# 概述

Supervisor是一个用来管理和控制进程的工具，通常用于在Linux/Unix系统上以非守护进程方式运行的程序。通过Supervisor，可以方便地启动、停止、重启服务，并实现服务的自动化管理（例如崩溃自动重启）。

# 原理

Supervisor的核心是一个客户端/服务器系统：

1、服务端 (supervisord)：

- supervisord是Supervisor的主进程。

- 它根据配置文件管理子进程的生命周期。

- 支持后台运行，也可以作为前台运行的调试工具。

2、客户端 (supervisorctl)：

- supervisorctl是用于与supervisord交互的命令行工具。

- 通过它可以查看进程状态、手动启动/停止/重启子进程等。

Supervisor通过配置文件定义需要管理的进程，并通过子进程启动和监控这些服务。如果某个进程意外退出，Supervisor可以按照配置规则重新启动它。

# 使用

## 安装 Supervisor

在 CentOS/RHEL 系列

sudo yum install epel-release

sudo yum install supervisor

## Supervisor的配置

默认的Supervisor配置文件路径为/etc/supervisor/supervisord.conf。在pip安装的情况下，可以生成默认配置：

echo\_supervisord\_conf > /etc/supervisord.conf

## 配置文件结构

典型的配置文件包含以下部分：

1、[supervisord]：主进程配置

管理supervisord的运行参数：

[supervisord]

logfile=/var/log/supervisord.log ; 主进程日志文件

pidfile=/var/run/supervisord.pid ; 主进程 PID 文件

childlogdir=/var/log/supervisor ; 子进程日志目录

2、[program:x]：子进程配置

每个子进程需要一个独立的配置块：

[program:example]

command=/path/to/your/script.sh ; 要运行的命令

autostart=true ; 是否随 supervisord 启动

autorestart=true ; 是否自动重启（always/unexpected/false）

startsecs=3 ; 程序启动后正常运行的时间（秒）

stderr\_logfile=/var/log/example.err ; 错误日志文件路径

stdout\_logfile=/var/log/example.log ; 标准输出日志路径

3、[include]：引入其他配置

允许将配置拆分成多个文件：

[include]

files = /etc/supervisor/conf.d/\*.conf

在/etc/supervisor/conf.d/中为每个服务创建单独的配置文件，例如：

/etc/supervisor/conf.d/example.conf

## Supervisor 的常用命令

1、启动Supervisor服务

sudo systemctl start supervisor

2、检查Supervisor服务状态

sudo systemctl status supervisor

3、通过supervisorctl管理进程

- 查看所有进程状态：

supervisorctl status

- 启动/停止/重启某个进程：

supervisorctl start example

supervisorctl stop example

supervisorctl restart example

- 重载配置文件（并重启新添加的进程）：

supervisorctl reread

supervisorctl update

4、调试运行supervisord

直接前台运行，查看实时输出：

supervisord -n

## 自动重启和守护特性

- autorestart参数：确保进程意外退出后自动重启：

autorestart=unexpected

- startretries参数：设置启动失败时的重试次数：

startretries=3

## 日志管理

- 主进程日志：

记录在supervisord.conf的logfile中。

- 子进程日志：

通过stdout\_logfile和stderr\_logfile配置分别记录标准输出和错误输出。

日志文件应定期清理，避免占用大量磁盘空间。

示例：管理一个 Flask 应用

创建配置文件 `/etc/supervisor/conf.d/flask\_app.conf`：

[program:flask\_app]

command=python /path/to/app.py

directory=/path/to/

autostart=true

autorestart=true

stderr\_logfile=/var/log/flask\_app.err.log

stdout\_logfile=/var/log/flask\_app.out.log

启动并检查状态：

sudo supervisorctl reread

sudo supervisorctl update

sudo supervisorctl status