# Git

# 软件模式

<https://design-patterns.readthedocs.io/zh_CN/latest/>

这个中文网址写的很好。这里提到了三个层次，非常贴切。

a) 能在白纸上画出所有的模式结构和时序图。

b) 能用代码实现；如果模式的代码都没有实现过，是用不出来的；即所谓，看得懂，不会用。

c) 灵活应用到工作中的项目中。

## UML类图和时序图



* 车的类图结构为<abstract>，表示车是一个抽象类；
* 它有两个继承类：小汽车和自行车；它们之间的关系为实现关系（realize），使用带空心箭头的虚线表示；
* 小汽车为与SUV之间也是继承关系，它们之间的关系为泛化关系（generalize），使用带空心箭头的实线表示，泛化关系表现为继承非抽象类；
* 小汽车与发动机之间是组合关系，使用带实心箭头的实线表示；
* 学生与班级之间是聚合关系，使用带空心箭头的实线表示；
* 学生与身份证之间为关联关系，使用一根实线表示；
* 学生上学需要用到自行车，与自行车是一种依赖关系，使用带箭头的虚线表示；

笔记：在梳理这些关系的时候，通过假设是否存在来分析两者之间的关系。

## 时序图

<https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/?ref=lbp>

通过箭头来查看异步消息。





有时候时序图的箭头很让人困惑。把两个对象理解成两个人在交谈，Sender: Hi，Welcome. Receiver: Got it (也许收到了，但是不作回答)。

## 状态图



**Transition –** We use a solid arrow to represent the transition or change of control from one state to another. The arrow is labelled with the event which causes the change in state.

导致，是event导致了状态的变化。

## 活动图

也就是流程图，用于流程控制。用例图只能用来表示high level 的内容。

 An activity diagram can be used to illustrate a business process (high level implementation) to a stand alone algorithm (ground level implementation). However, Use cases have a low level of abstraction. They are used to show a **high level** of implementation only.



例如下图 – 当蒸牛奶和添加咖啡这两项活动都完成后，我们将它们合并为一项最终活动。这里面有Fork和Join。



# Linux

## CentOS

CentOS适用于服务器。

需要参加红帽RHCSA、RHCE、RHCA认证，肯定要安装RHEL或CentOS系统。

## Ubuntu

Ubuntu则适用于个人桌面。但其实Ubuntu在服务器方面也并不比CentOS弱，并且ubuntu 相比 centos 在更新策略上要积极一些。

### Xubuntu

Xubuntu（发音为ZOO-bun-too）是一个Ubuntu Linux的官方派生版，它基于桌面环境Xfce，主要运行基于GTK+的程序，面向旧式电脑的用户和寻求更快捷的桌面环境的用户。

它与Ubuntu不完全相同，但使用Ubuntu的高质量软件源。

Xubuntu是一个完整的基于Ubuntu的GNU/Linux发行版，但是更为轻量，比使用Gnome和KDE的Ubuntu系统更有效率，因为其使用了Xfce桌面环境。

注意：有些Ubuntu的功能在Xubuntu里面被裁减了，比如gnome-system-monitor

### 安装问题

### 远程和传输文件的问题

## Command

<https://www.runoob.com/linux/linux-command-manual.html>

这个连接包含绝大部分Linux命令

|  |  |
| --- | --- |
| cd ../.. | 回到更目录 |
| service --status-all  service --status-all | grep '\[ + \]'  systemctl list-units  systemctl list-units -a  systemctl list-units -a --state=inactive | show all available services  systemctl list-units 是linux15以后才能用的 |
| mount  mount | grep /dev/sd  lsblk  lspci,lspci -v，-vv, -vvv  lsusb | * **ls**: List files in the file system. * **lsblk**: List block devices (for example, the drives). * **lspci**: List PCI devices. * **lsusb**: List USB devices. * **lsdev**: List all devices. |
| sudo ufw status |  |
| sudo ip addr add 10.102.66.200/24 dev enp0s25 | <https://ubuntu.com/server/docs/network-configuration>  /24 is required, enp0s25 is device which may different within PCs. |
| - shutdown -h now //立刻关机  - shutdown -h 1 //一分钟后关机  - shutdown -r now //立即重启 |  |
| su | （英文全拼：switch user） |
| sudo dpkg -i /home/\*.deb | 安装程序 |

* chmod

Usage: chmod [OPTION]... MODE[,MODE]... FILE...

or: chmod [OPTION]... OCTAL-MODE FILE...

or: chmod [OPTION]... --reference=RFILE FILE...

Change the mode of each FILE to MODE.

With --reference, change the mode of each FILE to that of RFILE.

-c, --changes like verbose but report only when a change is made

-f, --silent, --quiet suppress most error messages

-v, --verbose output a diagnostic for every file processed

--no-preserve-root do not treat '/' specially (the default)

--preserve-root fail to operate recursively on '/'

--reference=RFILE use RFILE's mode instead of MODE values

-R, --recursive change files and directories recursively

--help display this help and exit

--version output version information and exit

Each MODE is of the form '[ugoa]\*([-+=]([rwxXst]\*|[ugo]))+|[-+=][0-7]+'.

## 操作技巧

翻页：shift+page up，

快捷键： 打开主菜单 = Alt + F1   
\* 运行 = Alt + F2   
\* 显示桌面 = Ctrl + Alt + d   
\* 最小化当前窗口 = Alt + F9   
\* 最大化当前窗口 = Alt + F10   
\* 关闭当前窗口 = Alt + F4

## VI/VIM的使用

https://www.marquette.edu/mathematical-and-statistical-sciences/basic-vi-editor-commands.php

### 打开或创建文件和保存

|  |  |
| --- | --- |
| **Command** | **Effect** |
| ***vi filename*** | edit *filename* starting at line 1 |
| ***vi +n filename*** | edit *filename* beginning at line n |
| ***vi +filename*** | edit *filename* beginning at the last line |
| ***vi -r filename*** | recover *filename* after a system crash |
| ***vi +/patter filename*** | edit *filename* starting at the first line containing **pattern** |

|  |  |
| --- | --- |
| **Command** | **Effect** |
| **:w** | writes the contents of the work buffer to the file |
| **:q** | quit |
| **:q!** | quit without saving changes |
| **ZZ** | save and quit |
| **:wq** | save and quit |
| **:w *filename*** | saves to *filename* (allows you to change the name of the file) |

### 插入内容

|  |  |
| --- | --- |
| **Command** | **Insert Text** |
| **i** | before cursor |
| **a** | after cursor |
| **A** | at the end of the line |
| **o** | open a line below the current line |
| **O** | open a line above the current line |
| **r** | replace the current character |
| **R** | replace characters until <ESC>, overwrite |

### 移动光标

|  |  |
| --- | --- |
| **Command** | **Moves the cursor** |
| **SPACE, l (el), or right arrow** | space to the right |
| **h or left arrow** | space to the left |
| **j or down arrow** | down one line |
| **k or up arrow** | up one line |
| **w** | word to the right |
| **b** | word to the left |
| **$** | end of the line |
| **0**(zero) | beginning of the line |
| **e** | end of the word to the right |
| **-** | beginning of previous line |
| **)** | end of the sentence |
| **(** | beginning of the sentence |
| **}** | end of paragraph |
| **{** | beginning of paragraph |

### 删除

|  |  |
| --- | --- |
| **Command** | **Action** |
| **d0** | delete to beginning of line |
| **dw** | delete to end of word |
| **d3w** | delete to end of third word |
| **db** | delete to beginning of word |
| **dW** | delete to end of blank delimited word |
| **dB** | delete to beginning of blank delimited word |
| **dd** | delete current line |
| **5dd** | delete 5 lines starting with the current line |
| **dL** | delete through the last line on the screen |
| **dH** | delete through the first line on the screen |
| **d)** | delete through the end of the sentence |
| **d(** | delete through the beginning of the sentence |
| **x** | delete the current character |
| **nx** | delete the number of characters specified by **n.** |
| **nX** | delete **n** characters before the current character |

## BASH的使用

<https://linuxconfig.org/bash-scripting-tutorial-for-beginners>

## 工具软件

### Putty

·打开SSHD服务。具体方法：

a) “sudo apt-get install openssh-server”。

b) 输入指令“/etc/init.d/sshd start”开启SSHD服务。（Optional，反正这步我没用到）

## 文件基本属性

$ ls -l



## 用户和用户组

$ cat /etc/passwd

$ less /etc/passwd

$ more /etc/passwd



## 磁盘管理

列出所有分区信息 # fdisk -l

## 翻译

Usr =

## 编程

https://code.visualstudio.com/docs/remote/remote-overview

# 数据库

数据库Sqlite，Mysql，SqlServer

## Mysql

## SqlServer

## Sqlite

# 胶片打印机

胶片打印机的软件要点。如果其他要点的内容放入其他章节，比如数据库连接相关的放入到数据库中。

## TUX

Data.\_connectString = "server=127.0.0.1;uid=dbadmin; pwd=dbadmin1;database=mim; CharSet=utf8";

Tux系统的数据库连接字符串。只能通过putty方式连接。

## WINDOWS

### 6950

# 协议

## SSH

**Secure Shell**（安全外壳协议，简称**SSH**）是一种加密的[网络传输协议](https://zh.wikipedia.org/wiki/%E7%BD%91%E7%BB%9C%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE" \o "网络传输协议)，可在不安全的网络中为网络服务提供安全的传输环境[[1]](https://zh.wikipedia.org/wiki/Secure_Shell" \l "cite_note-rfc4251-1)。SSH通过在网络中创建[安全隧道](https://zh.wikipedia.org/w/index.php?title=%E5%AE%89%E5%85%A8%E9%9A%A7%E9%81%93&action=edit&redlink=1)来实现SSH客户端与服务器之间的连接[[2]](https://zh.wikipedia.org/wiki/Secure_Shell" \l "cite_note-rfc4252-2)。SSH最常见的用途是远程登录系统，人们通常利用SSH来传输[命令行界面](https://zh.wikipedia.org/wiki/%E5%91%BD%E4%BB%A4%E8%A1%8C%E7%95%8C%E9%9D%A2" \o "命令行界面)和远程执行命令。SSH使用频率最高的场合是[类Unix系统](https://zh.wikipedia.org/wiki/%E7%B1%BBUnix%E7%B3%BB%E7%BB%9F" \o "类Unix系统)，但是[Windows](https://zh.wikipedia.org/wiki/Windows" \o "Windows)操作系统也能有限度地使用SSH。2015年，微软宣布将在未来的操作系统中提供原生SSH协议支持[[3]](https://zh.wikipedia.org/wiki/Secure_Shell" \l "cite_note-3)，[Windows](https://zh.wikipedia.org/wiki/Windows" \o "Windows) 10 1803版本已提供[OpenSSH](https://zh.wikipedia.org/wiki/OpenSSH" \o "OpenSSH)工具[[4]](https://zh.wikipedia.org/wiki/Secure_Shell" \l "cite_note-4)。

# 正则表达式

# 高等数学

# 网络编程

# 概率论与数理统计

# C#编程

# C++编程

# 数据结构

# 算法

# 线性代数

不能把矩阵放在分母上。

## 行列式

# Python