# 1：liunx

## 1.1：基础

### 1.1.1：档案权限与目录配置

chmod：改变权限

各权限分数 r:4 ； w:2 x:1

方法1:chmod [-R] xyz 档案名 #-R是递归整个文件夹里的内容使其一并更改权限

方法2:chmod u=rwx,g=rwx,o=rwx,a=rwx 档案名 # a包括了u,g,o

su 切换用户 exit #离开用户

chgrp #改变档案所属群组 chgrp groupName xxx

chown #改变档案拥有着 chown root xxx

目录：

/etc/passwd #保存所有账户信息

/etc/shadow #保存密码信息

/etc/保存组信息

######################关于预设权限######################

umask 查看预售权限 输出0022

umask -S 输出u=rwx,g=rwx,o=rx

0022的意思我们只关注后三位，意思是被拿掉的权限分，例如022意思就是

用户有全部权限，组被拿掉写入权限，其它用户被拿掉写入权限

注意：档案默认不拥有X权限；

umask 000 #改变预设权限,默认新建的目录用有所有权限

######################特殊权限######################

SUID ,SGID,SBIT ；

SUID：一个用户在执行有此权限的二进制文件时，暂时拥有该文件所有者权限

SGID：...暂时拥有该文件所属群组权限

权限分： SUID:4 SGID:2 SBIT:1

增加特殊权限:chmod u=rwxs,g=rwxs ...

### 1.1.2：防火墙

检查是否安装防火墙 rpm -qa

如果没有安装改组件，可以通过yum install iptables进行安装(必须联网)

查看防火墙状态 service iptables status

立即停止防火墙 service iptables stop

立即启用防火墙 service iptables start

查看端口开放情况 Iptables -L -n

### 1.1.3：其它

复制文件到其它机器 scp ./uc-jsw.zip root@192.168.11.211:/usr/local/

查看进程 ps -ef|grep tomcat

实时输出 tailf xxx.log

查看整个文本 cat xxx

man ls #man用于查看某个命令的用法

ldd /usr/local/nginx/sbin/nginx) #ldd检测某条命令需要哪些共享库文件的支持

ldconfig -p | grep libpcre #查看libpcre动态库是否存在

/sbin/ldconfig #刷新动态库缓存

##########ifconfig 为同一ip配置多个别名##########

ifconfig eth0:0 192.168.1.90 netmask 255.255.255.0 up

### 1.1.4：liunx档案与目录管理

. #此层目录

.. #上层目录

- #前一个工作目录

~ #当前用户家目录

~root #root用户家目录

chgrp:改变文件或档案所属组 chgrp 组名 档案名

chown：改变文件或档案所属用户; chown 用户名 档案名

cp [

-r #递归复制目录

-a #复制全部档案信息] 源路径 新路径

-i #若目标文件已经存在，在覆盖前进行提示

-p #连同档案属性一起复制过去

]

rm [

-r #递归删除

-f #忽略警告信息

-i #二次确认

]

mv[

-f #若目标文件存在，则强制覆盖

-i #若目标文件存在，则询问是否覆盖

-u #若目标文件存在且source比较新，则进行覆盖

]

mget 下载路径 //下载文件至当前目录

cd #变换路径（change directory）

pwd #显示当前路径(print working directory)

mkdir #[p可建立多层路径][m 后面可加权限分数] 建立文件夹

rmdir #[r递归删除，若不加只能删除当前目录，且必须为空目录] 删除目录

echo #打印，列出

$PATH #$表示后面接的是变量

echo $PATH #打印的是可执行文件(命令)的搜索路径，PATH="$PATH":/root 意思是将root目##录加入path中；

ls [

-a #包括隐藏档案全部档案

-l #显示完整档案内容

-h #显示档案容量

-r #将排序结果反向输出

-t #按时间排序

-i #显示完整的档案名，显示inode

-S #按容量大小排序

--full-time #时间格式化显示

--time=mtime|ctime|atime #显示的时间类型mtime是modification time缩写，当内容数据变更时，此时间将会更新，默认显示此类型时间； ctime是status time，当档案状态比如权限等改变时，此时间将改变；atime是access time，内容被读取时，时间改变

]

输入 basename|dirname  /etc/yum/version-groups.conf

basename #取得最后档案名 输出：version-groups.conf

dirname #取得路径名 输出： /etc/yum

############文档内容检索####################

cat #由第一行开始显示档案内容

[

-A #显示特殊符号

-n #显示行号

]

tac #由最后一行开始显示

nl #输出行号显示

head # head -2 xxx 从头开始显示两行

tail # tail -2 xxx 从尾开始显示2行

od #以二进制显示文件内容

more #页一页显示档案内容

[

空格键:向下翻一页

回车键：向下翻一行

:f 立即显示当前行

q 立即离开More

b 往回翻页

/ 输入/后再输入要搜索的字符串再按回车

n 继续往下搜索同字符串

]

less#一页一页显示档案内容

touch #建立新的档案

file #查看文档类型

####################文档检索####################

连续按两次tab键可以列举可执行指令

which #搜索可执行指令路径 ，例如 which redis-cli

[

-a #列出path下所有能匹配到该指令的路径。默认只列出首次匹配的

]

whereis #在数据库中搜寻档案 ,例如 whereis aaa

find :在硬盘中搜索







### 1.1.5：liunx磁盘与文件系统

df [指定目录] #列出文件系统使用状况

[

-k #以kByte容量显示文件系统

-m #以mByte容量显示文件系统

-h #以G/M/K等能读懂的单位显示容量

-T #显示文件系统类型，比如EXT3...

-i #不用硬盘容量，而用inode的数量来显示

]

du -sh #列出此目录下总使用容量

ln 源档案名 链接档案名 #建立实体链接，例如 ： ln test/cy testLn

ln -s 源路径或者档案名 链接路径 #建立快捷方式

fdisk -l ：显示磁盘信息

####################挂载ISO文件####################

1>建立挂载点

mount -o loop /tmp/isoloop xxx.iso

这是用df就可以看到效果了，可以进入/tmp/isoloop查看iso文件内容

####################建立大档案进行挂载####################

1>首先建立一个大型档案

dd if=/dev/zero of=/tmp/swaploop bs=1M count=512

bs是block大小，count是总共几个bs

2>格式化档案(就像是格式化磁盘分区)

mkfs -t ext3 /tmp/swaploop

3>建立挂载点

mount -o loop /tmp/swaploop /tmploop/ #将/tmp/swaploo这个档案(磁盘分区)挂载到/tmploop

umount /tmploop #卸载挂载点

########################配置swap内存置换空间######################

1>用dd建立一个大型档案

dd if=/dev/zero of=/tmp/swaploop bs=1M count=5000

2>将档案格式化成swap文件格式

mkswap /tmp/swaploop

3>启动swap

swapon /tmp/swaploop

swapoff /tmp/swaploop #关闭swap

接下来可以使用free查看内存信息看到效果！

### 1.1.6：档案与文件系统的压缩与打包

常见压缩文件格式

.gz:gzip程序压缩的档案

.gz2:bzip2程序压缩的档案

.tar:tar程序打包的档案，未压缩

.tar.gz:tar打包且经过gzip压缩

.tar.bz2：tar打包且经过bzip2压缩

gzip|bzip2

[

-v #压缩文件指令

-d #解压缩指令

-number #压缩等级，默认为6，最高为9，-1最快压缩最差，9最慢压缩最好

]

zcat|bzcat: 如果压缩的是文本文件，可以用zcat直接查看； 例如：zcat cy.gz

tar #打包指令  
[

-c #建立打包档案

-v #在压缩/解压过程中，输出正在处理的文件名

-t #查看档案内目录数据

-x #解压缩

-z #通过gzip压缩或解压缩

-j #通过bzip2压缩或解压缩

-C #解压缩到特定目录

-f #要处理的档案名，如果是压缩，则是压缩后的文件，比如xxx.tar.gz

-p #保留备份数据的原本权限与属性

-P #

--exclude=FILE #在压缩过程中，排除指定文件

]

打包：tar -cv -f result.tar sourceName

打包并压缩：tar -cvz -f 打包并压缩：tar -cvz -f result.tar.gz sourceName

.tar.gz sourceName

打包并压缩：tar -cvj -f result.tar.bz2 sourceName

解包到当前目录：tar -xvf result.tar

解包到指定目录：tar -xv -C xxx -f result.tar

如果是打包并压缩的就可以使用:tar -xvz|j f...

查看文档内容：tar -tvf result.tar

##############只解压指定文件：###############

1>先找到要解压的文件

tar -tvzf result.tar.gz |grep 文件名

假设要解压testgzip/t1这个文件到result文件下

接下来：tar -xzv -C result -f testgzip.tar.gz testgzip/t1

###########一边压缩一遍解压缩到当前目录########

cd /usr/cy

tar -cvzf - ../testgzip|tar -xvzf -



### 1.1.7：VIM程序编辑器

**vi命令大全：**













vim编辑器





### 1.1.8：shell编程

alias cy=”ls -al” #指令别名

## 1.2：服务安装

### 1.2.1：第三方包通用命令

用于linux源码安装软件，一般下载源码包得到文件：xxxx.tgz

解包软件 tar zxf xxxx.tgz

配置 cd xxxx ./configure

编译 make

安装 make install

卸载 make uninstall

## 1.3：网络相关

重启网卡 service network restart

## 1.4：系统监控

# 2：windows

# 3：mysql

## 3.1：随记

删除索引：DROP INDEX indexName ON dbName.tableName

查看索引：SHOW INDEX FROM tableName

创建索引：

ALTER TABLE dbName.tableName ADD INDEX indexName(columnName);

如果索引不存在则增加索引：

USE prms\_db;

DROP PROCEDURE IF EXISTS index\_add;

DELIMITER ;;

CREATE PROCEDURE index\_add()

BEGIN

DECLARE CurrentDatabase VARCHAR(100);

SELECT DATABASE() INTO CurrentDatabase;

IF NOT EXISTS (SELECT \* FROM information\_schema.statistics WHERE table\_schema=CurrentDatabase AND table\_name = 'panicbuy\_remind' AND index\_name = 'index\_commodity\_id') THEN

ALTER TABLE prms\_db.panicbuy\_remind ADD INDEX index\_commodity\_id(commodity\_id);

ELSE

SELECT 'exist';

END IF;

END;;

DELIMITER ;

CALL index\_add();

删除表：DROP TABLE IF EXISTS `tableName`;  
插入数据：INSERT IGNORE INTO tableName (columnName1,columnName2) values (‘’,’’);

新增字段：

-- 创建一个存储过程判断字段是否存在

drop PROCEDURE if EXISTS add\_column;

DELIMITER ;;

create PROCEDURE add\_column()

BEGIN

-- TABLE\_SCHEMA 数据库名

-- table\_name 表名

-- COLUMN\_NAME 列名

if not EXISTS(select 1 FROM information\_schema.COLUMNS WHERE TABLE\_SCHEMA='cims\_db' AND table\_name='on\_shelf' AND COLUMN\_NAME='status\_type') then

-- 要执行的字段新增

ALTER TABLE `cims\_db`.`on\_shelf`

ADD COLUMN `status\_type` VARCHAR(50) NULL DEFAULT '0' COMMENT '状态类型' AFTER `status`;

ELSE

select 'exist';

end if;

end

;;

DELIMITER ;

CALL add\_column();

-- 结束存储过程创建

**创建表：**

CREATE TABLE IF NOT EXISTS test\_proc(

id VARCHAR(32) NOT NULL COMMENT '主键id',

default\_name VARCHAR(100) NULL DEFAULT NULL COMMENT '测试名',

PRIMARY KEY(id),

INDEX name\_index (default\_name)

)

ENGINE = INNODB

DEFAULT CHARSET = utf8

COMMENT = '测试存储过程';

**删除表**：DROP TABLE IF EXISTS test\_proc;

查看所有存储过程： SHOW PROCEDURE STATUS

删除存储过程：drop PROCEDURE if EXISTS name

查看存储过程创建代码:SHOW CREATE PROCEDURE name;

删除表：DROP TABLE IF EXISTS tableName

**2 .数据操纵语言DML**  
数据操纵语言DML主要有三种形式：  
1) 插入：INSERT  
2) 更新：UPDATE  
3) 删除：DELETE

1. **数据定义语言DDL**  
   数据定义语言DDL用来创建数据库中的各种对象-----表、视图、  
   索引、同义词、聚簇等如：  
   CREATE TABLE/VIEW/INDEX/SYN/CLUSTER

## 3.2：函数以及存储过程记录

==============通用创建索引存储过程==========

DELIMITER $$

USE `prms\_db`$$

DROP PROCEDURE IF EXISTS `add\_index\_proc`$$

CREATE DEFINER=`root`@`%` PROCEDURE `add\_index\_proc`(t\_name VARCHAR(100),\_index\_name VARCHAR(100),\_index\_type VARCHAR(100),\_index\_method VARCHAR(100),col\_names VARCHAR(100))

SQL SECURITY INVOKER

BEGIN

DECLARE CurrentDatabase VARCHAR(100);

DECLARE sql\_str VARCHAR(1000);

DECLARE default\_index\_type VARCHAR(100) DEFAULT 'index';

DECLARE flag INT;

SELECT DATABASE() INTO CurrentDatabase;

SELECT 1 FROM information\_schema.statistics WHERE table\_schema=CurrentDatabase AND table\_name = t\_name AND index\_name = \_index\_name ;

SELECT CurrentDatabase,t\_name,\_index\_name,flag;

IF NOT EXISTS(SELECT 1 FROM information\_schema.statistics WHERE table\_schema=CurrentDatabase AND table\_name = t\_name AND index\_name = \_index\_name ) THEN

SET @default\_index\_type='index';

SET @sql\_str=CONCAT('ALTER TABLE ',t\_name);

IF \_index\_type!= '' THEN

SET @default\_index\_type=CONCAT(\_index\_type,' ',@default\_index\_type);

END IF;

SET @sql\_str=CONCAT(@sql\_str,' add ',@default\_index\_type,' ',\_index\_name,'(',col\_names,')');

SELECT @sql\_str,\_index\_type,default\_index\_type,col\_names;

IF \_index\_method!= '' THEN

SET @sql\_str=CONCAT(@sql\_str,' using ',\_index\_method);

END IF;

PREPARE st FROM @sql\_str;

EXECUTE st;

DEALLOCATE PREPARE st;

END IF;

END$$

DELIMITER ;

**调用示例:**

**CALL add\_index\_proc('表名,'索引名','索引类型','',’字段名');**

==============通用添加字段存储过程==============

DELIMITER $$

USE `prms\_db`$$

DROP PROCEDURE IF EXISTS `add\_column\_proc`$$

CREATE DEFINER=`root`@`%` PROCEDURE `add\_column\_proc`(t\_name VARCHAR(100),col\_name VARCHAR(100),datatype VARCHAR(100),remark VARCHAR(255))

SQL SECURITY INVOKER

BEGIN

DECLARE CurrentDatabase VARCHAR(100);

DECLARE sql\_str VARCHAR(1000);

SELECT DATABASE() INTO CurrentDatabase;

IF NOT EXISTS(SELECT 1 FROM information\_schema.COLUMNS WHERE TABLE\_SCHEMA=CurrentDatabase AND table\_name=t\_name AND COLUMN\_NAME=col\_name) THEN

SET @sql\_str=CONCAT('ALTER TABLE ',t\_name ,' ADD COLUMN ' ,col\_name,' ', datatype,' DEFAULT NULL NULL COMMENT \'',remark,'\'');

PREPARE st FROM @sql\_str;

EXECUTE st;

DEALLOCATE PREPARE st;

END IF;

END$$

DELIMITER ;

**调用示例：**

**CALL add\_column\_proc('表名','列名','数据类型','备注');**

# 4：java

# 5：spring

## 5.1：总览

spring呢主要有ioc,aop,事务处理，jdbc，orm，与hibernate，mybatis等的集成，远程调用，还有众多实用的模板，比如sql，redis，mq等等的模板

## 5.2：ioc容器



# 6：redis

## 6.1:redis安装

cd usr/local/down/

wget http://download.redis.io/releases/redis-3.0.7.tar.gz #下载tar.gz文件到当目录

tar xzf redis-3.0.7.tar.gz

yum install gcc

make PREFIX=/uer/local/redis install &&　make　#安装到指定目录

cd /usr/local/redis

cp /usr/local/down/redis-3.0.7/utils/redis\_init\_script /etc/rc.d/init.d/redis #复制脚本到注册表

mkdir /etc/redis

cp /usr/local/down/redis-3.0.7 /etc/redis/6379.conf

#设置密码

vi /etc/redis/6379.conf

#requirepass foobared去掉注释，foobared改为自己的密码

#启动服务

service redis start

开启端口 -A INPUT -m state --state NEW -m tcp -p tcp --dport 6379 -j ACCEPT

重新加载防火墙规则 service iptables restart

#链接redis redis-cli

# 7：nginx

## 7.1：安装nginx:

官网路径：nginx.org/download/nginx-1.10.2.tar.gz

cd /usr/local/download

#安装c++环境

yum install -y gcc gcc-c++

#安装pcre库

wget http://nchc.dl.sourceforge.net/project/pcre/pcre/8.39/pcre-8.39.tar.gz

tar -xvf ...

./configure

make && make install

#安装zlib库

wget <http://zlib.net/zlib-1.2.8.tar.gz>

tar -xvf ...

./configure

make && make install

#安装openSSL

wget https://www.openssl.org/source/openssl-1.1.0c.tar.gz

...

#依赖库安装完毕，开始安装nginx

wget nginx.org/download/nginx-1.10.2.tar.gz

tar -xvf nginx-1.10.2.tar.gz

cd ./nginx-1.10.2

./configure

默认会安装在 /usr/local/nginx

## **7.2：启动，停止，重启ng**

cd /usr/local/nginx

目录下有4个文件夹，其中sbin里存放的就是ng服务器的主程序

1>启动服务

./sib/nginx

2>停止服务

kill -TERM|INT|QUIT|HUP port #TERM或INT是快速停止ng服务，丢失所有处理中请求，QUIT是平缓停止服务，处理完正在处理的请求，但不接受新的请求。HUP是使用新的配置文件启动进程，之后平缓停止原有进程。也就是所谓的平缓重启；

## 7.3：nginx.conf配置

1：全局指令

user :指定可以运行nginx服务的用户 列：user root;

worker\_processes：指定同时允许创建的工作进程; 列 worker\_processes 3|auto;

pid :指定nginx主进程号存放路径

error\_log:异常日志Log保存配置 ,此指令可在http块，server和location块中配置

例如error\_log logs/error.log debug|info|notice|warn|error|crit|alert|emerg

include :引入其它配置文件,此指令可放任意地方 ；列：include /usr/xxx.conf

2：events配置

acept\_mutex on|off ；默认为On，对process进行序列号，防止多个进程对连接争抢

multi\_accept on|off ;设置是否允许同时接受多个网络请求，默认为off ，off是每个process一次只能接受一个新到达的网络连接

worker\_connections number:每一个process同时开启的最大连接数

use epoll: 配置事件驱动模型

3:http配置

access\_log filePath [formatName]：访问日志,filePath是日志保存路径，formatName是日志格式化字符串名称

sendfile\_max\_chunk 128k;设置每个process调用sendfile()传输的数据量不能超过这个值，默认为0不限制

keepalive\_timeout outTime [keep-alive]：设置服务端对连接的保持时间，keep-alive是设置客户端保持连接时间 ； 例如：keepalive\_timeout 120s 120s ; 该指令可在http，server,location中配置，第二个时间参数是客户端主动与服务器断开连接的事件

keepalive\_requests number配置单连接请求连接上线；默认为100

nginx是一种模块化的结果，基本可以分为核心模块，标准http模块，可选http模块，邮件服务模块和其它第三方模块

send\_timeout #服务器等待时间，也就是会话超时时间 ，例如 send\_timeout 10s;

client\_header\_buffer\_size #允许接受客户端请求头的缓冲区大小,

如： client\_header\_buffer\_size 4k;

## 7.4：nginx的模块化

查看ng当前支持的模块：

cd目录到下载的ng编译后文件所在地，cd ./objs ; cat ngx\_modules.c

# 8：tomcat

# 9：系统运维

# 10：cas

# 11：ftp

# 12：rabbit

# 13：dubbo

# 14：email

# 15：zookeeper

# 16：需要解决的问题记录

1：同步，异步，阻塞，非阻塞等具体概念的理解

2：liunx内核参数和对nginx 的优化

3：squid缓存服务器