

## 华东师范大学软件学院课程作业

课程名称:	年级: 2023 级本科	姓名:
作业主题:	学号: 1023510	作业日期: 2024/01/01
指导老师:	组号:	

## 目录

一 代码块	1 五 TCP 三次握手图	3
二 文内引用	1 六 UDP 协议头的格式	4
三 公式模块	1 七 多行路线走势图	4
1 解答 Colorbox . . . . .	1 八 图表	4
2 定理 Colorbox . . . . .	1 1 三线图 . . . . .	4
3 定义 Colorbox . . . . .	2 2 多行居中表格 . . . . .	4
4 命题 Colorbox . . . . .	3 3 更美观的表格 . . . . .	5
5 例子 Colorbox . . . . .	4 4 矩阵 . . . . .	5
6 公理 Colorbox . . . . .	5 5 多格表格 . . . . .	5
7 注释 Colorbox . . . . .	2 九 建模图	5
8 评论 Colorbox . . . . .	2 十 甘特图	6
9 证明 Colorbox . . . . .		
四 超链接		

## 一 代码块

```
1 import pandas as pd
2 import numpy as np
```

## 二 文内引用

*This is a reference.*

### 三 公式模块

#### 1 解答 Colorbox

解答 三 .1: 解答

$$\text{Deralive}^2 = \text{Dera}^2 + \text{live}^2.$$

(Deralive Equation)

#### 2 定理 Colorbox

定理 三 .2: Pythagorean Theorem

Deralive is a normal person.

#### 3 定义 Colorbox

定义 三 .3: Deralive

Deralive is a person.

#### 4 命题 Colorbox

命题 三 .4: Deralive is a normal person.

Deralive is a normal person.

#### 5 例子 Colorbox

Contoh 三 .5

Deralive is a normal person.

#### 6 公理 Colorbox

公理 三 .6: Axiom 1

Deralive is a normal person.

#### 7 注释 Colorbox

Annotation

This is a annotation.

#### 8 评论 Colorbox

Remark

This is a remark.

## 9 证明 Colorbox

PROVE: Deralive is alive.

$\because$  Deralive is alive.

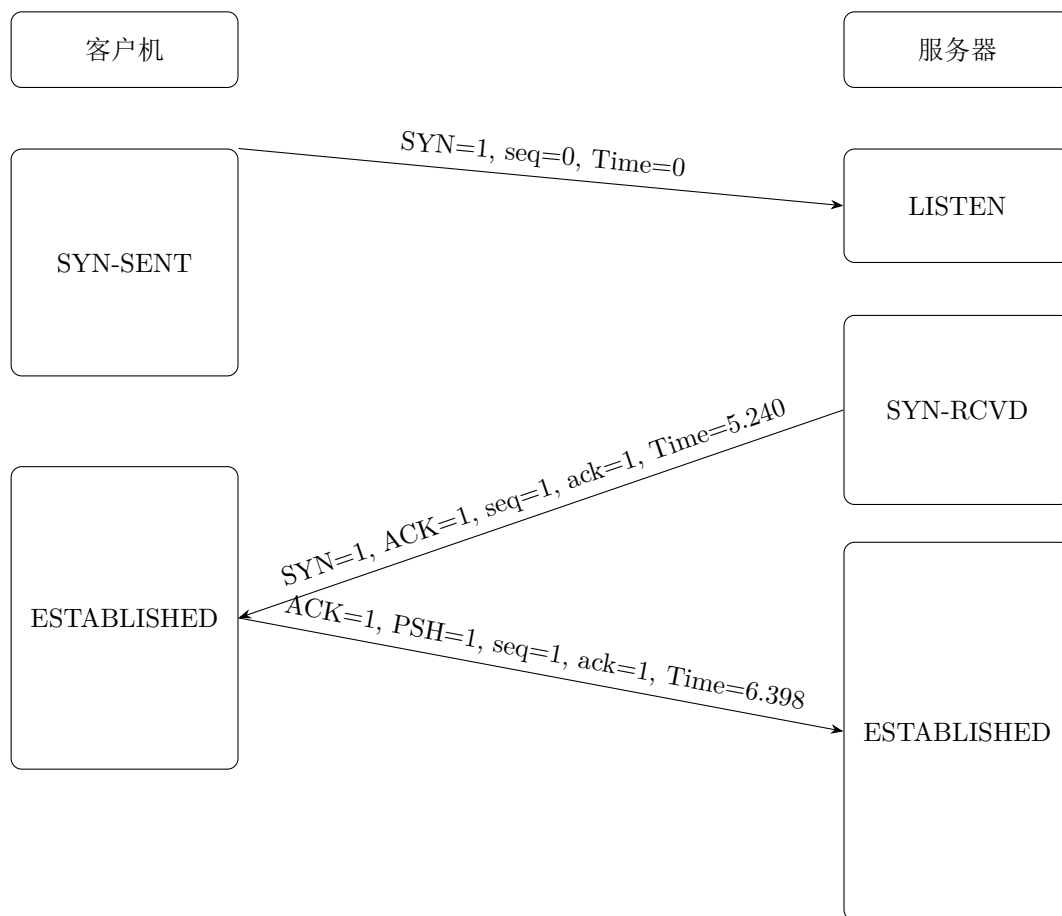
$\therefore$  Deralive is alive.

Q.E.D. ■

## 四 超链接

- 百度: [www.baidu.com](http://www.baidu.com)

## 五 TCP 三次握手图



## TikZ 节点的锚点参数

在 TikZ 中, 节点的锚点 (anchor) 定义了节点的不同参考位置, 类似于方向上的“东南西北”。这些锚点允许你灵活地连接节点或对齐它们。

基础方向: 东南西北

- north: 顶部中心

- south: 底部中心

- east: 右侧中心

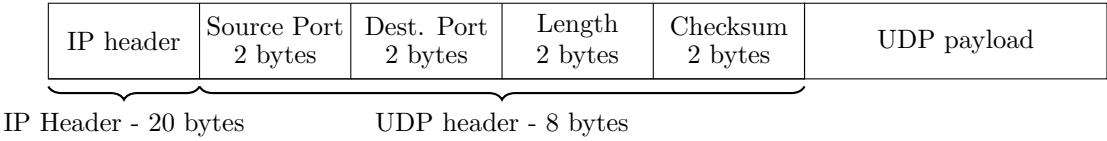
- west: 左侧中心

组合方向: 东北、西北等

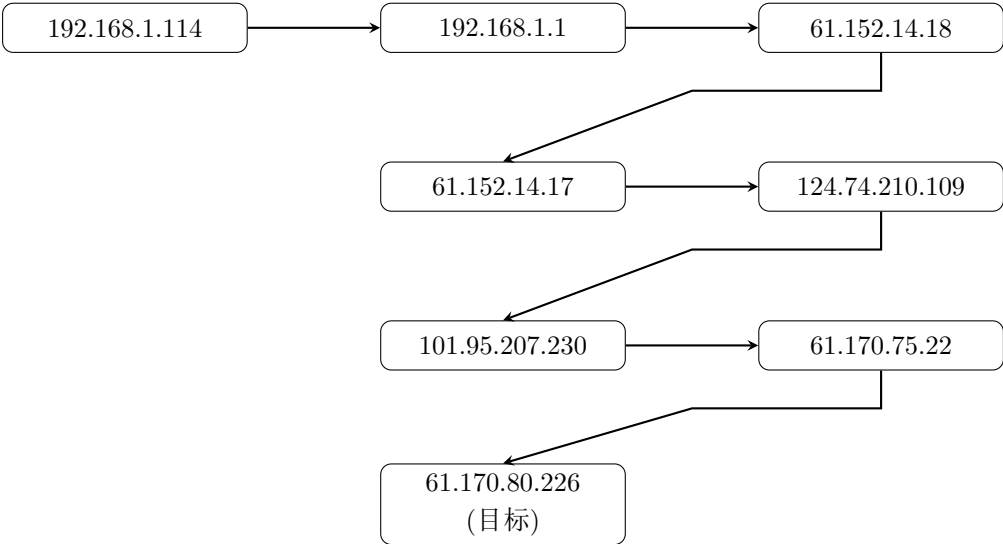
- north east: 右上角

<ul style="list-style-type: none"><li>• <b>north west:</b> 左上角</li></ul>	<b>中心锚点</b> <ul style="list-style-type: none"><li>• <b>center:</b> 节点的中心点</li></ul>	<ul style="list-style-type: none"><li>• <b>mid east:</b> 右侧中间</li><li>• <b>mid west:</b> 左侧中间</li></ul>
<ul style="list-style-type: none"><li>• <b>south east:</b> 右下角</li></ul>	<b>边缘锚点（中间位置）</b> <ul style="list-style-type: none"><li>• <b>mid:</b> 整个节点的中间</li><li>• <b>mid north:</b> 顶部中间</li><li>• <b>mid south:</b> 底部中间</li></ul>	<b>文字锚点（与文字对齐相关）</b> <ul style="list-style-type: none"><li>• <b>base:</b> 与基线对齐</li><li>• <b>text:</b> 与文本的边界对齐</li><li>• <b>text north, text south</b> 等: 文本边缘的方向</li></ul>
<ul style="list-style-type: none"><li>• <b>south west:</b> 左下角</li></ul>		

六 UDP 协议头的格式



七 多行路线走势图



八 图表

1 三线图

<i>F</i>	<i>SF</i>	<i>R</i>	<i>SV</i>	<i>M</i>	<i>T</i>
6.8	8.2	7.8	6.8	7.9	
8.8	8.9	9.2	9.0	9.3	
5.6	5.9	6.0	9.0	5.3	
4.6	7.8	3.8	5.1	4.5	

2 多行居中表格

表 1: 作业三的数据

编号	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$\rho_y$	$y_1$	$y_2$
1	8.6	9.1	9.2	8.8	8.9	0.01	8.927694	8.927671
	8.6	9.1	9.2	8.8	8.9	0.55		9.114381
2	6.8	7.9	5.9	6.6	6.1	0.01	6.614913	6.614570
	6.8	7.9	5.9	6.6	6.1	0.55		6.596355
3	9.1	9.9	8.9	8.8	7.8	0.01	8.859329	8.859071
	9.1	9.9	8.9	8.8	7.8	0.55		8.845068
4	3.5	4.2	5.6	4.9	5.2	0.01	4.695127	4.694559
	3.5	4.2	5.6	4.9	5.2	0.55		4.663374

3  更美观的表格

Index	1	2	3	4	5	6	7	8	9	Value	Level
10	7.6628	5.4788	6.3639	7.0000	7.8744	3.7417	10.0000	9.0000	4.5721	6.4071	II
11	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	10.0000	7.2713	II
12	8.3255	3.2854	8.3666	9.0000	9.6583	8.3666	10.0000	9.0000	10.0000	7.6010	II
13	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	10.0000	7.2713	II
14	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	10.0000	7.2713	II
15	8.3255	3.2854	8.0722	9.0000	9.6583	8.3666	10.0000	9.0000	10.0000	7.5467	II
16	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	10.0000	7.2713	II
17	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	9.6481	7.2479	II

表 2: 结果表

4  矩阵

$$\mathbf{A} = \begin{bmatrix} 1 & 1/2 & 3 & 2 & 1/2 \\ 2 & 1 & 2 & 3 & 2 \\ 1/3 & 1/2 & 1 & 2 & 1/3 \\ 1/2 & 1/3 & 1/2 & 1 & 2 \\ 2 & 1/2 & 3 & 1/2 & 1 \end{bmatrix}$$

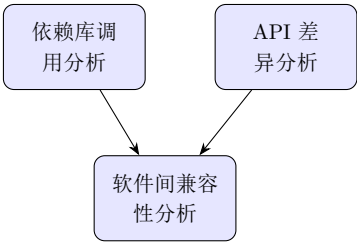
5  多格表格

进程	Allocation				Max			
	A	B	C	D	A	B	C	D
$P_0$	3	0	1	4	5	1	1	7
$P_1$	2	2	1	0	3	2	1	1
$P_2$	3	1	2	1	3	3	2	1
$P_3$	0	5	1	0	4	6	1	2
$P_4$	4	2	1	2	6	3	2	5

表 3: 系统资源快照

九  建模图

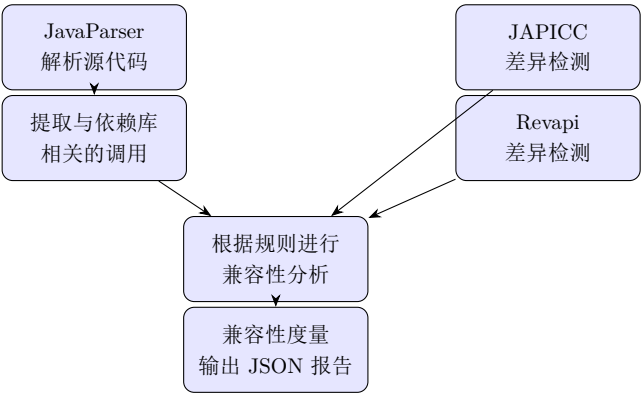
流程图：



流程分析：

主软件通过调用依赖库的 API，检测当前版本与默认版本之间的差异，评估这些差异是否引发兼容性问题。

当直接依赖软件的某个抽象类中添加了新的抽象方法时，只有主软件继承了该类才会出现兼容性问题。



流程分析：

- JavaParser 解析主软件源码，提取对依赖库的调用。
- 通过工具检测新旧版本依赖库之间的 API 差异。
- 基于兼容性规则分析差异的影响，度量兼容性问题，并输出 JSON 格式的结果报告。

注意事项：当一个 API 调用受到多个 API 差异的影响时，受到影响的严重程度取高值。

High > Medium > Low > No

十 甘特图

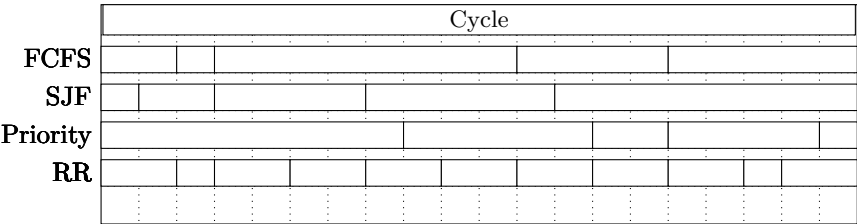


图 1: Gantt Chart