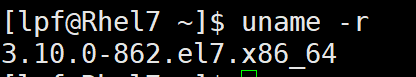
# Centos7 安装Reidis 单机版

## 需要的软件

1. redis-5.0.7.tar.gz

# 确定centos 7 操作系统的具体版本信息

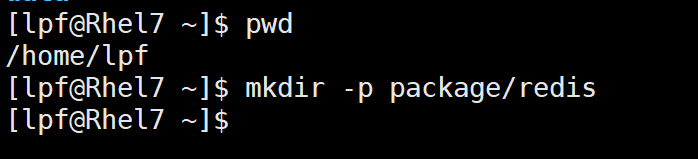
1. 通过 uname -r 命令查看linux 系统的内核版本



通过上图可以看到当前操作系统的内核版本是 3.10.0-862.el7.x86\_64。 同时也可以知道该操作系统的硬件架构是 x86\_64. 也就是我们常说的63位的操作系统。

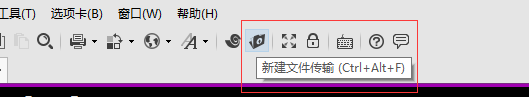
上传redis的源码包到linux系统

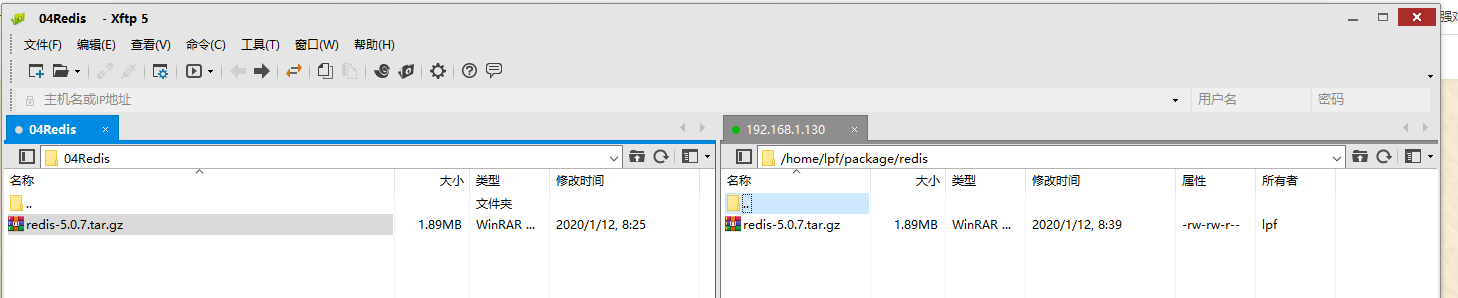
1. 在当前登录用户的home 目录下面创建一个package/redis 目录用于保存redis安装的源码



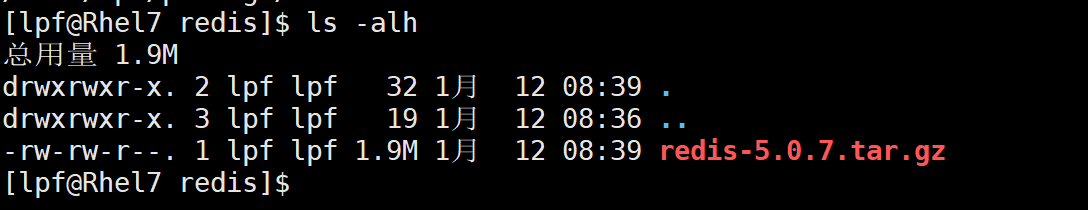
这里通过 $ > mkdir -p package/redis 命令来递归创建文件夹

1. 利用xshell 提供的文件上传插件将压缩包上传



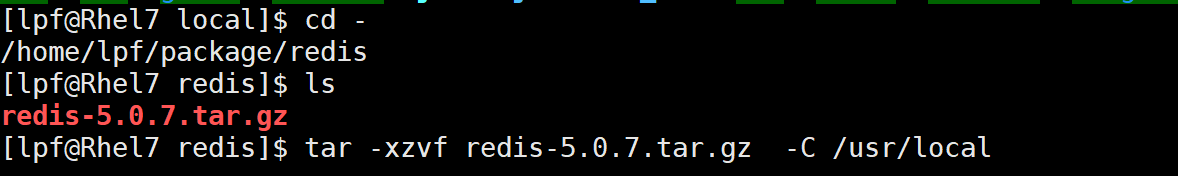


上传到linux 上面的redis压缩包的访问权限如下

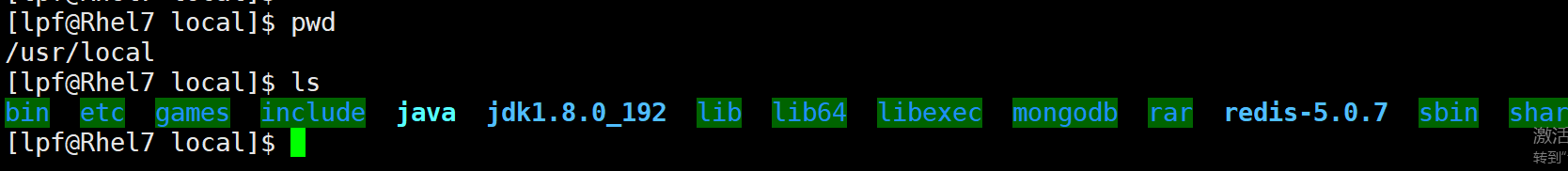


3． 解压reids-5.0.7.tar.gz 文件到 /usr/local 目录

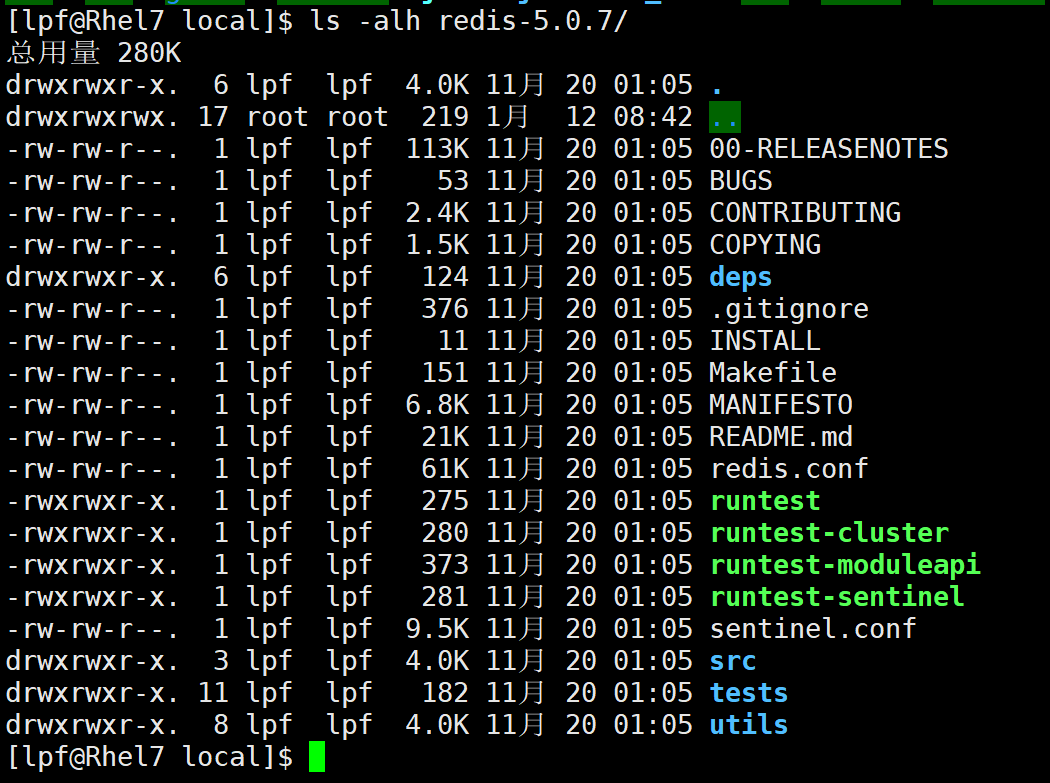
执行 tar -xzvf redis-5.0.7.tar.gz -C /usr/local 命令将 redis 的压缩包解压到 /usr/local 目录下



切换到 /user/local 目录后，可以看到解压缩后的 redis-5.0.7



解压后Redis文件的访问权限如下

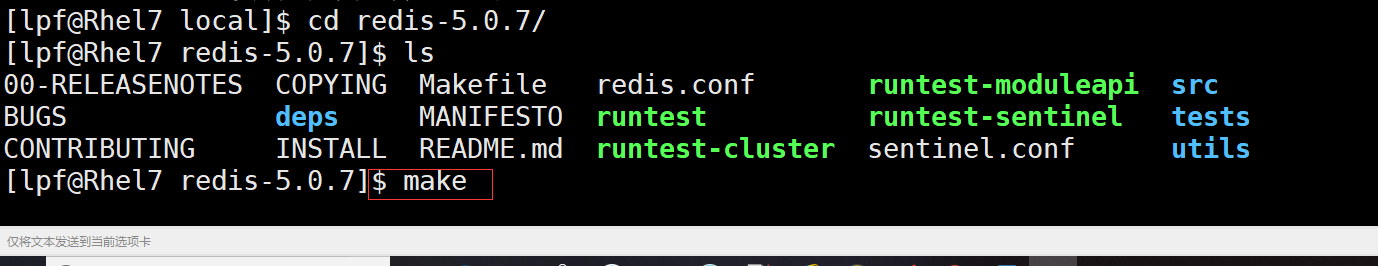


安装编译Redis

进入到 redis-5.0.7 目录 下

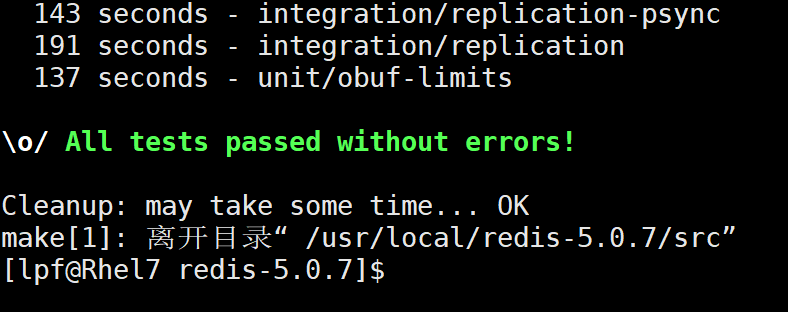
$ cd redis-5.0.7 目录

通过make 命令编译redis 源码



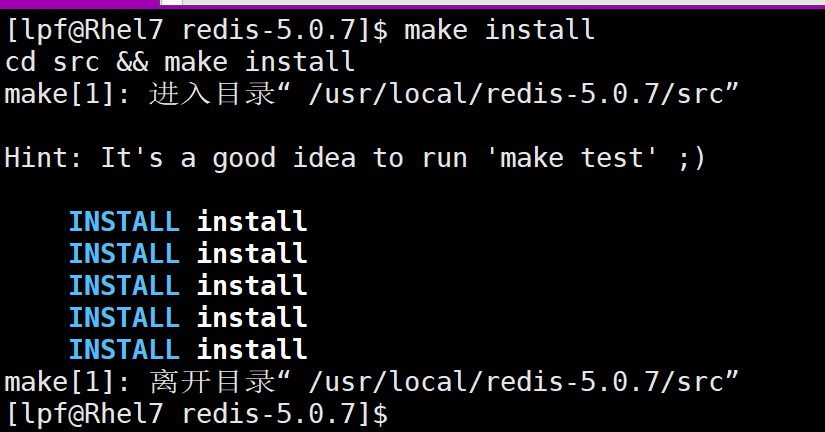
执行完make 之后，执行make test ，

执行make install 将编译好的二进制文件安装到centos7 中



出现这个说明编译成功了。

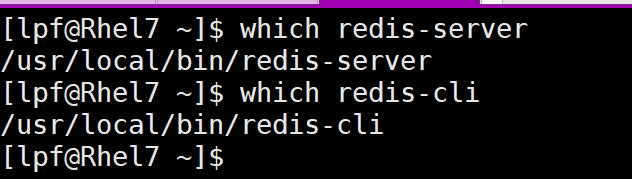
执行make install 命令安装redis 的二进制文件



出现如图所示的内容说明安装完成了。

查看redis-cli 和reids-server 的存放目录

通过which 命令查看reids-server 和redis-cli 命令存放在 /user/local/bin 目录下

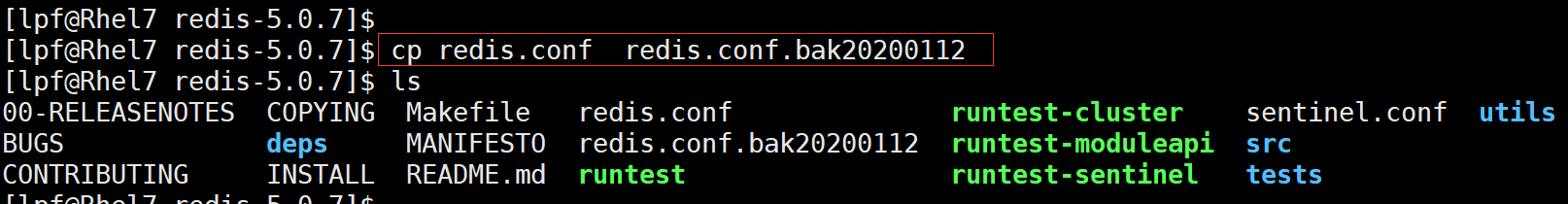


## 配置Redis

Reids 的配置文件注意是 安装目录下的redis.conf 文件

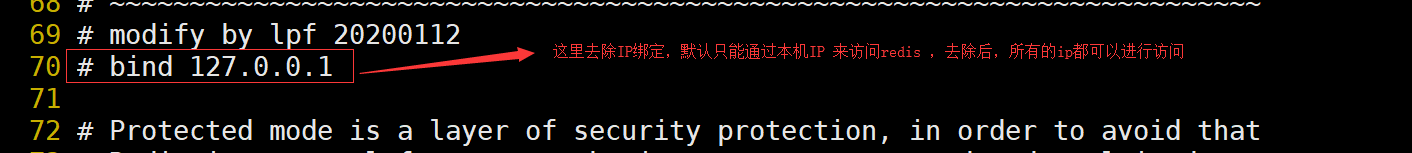
在修改redis的配置文件之前，先将redis.conf 文件备份。

执行 如下命令备份Reids 的配置文件 redis.conf 文件



## 去除本机绑定

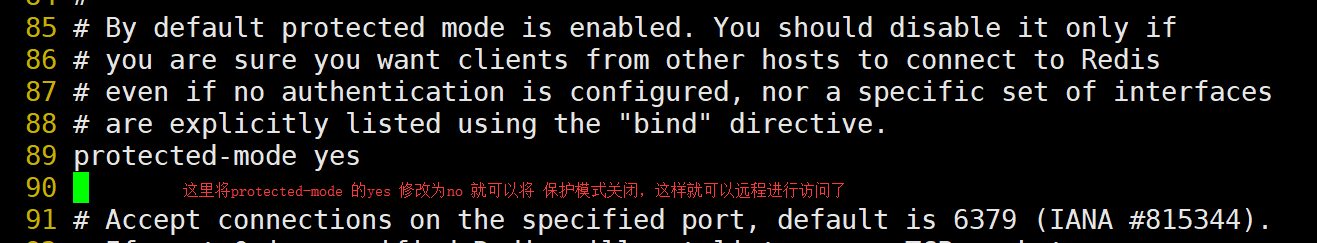
默认情况下，redis 是本机IP 绑定，只有特定的ip才能进行访问。



这里注意，原先的默认位置是滴69行， 可以通过vi快速跳转到69行。

### 2．关闭保护模式

Redis默认运行在保护模式下，redis的保护模式不允许远程访问，而且保护模式下访问redis是不需要密码的，因此，我们需要关闭保护模式。来让我们的redis可以被远程访问。



这里，注意，原先的行数是在89 行。

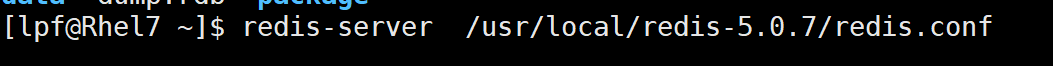
## 启动redis

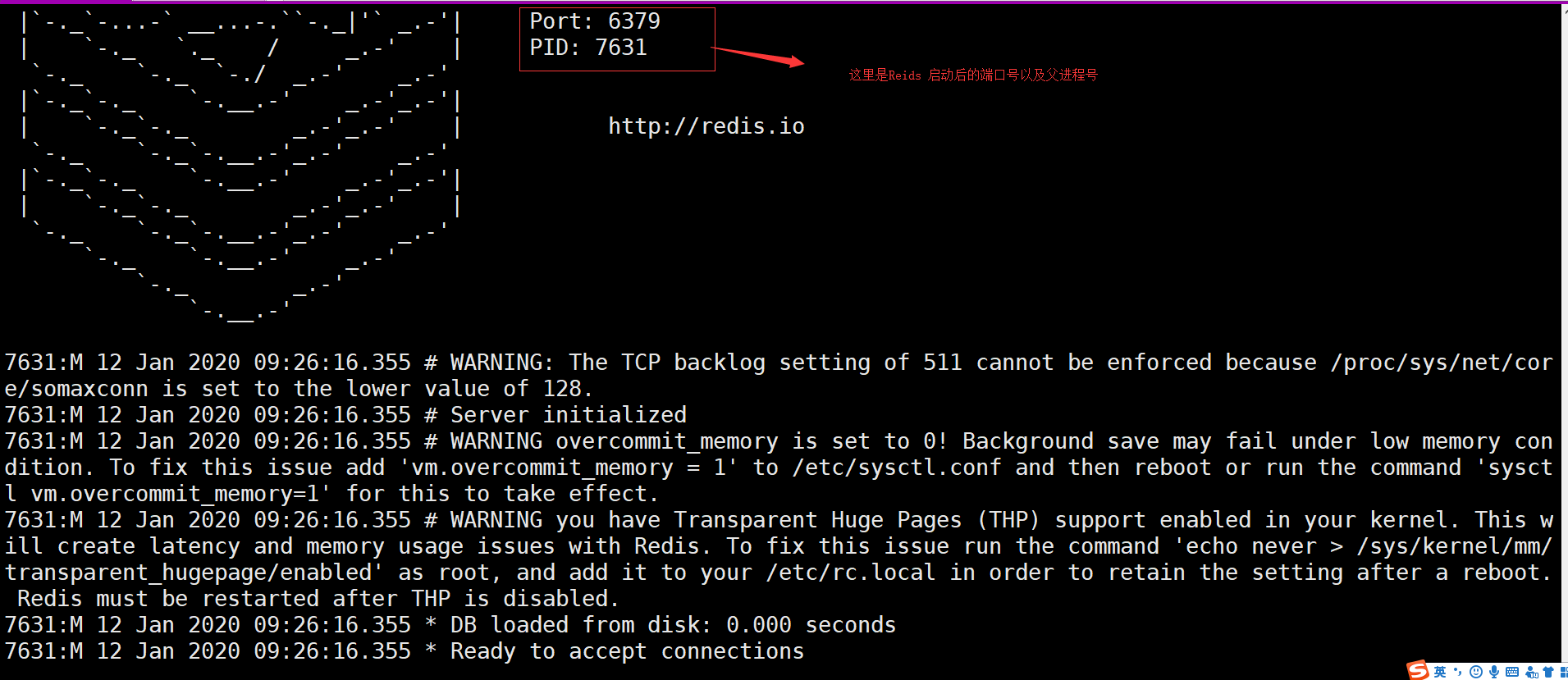
这里因为redis 安装在了 /usr/local 目录下，所有redis 的命令行会默认添加到PATH环境变量中。

### 启动redis 服务器

在任意目录执行 如下命令 安装redis.conf 中的配置信息启动redis 程序

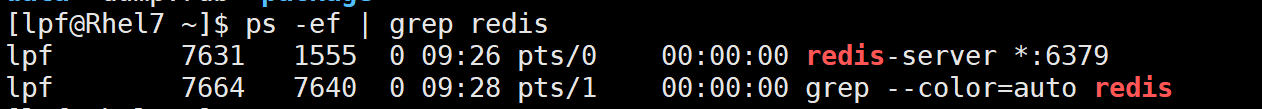
$ redis-server /usr/local/redis-5.0.7/redis.conf





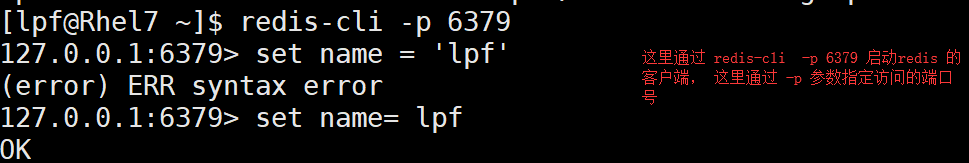
这里启动好redis 之后，切换到另一个控制台， 执行

$ ps -ef | grep redis 命令查看redis 启动后的进程。 这里可以看到reids 启动后打开了 6379 端口。



### 启动redis 客户端

通过执行 redis-cli -p 6379 命令启动redis 客户端

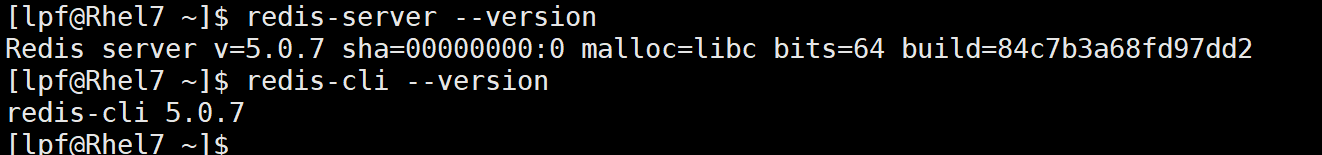


之后，就可以通过reids 的客户端来操作Reids 数据库了。

可以通过quit 或者exit 命令退出Redis 客户端

### 查看redis 服务端和客户端的版本号

通过 redis-server –-version 和 redis-cli –-version 命令查看redis服务器端和客户端的版本号

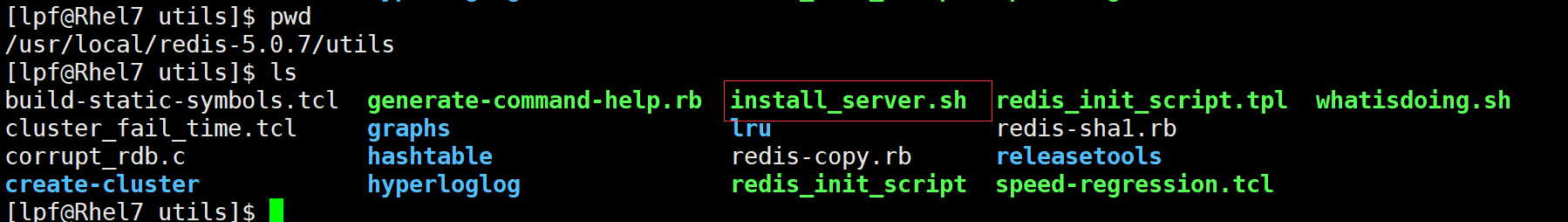


## 设置Redis 开机自启

### 升级Redis服务为一个系统服务

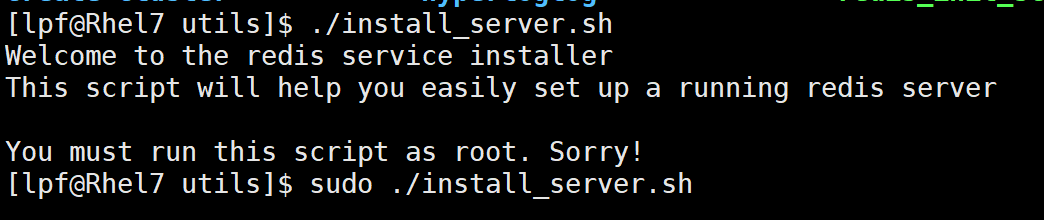
想让Redis 开机自启，首先需要将redis 升级为一个系统服务，redis 已经为我们提供好了解决方案。

在redis 的安装路径下的 utils 目录下，有如下内容，其中一个 install\_server.sh 脚本

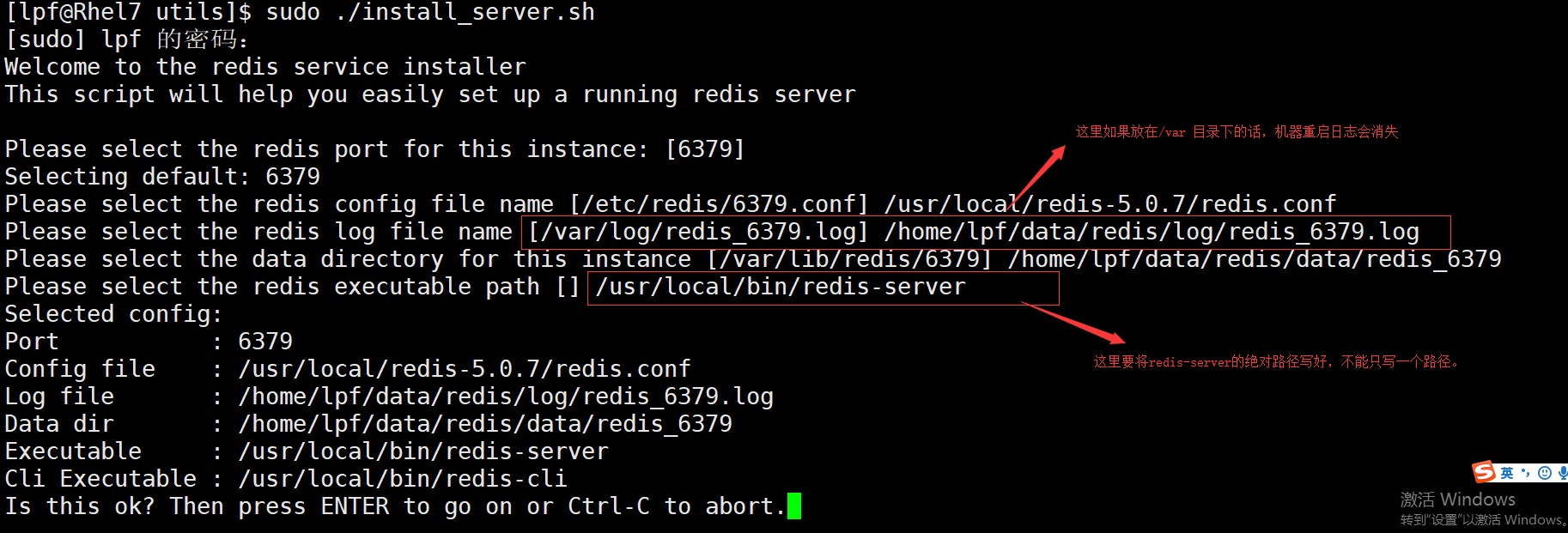


Utils/install\_server.sh 脚本会帮我们做一些配置，将reids 升级为系统服务。

利用sudo 命令执行 install\_server.sh 脚本

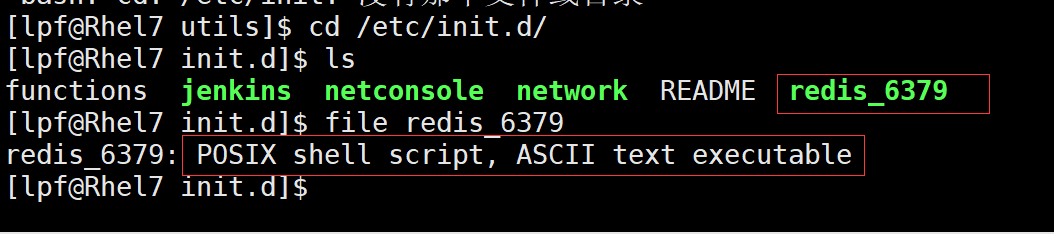


执行install\_server.sh 的过程中，需要手动输入一些参数如下



这里设置成功后会在 /etc/init.d/ 目录下有一个可执行文件

/etc/init.d/redis-6379

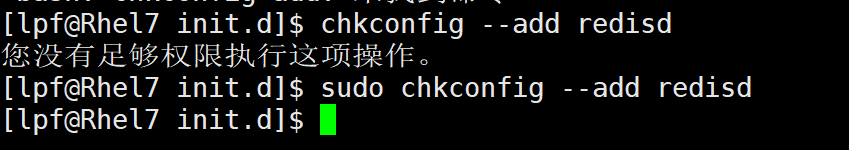


### 修改服务启动的脚本文件名称

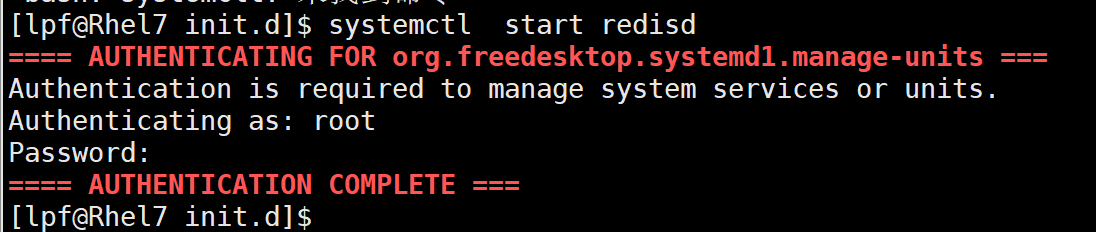
        上边的名称看起来很别扭呀，我们来修改一下，进入/etc/init.d目录。通过mv命令进行修改：mv /etc/init.d/redis\_6379 /etc/init.d/redisd 我这里将文件名修改为了redisd。

通过上面的两步，Redis 就升级为了系统服务。 接下来我们只要设置服务开机自启就可以了。 我们只要执行一条命令

$ > chkconfig –add redisd



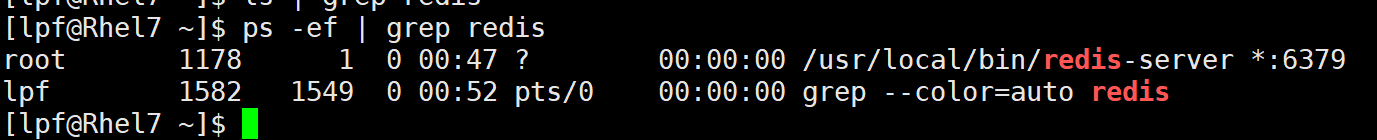
使用Linux 的系统命令systemctl



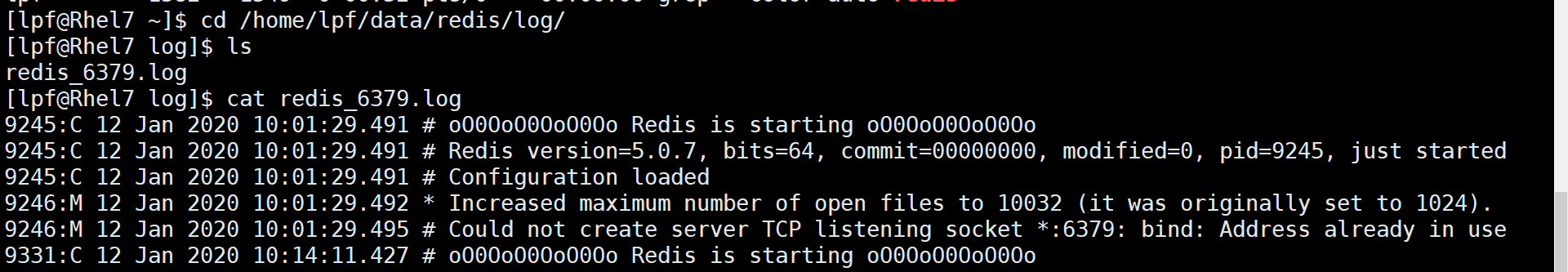
### 验证Redis 开机自启设置成功

在centOS 7 系统中执行 reboot 命令重启操作系统。

1. 重启后，执行 ps -ef | grep redis 查看redis 进程是否启动



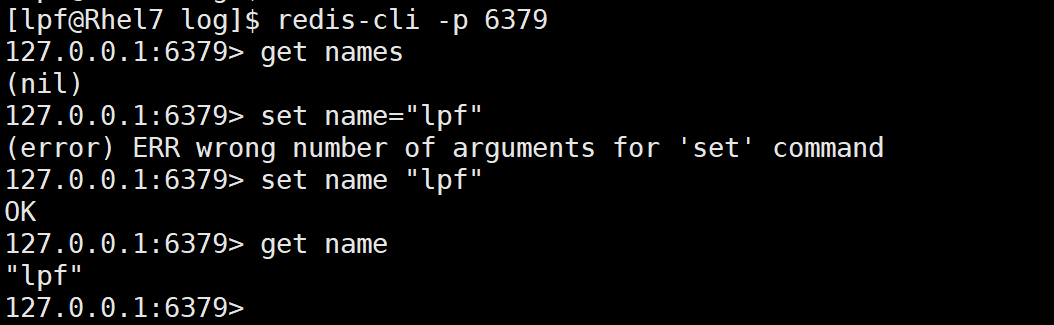
1. 根据执行的设置，查看我们设置的Redis的操作日志的日志



这里我们在指定目录找到了redis-server 启动的日志，说明生效了 。

1. 执行 redis-cli -p 6379 查看是否是安装指定的端口启动的。

其实这里可以不执行，之前通过ps 也可以看到是通过指定端口6379启动的。



这里可以通过客户端访问到Redis 服务器了。 并且可以正常的执行redis。

## Windows系统连接Linux系统上面的Redis

使用Redis Desktop Manager， 在连接之前需要让Linux 打开防火墙

>>>关闭防火墙

systemctl stop firewalld.service            #停止firewall  
systemctl disable firewalld.service        #禁止firewall开机启动

>>>开启端口

firewall-cmd --zone=public --add-port=80/tcp --permanent

命令含义

--zone #作用域  
--add-port=80/tcp #添加端口，格式为：端口/通讯协议  
--permanent #永久生效，没有此参数重启后失效

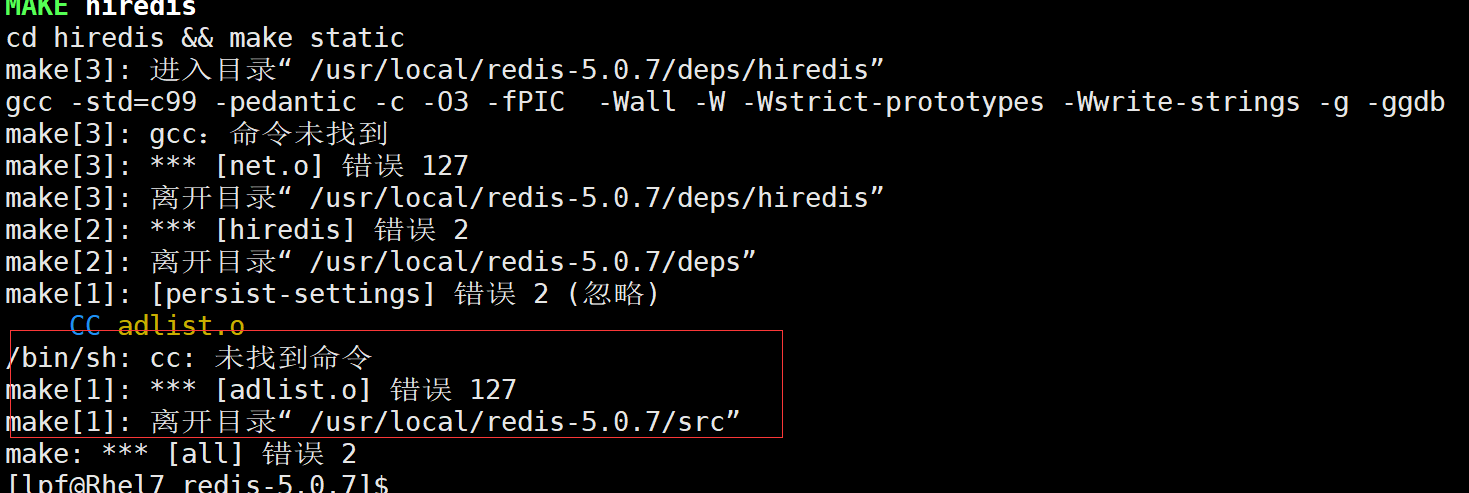
Linux系统关闭Redis 服务

》 redis-cli shutdown

或者kill redis 进程的pid

## 遇到的问题：

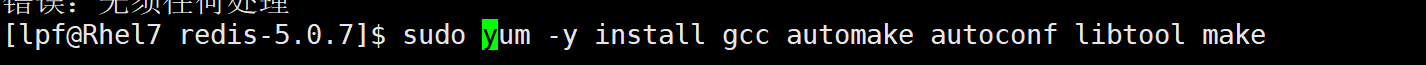
1. 缺少编译器 cc



解决办法：

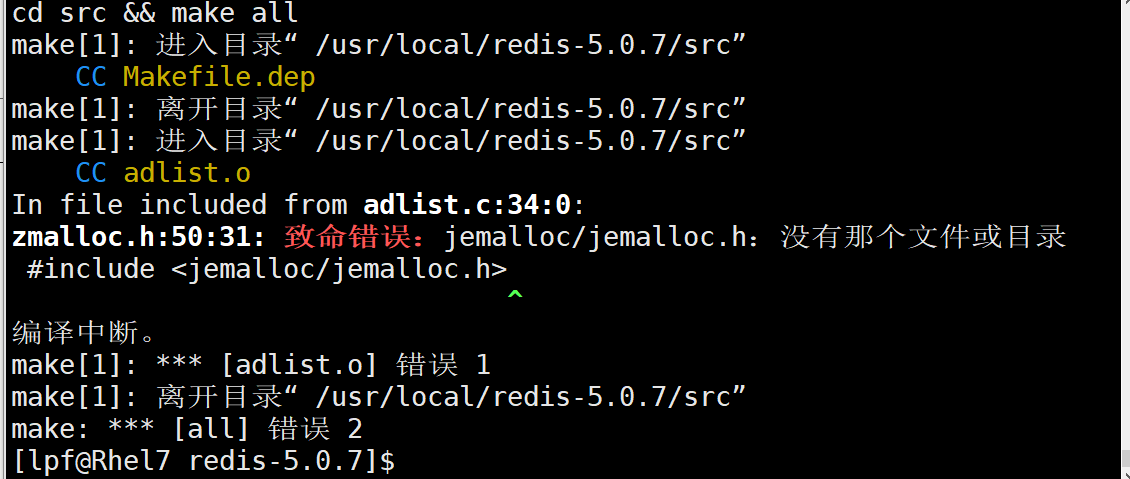
通过yum 命令安装gcc 编译器

$ > yum -y install gcc automake autoconf libtool make



注意，这样要用root 权限进行安装

1. zmalloc.h:50:31: 致命错误：jemalloc/jemalloc.h：没有那个文件或目录

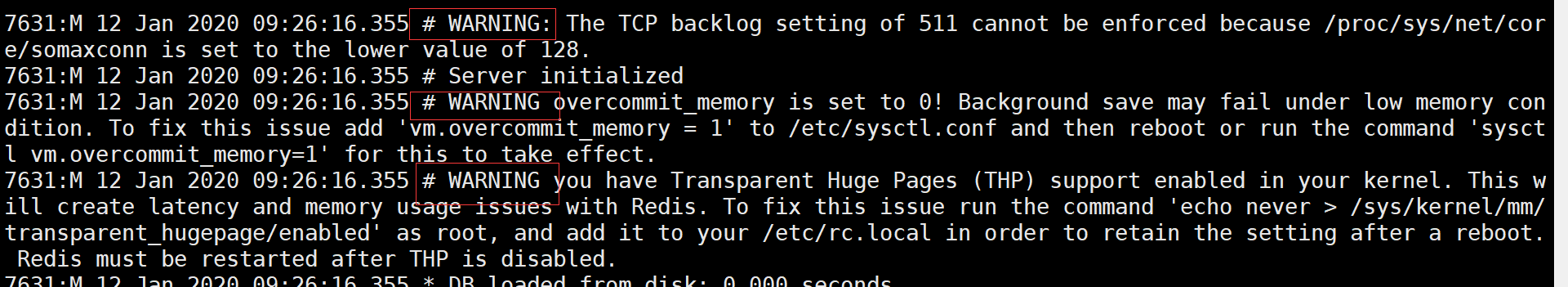


解决方案：

执行命令 ： make MALLOC=libc

$> make MALLOC=libc

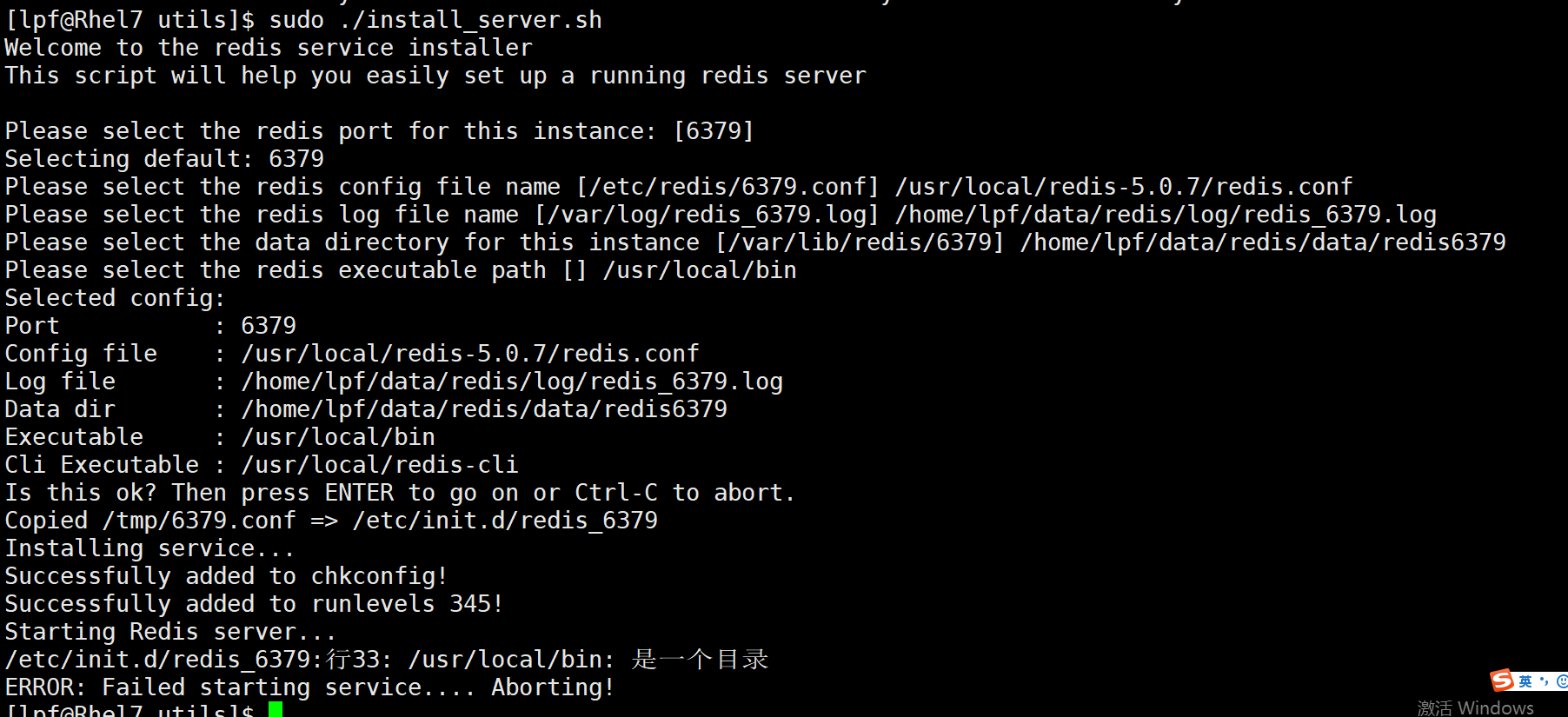
Redis服务端启动后遇到的一些警告：



|  |
| --- |
| 7631:M 12 Jan 2020 09:26:16.355 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/cor  e/somaxconn is set to the lower value of 128.7631:M 12 Jan 2020 09:26:16.355 # Server initialized  7631:M 12 Jan 2020 09:26:16.355 # WARNING overcommit\_memory is set to 0! Background save may fail under low memory con  dition. To fix this issue add 'vm.overcommit\_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit\_memory=1' for this to take effect.7631:M 12 Jan 2020 09:26:16.355 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This w  ill create latency and memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent\_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabled. |

### 执行redis的安装路径下的 /utils/install\_server.sh 脚本遇到的问题

/etc/init.d/redis\_6379:行33: /usr/local/bin: 是一个目录



只要在前面配置的时候，添加上具体执行那个命令即可。

### 缺少 tcl 8.5

(没有解决)

You need tcl 8.5 or newer in order to run the Redis test

解决办法：

wget http://downloads.sourceforge.net/tcl/tcl8.6.1-src.tar.gz

sudo tar xzvf tcl8.6.1-src.tar.gz -C /usr/local/

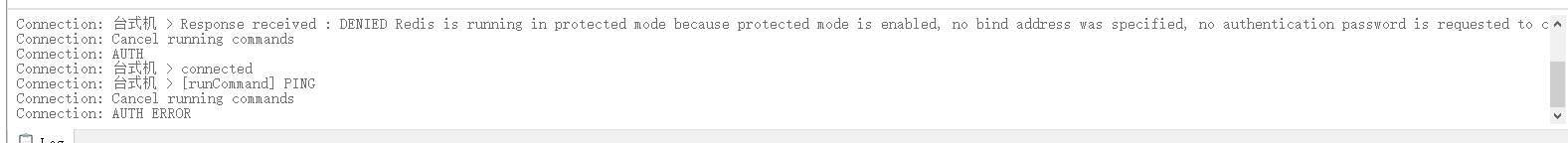
cd /usr/local/tcl8.6.1/unix/

sudo ./configure

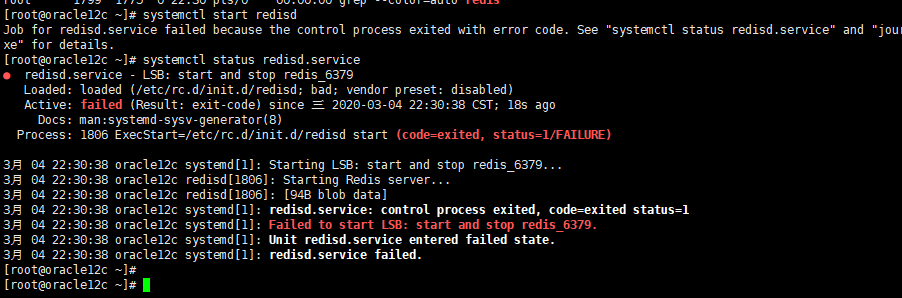
sudo make

sudo make install

### 问题三



问题四：使用redis安装包utils下面的install 注册redis为系统服务之后，修改 /etc/init.d/redis\_6379 的名字为 redisd，之后，使用systemctl start redisd 没有启动。



通过systemctl status redisd.service 之后，显示系统认为的名称依然是redis\_6379。

不知道什么原因

通过systemctl list-units 查看centos有哪些系统服务 ，显示如下

