CentOS 6版本的系统服务是/etc/init.d启动脚本的方式，CentOS 7采用强大的systemctl来管理系统服务，大幅提高了系统服务的运行效率，但是服务的配置和以前版本完全不同，这是很大的进步，systemctl太简单易用了。

CentOS7添加自定义系统服务的步骤如下：

1）编写自定义的系统服务脚本文件；

2）用systemctl命令把自定义的系统服务设置为开机/关机自启动/停止。

本文以Oracle数据库为例子来介绍添加自定义系统服务的知识。假设ORACLE\_HOME环境变量的值是/oracle/home，各位根据自己的实际情况调整脚本的内容，把文中/oracle/home替换成您ORACLE\_HOME的值。

# 一、编写Oracle数据库启动/重启/关闭的脚本

## 1、启动Oracle数据库的shell脚本

启动Oracle数据库的脚本为/oracle/home/bin/dbstart，内容如下：

sqlplus / as sysdba <<EOF

startup;

EOF

修改脚本的权限为可执行。

chmod +x /oracle/home/bin/dbstart

## 2、重启Oracle数据库的shell脚本

启动Oracle数据库的脚本为/oracle/home/bin/dbrestart，内容如下：

sqlplus / as sysdba <<EOF

shutdown immediate;

startup;

EOF

修改脚本的权限为可执行。

chmod +x /oracle/home/bin/dbrestart

## 3、关闭Oracle数据库的shell脚本

启动Oracle数据库的脚本为/oracle/home/bin/dbshut，内容如下：

sqlplus / as sysdba <<EOF

shutdown immediate;

EOF

修改脚本的权限为可执行。

chmod +x /oracle/home/bin/dbshut

# 二、编写自定义服务的配置文件

系统服务的启动/重启/停止由它的配置文件决定，本文把Oracle数据库的系统服务命名为oracle.service。

创建服务配置文件/usr/lib/systemd/system/oracle.service，内容如下：

[Unit]

Description=Oracle RDBMS

After=network.target

[Service]

Type=simple

ExecStart=/usr/bin/su - oracle -c "/oracle/home/bin/dbstart >> /tmp/oracle.log"

ExecReload=/usr/bin/su - oracle -c "/oracle/home/bin/dbrestart >> /tmp/oracle.log"

ExecStop=/usr/bin/su - oracle -c "/oracle/home/bin/dbshut >> /tmp/oracle.log"

RemainAfterExit=yes

[Install]

WantedBy=multi-user.target

接下来介绍服务配置文件各部分的含义。

## 1、Unit部分

Unit部分是启动顺序与依赖关系。

[Unit]

Description=Oracle RDBMS

After=network.target

Description字段：给出当前服务的简单描述。

Documentation字段：给出文档位置。

After字段：表示本服务应该在某服务之后启动。

Before字段：表示本服务应该在某服务之前启动。

After和Before字段只涉及启动顺序，不涉及依赖关系。设置依赖关系，需要使用Wants字段和Requires字段。

Wants字段：表示本服务与某服务之间存在“依赖”系，如果被依赖的服务启动失败或停止运行，不影响本服务的继续运行。

Requires字段，表示本服务与某服务之间存在“强依赖”系，如果被依赖的服务启动失败或停止运行，本服务也必须退出。

## 2、Service部分

Service部分定义如何启动/重启/停止服务。

[Service]

Type=simple

ExecStart=/usr/bin/su - oracle -c "/oracle/home/bin/dbstart >> /tmp/oracle.log"

ExecReload=/usr/bin/su - oracle -c "/oracle/home/bin/dbrestart >> /tmp/oracle.log"

ExecStop=/usr/bin/su - oracle -c "/oracle/home/bin/dbshut >> /tmp/oracle.log"

RemainAfterExit=yes

1）启动类型（Type字段）

Type字段定义启动类型。它可以设置的值如下。

simple（默认值）：ExecStart字段启动的进程为主进程。

forking：ExecStart字段将以fork()方式启动，此时父进程将会退出，子进程将成为主进程。

oneshot：类似于simple，但只执行一次，Systemd会等它执行完，才启动其他服务。

dbus：类似于simple，但会等待D-Bus信号后启动。

notify：类似于simple，启动结束后会发出通知信号，然后Systemd再启动其他服务。

idle：类似于simple，但是要等到其他任务都执行完，才会启动该服务。

2）启动服务（ExecStart字段）

启动服务时执行的命令，可以是可执行程序、系统命令或shell脚本。

3）重启服务（ExecReload字段）

重启服务时执行的命令，可以是可执行程序、系统命令或shell脚本。

4）停止服务（ExecStop字段）

停止服务时执行的命令，可以是可执行程序、系统命令或shell脚本。

5）如果RemainAfterExit字段设为yes，表示进程退出以后，服务仍然保持执行。

6）服务配置文件还可以读取环境变量参数文件，我个人认为比较麻烦，没有必要，就不介绍了，设置程序的环境变量有很多种方法，可以在脚本中配置，也可以用“su –”的方法。

## 3、Install部分

Install部分定义如何安装这个配置文件，即怎样做到开机启动。

[Install]

WantedBy=multi-user.target

WantedBy字段：表示该服务所在的Target。

Target的含义是服务组，表示一组服务。WantedBy=multi-user.target指的是，oracle所在的Target是multi-user.target（多用户模式）。

这个设置非常重要，因为执行systemctl enable oracle.service命令时，oracle.service会被链接到/etc/systemd/system/multi-user.target.wants目录之中，实现开机启动的功能。

## 4、重启行为

Service部分还有一些字段，定义了重启行为。

1）KillMode字段

KillMode字段定义Systemd如何停止sshd服务，可以设置的值如下：

control-group（默认值）：当前控制组里面的所有子进程，都会被杀掉。

process：只杀主进程。

mixed：主进程将收到SIGTERM信号，子进程收到SIGKILL信号。

none：没有进程会被杀掉，只是执行服务的stop命令。

2）Restart字段

Restart字段定义了服务程序退出后，Systemd的重启方式，可以设置的值如下：

no（默认值）：退出后不会重启。

on-success：只有正常退出时（退出状态码为0），才会重启。

on-failure：非正常退出时（退出状态码非0），包括被信号终止和超时，才会重启。

on-abnormal：只有被信号终止和超时，才会重启。

on-abort：只有在收到没有捕捉到的信号终止时，才会重启。

on-watchdog：超时退出，才会重启。

always：不管是什么退出原因，总是重启。

3）RestartSec字段。

RestartSec字段：表示Systemd重启服务之前，需要等待的秒数。

# 三、使用自定义的服务

## 1、重新加载服务配置文件

每次修改了服务配置文件后，需要执行以下命令重新加载服务的配置文件。

systemctl daemon-reload

## 2、启动/停止/启重oracle服务

systemctl start oracle # 启动oracle服务。

systemctl restart oracle # 重启oracle服务。

systemctl stop oracle # 关闭oracle服务。

## 3、把oracle服务设置为开机/关机自启动/停止

systemctl is-enabled oracle # 查看oracle服务是否是开机自启动。

systemctl enable oracle # 把oracle服务设置为开机自启动。

## 4、查看Oracle实例启动/停止的日志

Oracle实例启动的日志在/tmp/oracle.log文件中。

注意，只有通过systemctl启动/关闭Oracle实例和监听才会写日志，手工执行脚本不写日志。