**软件版本号命名规则解读**

**[name].x.y.z-[state]**

* **name** 为可选段，一般为v，表示Version;
* **x.y.z** 为各版本的序号，一般都会遵循[语义化版本 2.0.0 | Semantic Versioning](https://link.zhihu.com/?target=https%3A//semver.org/lang/zh-CN/" \t "_blank)  
  实际上基于此规范，name段不会出现；
* **state** 可选段，用来表示当前软件的状态。例如：b 表示bate版，即测试版。

**序号段核心规则**

序号段就是 x.y.z 这一段，x、y、z为非负整数。

|  |  |
| --- | --- |
| 序号 | 说明 |
| x | 主版本号(major)无法向下兼容时，需要递增 |
| y | 次版本号(minor)新增新的特性时，需要递增 |
| z | 修订版本号(patch)修复问题时，需要递增 |

**0.y.z** 表示当前软件处于研发阶段，软件并不稳定

**1.0.0** 表示当前软件为初始的稳定版，后续的更新都基于此版本

对于系统基础的软件，一般在版本号中会标明适用的操作系统名称，例如上面截图中的Golang的版本命名。

**版本状态段规则**

|  |  |  |
| --- | --- | --- |
| 标识 | 说明 | 含义 |
| α 或 a | alpha 版 | 内测版，Bug多 |
| β 或 b | beta 版 | 公测版，有缺陷 |
| γ 或 g | Gamma 版 | 成熟测试版，接近发行版 |
| rc | ReleaseCandidate 版 | 预发布版，有时会进一步细分：rc1、rc2 |
| Demo | 演示版 | 演示用，不做升级 |
| SP | SP1 | service pack,升级包 |
| Trial | 试用版 | 试用版 |
| Unregistered | 未注册版 | 没有注册的版本，功能上有限制，这个大家懂的 |
| Lite | 精简版 | 只包含核心功能 |
| enhance | 增强版 | 增强版 |
| free | 免费版 | 自由使用的版本 |
| release | 发行版 | 有时间限制 |
| upgrade | 升级版 | 有功能增强或者修复了Bug |
| Retail | 零售版 | 单独发售 |
| Cardware | 共享版 | 使用公用许可证 |

实际上大部分软件版本号命名都遵守上述规则。

**总结**

软件版本号格式通常为：**x.y.z**

* **x** 主版本号，不向下兼容时递增；
* **y** 次版本号，向下兼容，有新增特性时递增；
* **z** 修订版本号，向下兼容，修复问题时递增；
* **a、b、rc** 分别表示当前软件的发行状态。

