# 1：liunx

## 1.1：基础

### 1.1.1：文件系统

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

rm [r:递归删除,f:忽略警告信息,i:二次确认] 文件名 或者 name\*

mget 下载路径 //下载文件至当前目录

. 代表此层目录

.. 代表上层目录

- 代表前一个工作目录

~ 代表【目前用户】所在的家目录

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

### 1.1.2：权限系统

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

方法2:chmod u=rwx,go=rwx,a=rwx 档案名

su 切换用户

mkdir[-mp] name ：-m：预设权限 ； -p递归建立所有文件夹 ；如：mkdir -m 774 name1 ; mkdir -p name1/name2/name3

### 1.1.3：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

### 1.1.4：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

安装ftp yum -y install vsftpd ftp

**编辑文本 vi**

**移动光标类命令**h ：光标左移一个字符   
l ：光标右移一个字符   
space：光标右移一个字符   
Backspace：光标左移一个字符   
k或Ctrl+p：光标上移一行   
j或Ctrl+n ：光标下移一行   
Enter ：光标下移一行   
w或W ：光标右移一个字至字首   
b或B ：光标左移一个字至字首   
e或E ：光标右移一个字至字尾   
) ：光标移至句尾   
( ：光标移至句首   
}：光标移至段落开头   
{：光标移至段落结尾   
nG：光标移至第n行首   
n+：光标下移n行   
n-：光标上移n行   
n$：光标移至第n行尾   
H ：光标移至屏幕顶行   
M ：光标移至屏幕中间行   
L ：光标移至屏幕最后行   
0：（注意是数字零）光标移至当前行首   
$：光标移至当前行尾   
  
 **屏幕翻滚类命令**Ctrl+u：向文件首翻半屏   
Ctrl+d：向文件尾翻半屏   
Ctrl+f：向文件尾翻一屏   
Ctrl＋b；向文件首翻一屏   
nz：将第n行滚至屏幕顶部，不指定n时将当前行滚至屏幕顶部。

**搜索及替换命令**   
/pattern：从光标开始处向文件尾搜索pattern   
?pattern：从光标开始处向文件首搜索pattern   
n：在同一方向重复上一次搜索命令   
N：在反方向上重复上一次搜索命令

## 1.2：服务安装

### 1.2.1：redis的安装

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

## 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

# 7：nginx

# 8：tomcat

# 9：系统运维

# 10：cas

# 11：ftp

# 12：rabbit

# 13：dubbo

# 14：email

# 15：zookeeper