## **vn.py社区精选23 - 想用CentOS跑vn.py？看这一篇就能搞定！**

原创 用Python的交易员 VNPY  2020-06-12

收录于话题

#vn.py社区精选

25个

**原文作者：木头哥**

作为Python开发的开源项目，vn.py本身具有非常好的跨平台通用性，毕竟Python几乎可以在所有主流操作系统上运行。但对于Linux系统，官方团队只提供了对Ubuntu 18.04版本的支持（主要就是安装脚本）。本人一直用的是CentOS的服务器，折腾了几天，终于在上面把vnpy跑起来了。没有记录折腾的细节，只是记录了下正常的操作步骤，欢迎大家一起交流。以下内容全部基于CentOS 7.6版本，首先准备好一个全新安装的系统，然后跟着一步步操作即可。

**安装Python环境**

*# 在anaconda.com官网下载安装包，这里需要下载linux版本的x86\_64位的包*bash Anaconda3-2019.07-Linux-x86\_64.sh*# 如果不修改安装路径的话，按照默认设置即可。# 安装过anaconda后，需要将/root/.bashrc中的关于anaconda的部分屏蔽掉，不然后面vncserver无法正常启动。*

**安装Mate桌面**

yum groups install "X Window System" -yyum install epel-release -yyum groups install "MATE Desktop" -ysystemctl set-default graphical.target

**安装VNC Server**

yum install tigervnc-server -y  
*# 替换User为root，增加显示分辨率参数设置*sed -r -i "s/^(ExecStart.\*)<USER>(.\*%i)/\1root\2 -geometry 1920x1200 -depth 16/" /lib/systemd/system/vncserver@.servicesed -r -i "s/^(PIDFile.\*)home\/<USER>(.\*pid)/\1root\2/" /lib/systemd/system/vncserver@.service  
mv /lib/systemd/system/vncserver@.service /lib/systemd/system/vncserver@:1.servicesystemctl daemon-reloadvncpasswdsystemctl start vncserver@:1.servicesystemctl enable vncserver@:1.service  
*# 屏蔽默认桌面，启动mate桌面*sed -r -i "s@^/etc/X11/xinit/xinitrc$@# &@" /root/.vnc/xstartupecho "/usr/bin/mate-session &" >> /root/.vnc/xstartup  
*# 其它操作# 禁用selinux*sed -r -i "s/^(SELINUX=).\*/\1disabled/" /etc/selinux/config*# 关闭防火墙*systemctl stop firewalld.servicesystemctl disable firewalld.service  
reboot  
*# 如果是云服务器，需要确保开放了TCP 5901端口*

**安装VS Code**

rpm --import https:*//packages.microsoft.com/keys/microsoft.asc*sh -c 'echo -e "[code]\nname=Visual Studio Code\nbaseurl=https://packages.microsoft.com/yumrepos/vscode\nenabled=1\ngpgcheck=1\ngpgkey=https://packages.microsoft.com/keys/microsoft.asc" > /etc/yum.repos.d/vscode.repo'yum check-updateyum install code -y

**升级GCC版本**

注意GCC必须要使用9.1.0以上的版本，否则在编译vnpy的时候，会报-std=c++17相关的错误，另外GCC编译的时间很长，估计得几个小时。

yum install gcc gcc-c++ bzip2 m4 gmp-devel.x86\_64 -y  
wget https://mirrors.ustc.edu.cn/gnu/gcc/gcc-9.1.0/gcc-9.1.0.tar.gztar xvf gcc-9.1.0.tar.gzcd gcc-9.1.0/./contrib/download\_prerequisites  
cd gmp;mkdir temp;cd temp../configure --prefix=/usr/local/gmp-6.1.0make && make install  
cd ../../mpfr;mkdir temp;cd temp../configure --prefix=/usr/local/mpfr-3.1.4 --with-gmp=/usr/local/gmp-6.1.0make && make install  
cd ../../mpc;mkdir temp;cd temp../configure --prefix=/usr/local/mpc-1.0.3 --with-gmp=/usr/local/gmp-6.1.0 --with-mpfr=/usr/local/mpfr-3.1.4make && make install  
export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/mpc-1.0.3/lib:/usr/local/gmp-6.1.0/lib:/usr/local/mpfr-3.1.4/lib  
cd ../..;mkdir temp;cd temp../configure --disable-multilib --enable-languages=c,c++ --with-gmp=/usr/local/gmp-6.1.0 --with-mpfr=/usr/local/mpfr-3.1.4 --with-mpc=/usr/local/mpc-1.0.3make -j4 && make install

我的服务器是4核的，就在make后面加了-j4，大家可根据自己的情况调整，缩短编译时间。

**安装vn.py**

最后终于可以安装vn.py了，在此过程中会自动编译Linux上支持的交易接口，如CTP/OES等。

*# 切换到python环境*. ~/miniconda3/bin/activateyum install postgresql-devel\* libxkbcommon-x11 -y  
*# 下载vnpy的最新源码包并解压后执行*cd vnpybash install.sh