Linux：

https://www.kernel.org Linux内核官网

操作系统：

Unix - Minix - Linux

第一台计算机 - Pascal发明

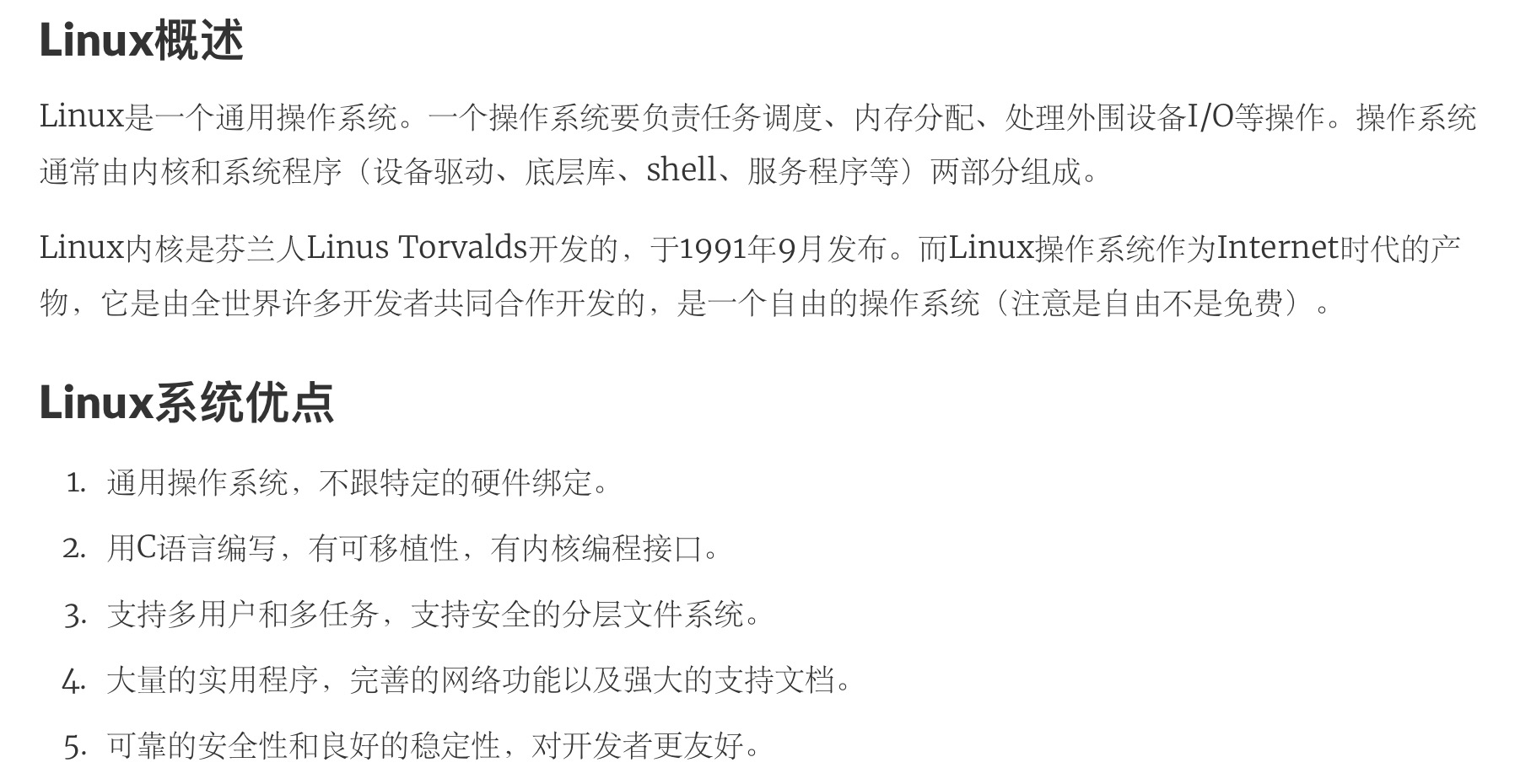
第一台数字计算机 - 差分机 - 由程序驱动 - Ada（第一个程序员）

第一台电子数字计算机 - ENIAC - 冯诺依曼体系

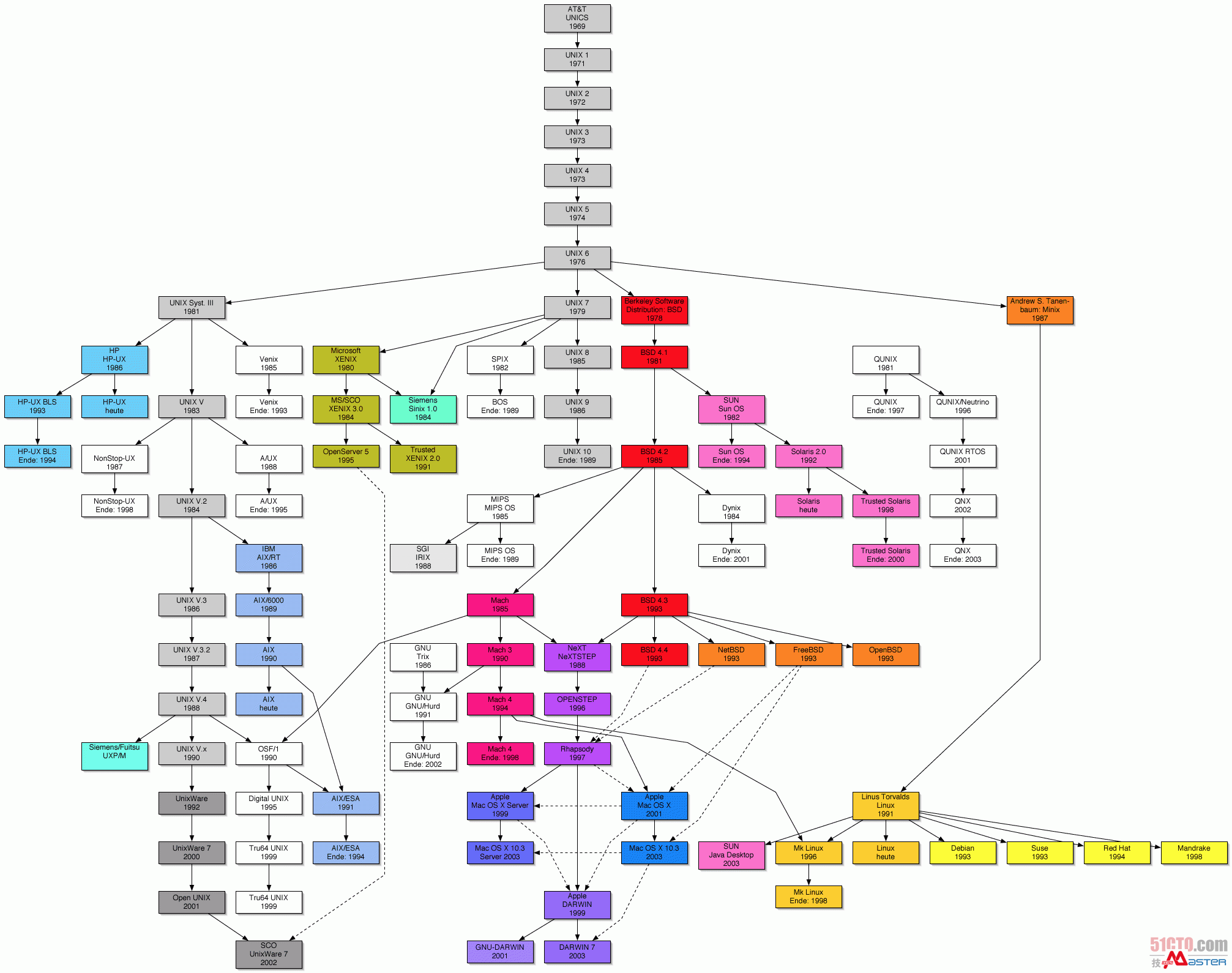
图灵

Linux发行版本：

RedHat Linux、Ubuntu、CentOS



操作系统发展史图：



Linux（centos7）命令： / 表示 或者

who / w：查看当前登录的所有用户

who am i / whoami：查看当前登录用户

last：查看之前登录过的用户历史记录

clear：清屏

ps：查看进程状态

ps -ef：查看进程

ps -aux：查看进程

kill 进程号：杀掉进程

netstat -nap：查看进程、端口、进程号

adduser 用户名：创建新用户

userdel 用户名：删除用户

passwd [用户名]：修改用户密码，不加用户名是修改的root的密码

ssh 用户名@ip地址：远程安全连接其他的服务器

logout / exit：退出

reboot：重启服务器

shutdown：关闭服务器

init 0：关机

init 6：重启

uname：查看当前系统

hostname：查看主机名

history：查看历史命令，默认保存最近1000条，可以使用 HISTSIZE=新值 来修改保存的条数

！num(命令行号)：重新执行改行命令

man 命令：查看’命令‘的使用方式

info 命令：查看’命令‘的使用方式，更详细

命令 --help：查看’命令‘的参数使用方法

whatis 命令：查看命令的简短描述

pwd：打印当前工作目录路径 print working directory

whereis 命令：查找包含 ’命令‘ 所有的目录路径

which 命令：查找到第一个含有 ’命令‘ 的目录路径

su 用户名：切换到该用户，但是不会切换到该用户的文件目录

su - 用户名：切换到该用户，并且切换到该用户的文件目录

sudo ...：以管理员身份执行

mkdir 目录名：创建目录

rmdir 目录名：删除空目录

rm 目录名 / 文件名：删除目录或者文件

-i：交互式删除，有提示 interactive

-f：不用提示，直接删除 force

-r：递归删除目录下的内容 recursive

ls：查看当前目录下的所有文件和目录（非隐藏）

-l：查看当前目录下的所有文件和目录，长格式（非隐藏）

-a：查看当前目录下的所有文件和目录

-al / -l -a：查看当前目录下的所有文件和目录，长格式

-r：反转显示目录下的文件和目录，按字母表降序

-R：递归显示目录下的文件和目录

touch 文件名：没有该文件就创建一个空文件，有只是改变其时间戳，可以修改文件的三个时间：修改内容的时间，更改权限的时间，最后访问的时间

cd：改变目录 change directory

cd /root：回到root主目录 （/root 绝对路径）

cd ~：回到当前用户主目录

cd ..：回到上一级

cd /：回到根目录

/etc：放置配置文件目录

/home：其他用户的用户主目录

/usr：用户数据、用户应用目录

cp 文件名 路径：将文件拷贝到路径下，可以重命名

scp 本地文件 用户名@ip地址:路径：将本地文件安全拷贝到目标服务器的某路径下

sftp 用户名@ip地址：安全的远程文件传输

mv 文件名/目录 路径：将文件/目录剪切（移动）到路径下，可以重命名

cat 文件名：查看文件所有内容

-n：查看文件内容并显示行号

| more/less：一页一页查看

| grep pattern：搜索文件中的内容，匹配模式pattern匹配的字符串

head -n 文件：查看文件前 n 行

tail -n 文件：查看文件最后 n 行

wget 资源路径：联网获取资源

wget -O filename 资源路径：给资源重命名

grep pattern 文件1：搜索文件1中的内容，匹配模式pattern匹配的字符串

-n：显示内容匹配的行号

-R：递归搜索，并显示文件路径

&：后台执行

> 文件2：输出重定向，将运行结果输出到文件2

2> 文件3：错误重定向，将错误输出到文件3

>> 文件4：追加输出重定向，将运行结果追加输出到文件4

<：输入重定向

find：查找文件或目录

jobs：查看正在执行的后台任务列表

fg %n：将编号为 n 的后台任务拿到前台执行

bg %n：将编号为 n 的后台暂停任务继续在后台执行

top：查看CPU的状态

wc 文件：统计文件的行数、单词数、字符数 word count

-l：行数

-w：单词数

uniq 文件：将文件中的内容去重显示，注意只能去相邻的重复的，不改变原文件内容

sort 文件：将文件中的内容进行排序显示，不改变原文件内容

diff 文件1 文件2：比较两个文件的差异，并显示不同之处

file 文件：显示文件的性质

date：查看当前时间

cal：查看日历

echo：回声命令

echo string：在屏幕上输出string

echo string > 文件：将string输出到文件中

echo $a：在屏幕上打印变量a的值

wall string：给用户发送信息string

ln：链接

ln 路径1/文件1 路径2/文件2：给路径1下的文件1在路径2下创建名为文件2的硬链接，存储的是文件1内容的引用，相当于备份，但是不会占用空间

ln -s 路径1/文件1 路径2/文件2：给路径1下的文件1在路径2下创建名为文件2的软链接，相当于快捷方式，不是引用

gzip 文件名：压缩

gunzip 文件名：解压缩

xz：压缩、解压缩

-z 文件名：压缩

-0 ... -9：压缩比，值越大，压缩率越大

-d 文件名：解压缩

tar：归档

-x：抽取，解归档

-v：显示解归档的过程

-f：指定文件名

-xvf 文件名：解归档使用

-t：查看归档文件的内容

-c：归档

-cvf 归档文件名 源文件路径：归档使用

alias 名称='命令 [参数]'：给 ‘命令 参数’ 起别名

unalias 名称：取消别名

!v：把刚刚以v开头的命令在执行一遍

vi / vim 编辑器的使用：

i / a：进入编辑模式

ESC：退出编辑模式

yy：复制本行

num yy：复制num行

p：粘贴

dd：删除本行

num dd：删除num行

dw：删除一个单词

dG：删除所有内容

d$：从当前位置删除到行末尾

u：撤销

G：去到最后一行

num G：去到num行

gg：回到第一行

Ctrl y：向上翻一行

Ctrl e：向下翻一行

Ctrl f：向下翻一页

Ctrl b：向上翻一页

ZZ：保存并退出

:!系统命令：在vim编辑器执行系统命令，执行完后回车，回到vim编辑器 eg： :!cal

:wq：保存并退出

:q!：不保存退出

:set nu：显示行号

:set nonu：取消行号

:syntax off：关闭高亮语法 :syntax on：开启高亮语法

:set ts=4：将制表键设置为4个空格

/string ：搜索string， 按 n 键向下搜索， 按 N 键向上搜索

:1,$s/string1/string2/g：将文件中从第一行到最后一行的所有string1换成string2

自定义快捷键：

:inoremap 快捷键名 对应的代码 eg： :inoremap pymain if \_\_name\_\_ == '\_\_main\_\_'

文件、目录权限：

- rwx rwx rwx:

类型 所有者权限 同组用户权限 其他用户权限

类型：文件（-）、目录（d）、链接（l）

r：read 4 - 0b(100)

w：write 2 - 0b(10)

x：excute 1 - 0b(1)

chmod 权限 文件/目录：修改文件/目录的权限

chmod 777 文件/目录

HTTP：

LAMP - Linux Apache MySQL PHP

LNMP - Linux Nginx MySQL Python

MySQL：

为什么要使用数据库

1、数据持久化 —— 将数据从一掉电数据就丢失的存储介质（内存）转移到持久存储介质（硬盘）

2、高效的存储和管理数据，方便检索数据

关系型数据库

理论基础：集合论和关系代数

用二维表组织数据 —— 行（记录）、列（字段），一张表称为一个实体

表与表之间有关系

SQL —— 结构化查询语言

Database - DB - 数据的仓库

DBS - 包括了DB，DBA，(R)DBMS

(R)DBMS - 数据库管理系统

- MySQL - 小巧而强大

- Oracle - 强大而且提供商业智能

- DB2(IMB) - 安全强大而且提供商业智能

- SQL Server

- SQLlite - 嵌入式关系型数据库（移动端）

关系型数据库的完整性：

1、实体完整性：每条记录都是独一无二的（主键、唯一约束、唯一索引）

2、参照（引用）完整性：表中的数据要参照其他表中已有的数据（外键）

3、域完整性：数据是有效的（数据类型、非空约束、默认值约束、检查约束）

SQL：

SQL(Structured Query Language)

关系型数据库的编程语言

-- DDL(数据定义语言): create(创建) / drop(删除) / alter(修改)

-- DML(数据操纵语言): insert(插入) / delete(删除) / update(更新)

-- DQL(数据查询语言): select(选择)

-- DCL(数据控制语言): grant / revoke / begin / commit / rollback

基本语句

select - 从数据库中提取数据

update - 更新数据库中的数据

delete - 从数据库中删除数据

insert into - 向数据库中插入新数据

create database - 创建新数据库

alter database - 修改数据库

create table - 创建新表

alter table - 变更（改变）数据库表

drop table - 删除表

create index - 创建索引（搜索键）

drop index - 删除索引

Python连接MySQL数据库：

**class** MySqlHelper(object):

*"""MySQLHelper"""*

**def** \_\_init\_\_(self, connect\_config):

*"""*

*构造器*

**:param** *connect\_config: 连接配置，dict*

*"""*

self.connect\_config = connect\_config

self.conn = **None**

self.cursor = **None**

**def** connect\_db(self):

*"""创建数据库连接"""*

self.conn = pymysql.connect(\*\*self.connect\_config)

self.cursor = self.conn.cursor()

**def** close\_db(self):

*"""关闭数据库连接"""*

**if** self.cursor:

self.cursor.close()

**if** self.conn:

self.conn.close()

**def** execute\_sql(self, sql, \*, param=**None**):

*"""*

*执行dql操作，即 select 语句*

**:param** *sql: sql语句，string*

**:param** *param: 参数列表，dict*

**:return***: 查询结果，list*

*"""*

res = **''**

**try**:

self.connect\_db()

self.cursor.execute(sql, param)

res = self.cursor.fetchall()

**except** BaseException **as** e:

print(e)

**finally**:

self.close\_db()

**return** res

**def** execute\_dml(self, sql, \*, param=**None**):

*"""*

*执行dml操作，即 update、delete、insert 语句*

**:param** *sql: sql语句，string*

**:param** *param: 参数列表，dict*

**:return***: 执行结果，int [1：成功，0：正常失败，-1：错误失败]*

*"""*

**try**:

self.connect\_db()

count = self.cursor.execute(sql, param)

self.conn.commit()

**if** count:

res = 1

**else**:

res = 0

**except** BaseException **as** e:

print(e)

self.conn.rollback()

res = -1

**finally**:

self.close\_db()

**return** res

测试：

**def** main():

config = {

**'host'**: **'localhost'**,

**'user'**: **'root'**,

**'password'**: **'root'**,

**'database'**: **'hrs'**,

**'port'**: 3306,

**'charset'**: **'utf8'**,

**'autocommit'**: **False**,

**'cursorclass'**: pymysql.cursors.DictCursor *# cursorclass设置cursor游标的类型*

}

sqlhelper = MySqlHelper(config)

sql = **'select dno, dname, dloc from tbDept where dno=%(no)s'**

param = {

**'no'**: 10

}

res = sqlhelper.execute\_sql(sql, param=param)

print(res)

sql = **'insert into TbDept values (%(no)s, %(name)s, %(loc)s)'**

param = {

**'no'**: 80,

**'name'**: **'sda'**,

**'loc'**: **'DSfcz'**

}

res = sqlhelper.execute\_dml(sql, param=param)

print(res)

**if** \_\_name\_\_ == **'\_\_main\_\_'**:

main()