指令上传下载
- vim文本编辑器
- 编译调试工具
  - GCC：编译工具
  - GDB：调试工具
  - 库文件
  - Makefile：工程项目管理器
- 系统编程：操作系统
  - 文件操作：基本的增删查改&&**IO模型之多路复用**
  - 进程
  - 线程
  - 网络编程
  - 服务器框架
  - MySQL数据库：底层&&B树
- 杂七杂八
  - 性能检测工具
  - 性能调优：负载均衡
  - HTTP服务器
  - 五种IO模型
- 项目02：百度网盘
  - 难度不同，量力而为

## 虚拟机的Xshell的配置

- 安装ssh
- 配置Xshell
- 打开之前先ping虚拟机ip
- 优点：可以在windows下操作，并且可以避免图形化占据资源

## Linux历史

- Linux是一个自由开放源码的类UNIX系统，所以我们在说Linux之前，先说说UNIX
  - UNIX
    - UNIX的结构：内核，系统调用，应用程
    - 硬件：进程&&内存&&文件
    - 内核：操作硬件和环境
    - 环境：系统调用
    - shell：命令解释器
    - library routines：公共函数库

## 用户相关命令

- root用户
  - 所有功能
  - 命令提示符为#
- 普通用户
  - 部分功能
  - 命令提示符为 $
- 用户和组 组：用户的集合，方便共享组内资源
- exit：退出用户 找结构储存 先in的后最后退出
- su：switch user
  - 切换用户，如果无参数则切换到根用户
  - -su [其他用户名]
  - 从根用户切换到其他用户不需要密码
- 添加用户
  - 对于“useradd +用户名”而言，会创建一个/bin/bash文件
  - bin/sh鏈于bin/bash
  - useradd -m 用户名 -s bin/bash
  - 对于“useradd +用户名” 不会删除家目录。
  - 需要给底层出现号（找储存）
  - 即将删除的账号不能在用户登陆状中
- 删除命令
  - 同时删除家目录
  - 删除用户
  - 删除root权限
  - 删除密码
  - 删除组
  - 删除用户组
  - 删除文件
  - 删除目录
  - 删除目录及其内容
  - 强制删除
  - 删除空目录