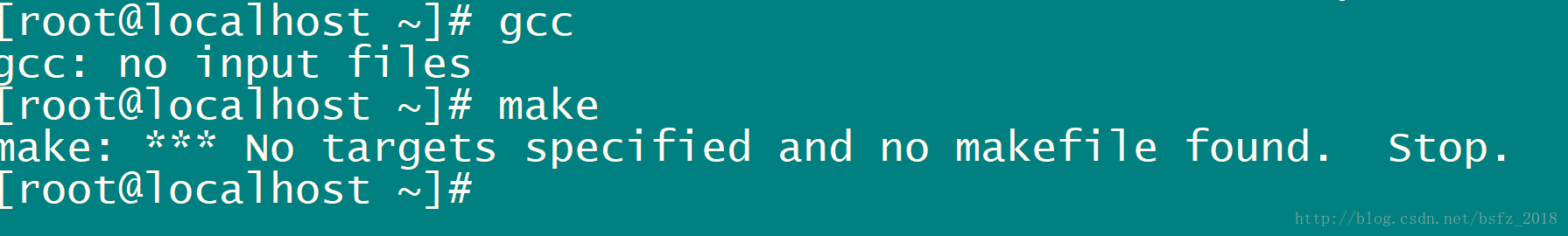
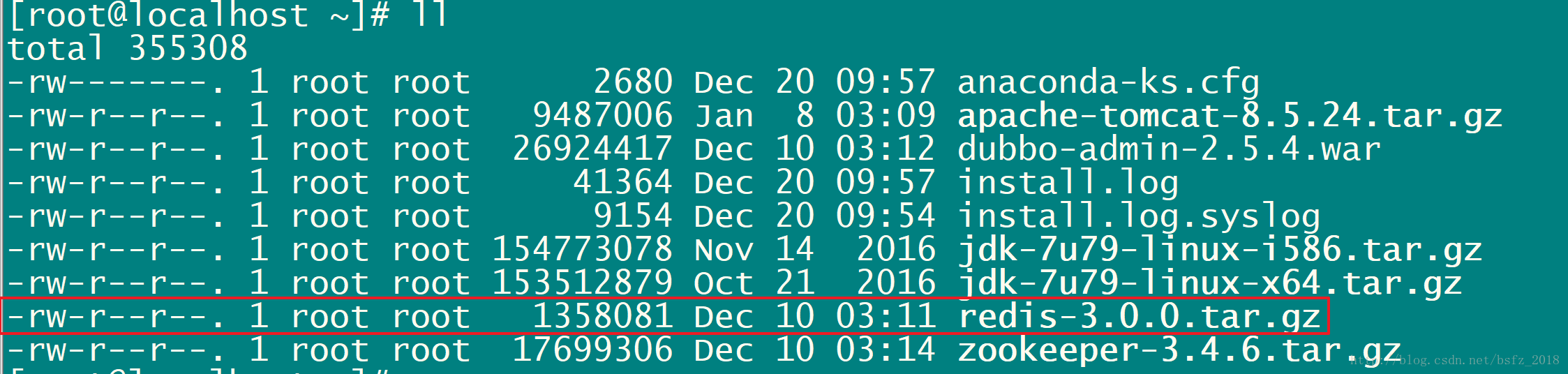
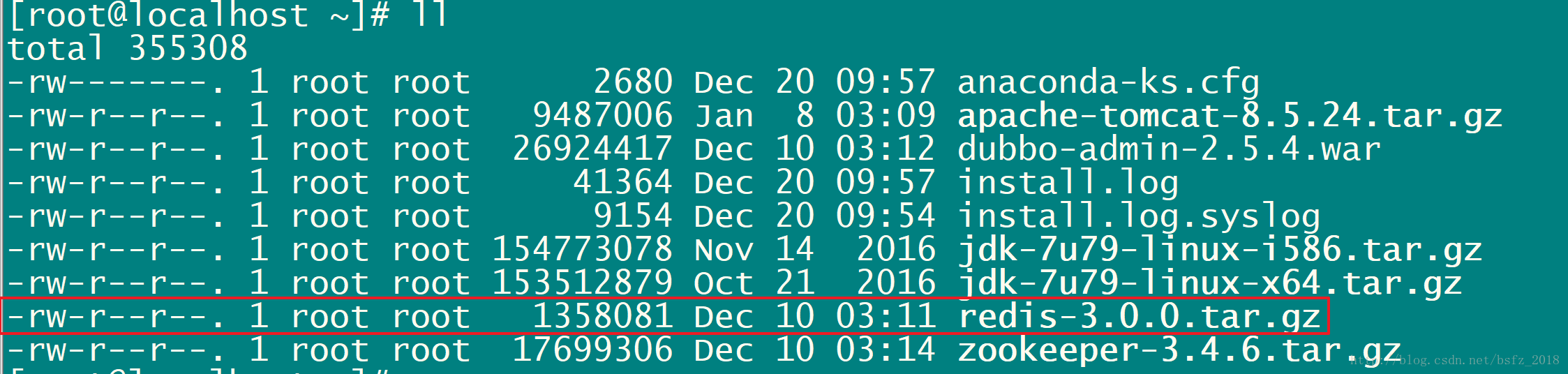
1 Redis为c语言编写，因此安装需要配置c语言编译环境gcc，查看linux是否安装了该编译环境，可以直接输入命令查看如下图所示：



上图表示已经存在该编译环境，而如果提示commond not find则意味着本机并无该环境，因此需要安装gcc环境，采用yum install gcc-c++进行安装

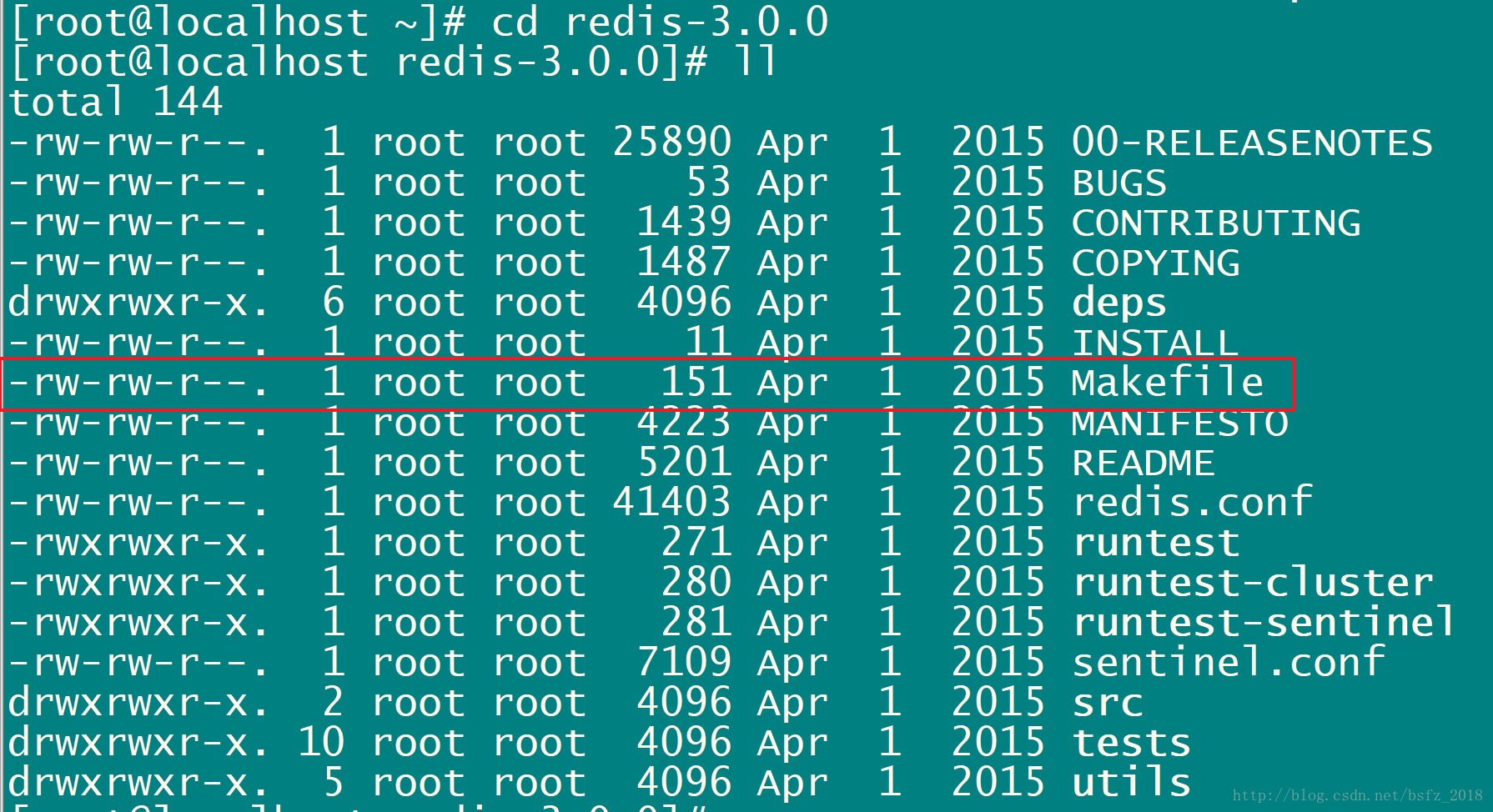


2 然后上传redis的tar包到linux上，本次上传到root用户宿主目录下

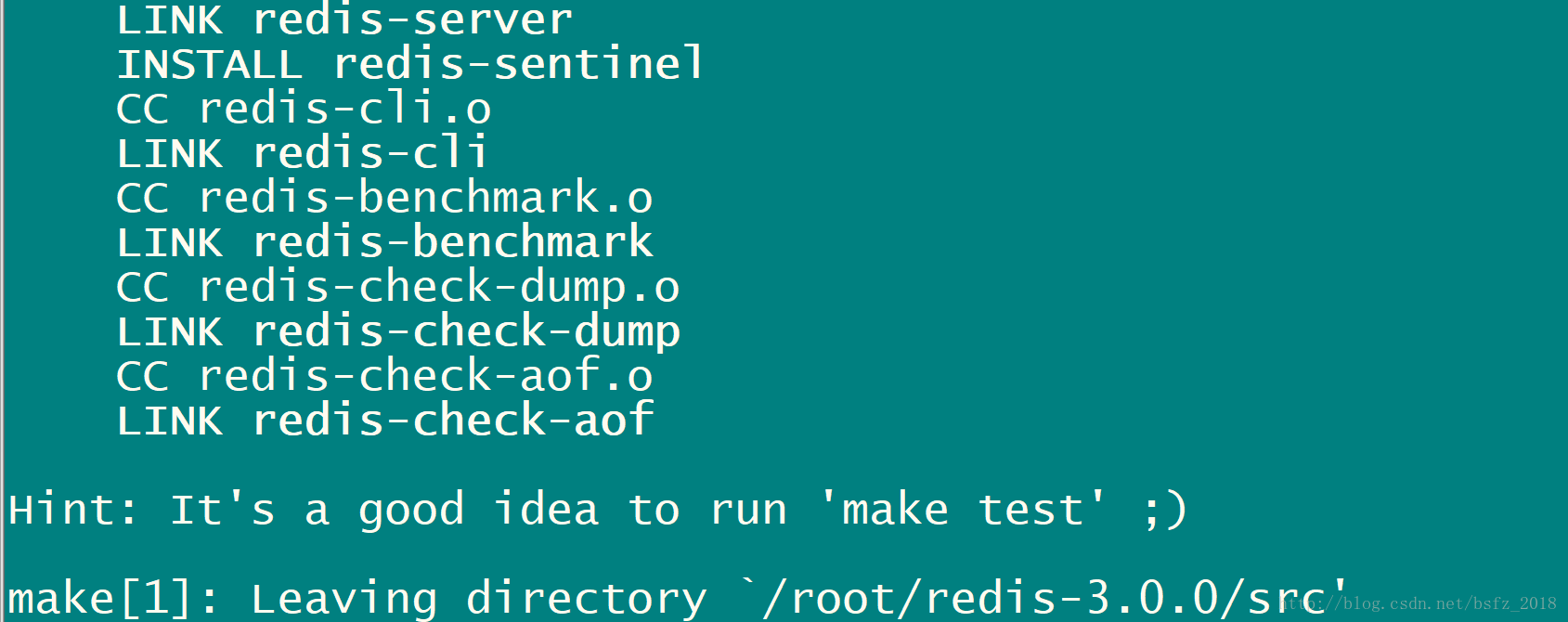


3 解压缩到本目录 tar -xzcf redis-3.0.0.tar.gz

4 进入解压缩目录中，可以查看到makefile文件



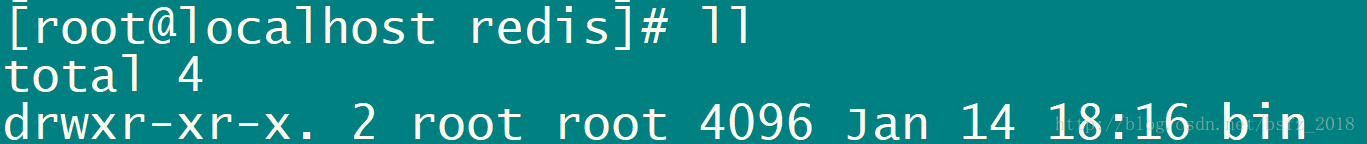
5 当前路径下输入make进行编译，编译完毕会生成如下文件



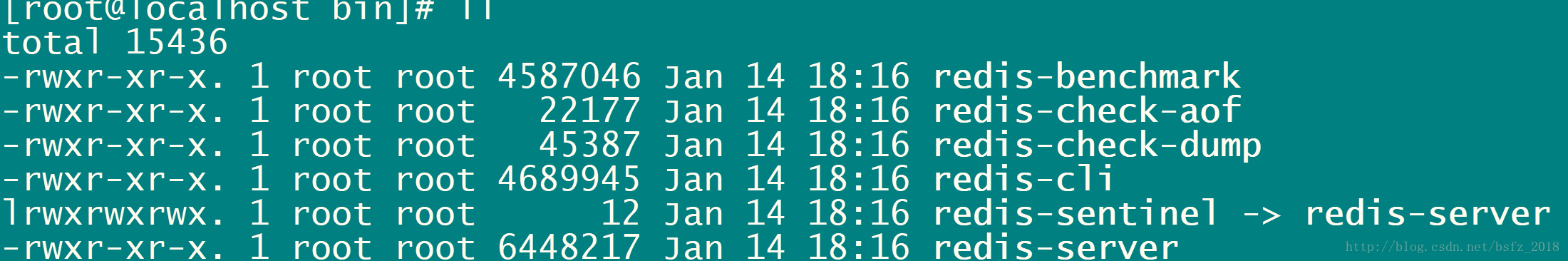
6 接下来进行安装，本次安装到如下目录中

https://img-blog.csdn.net/20180115101657440?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvYnNmel8yMDE4/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast

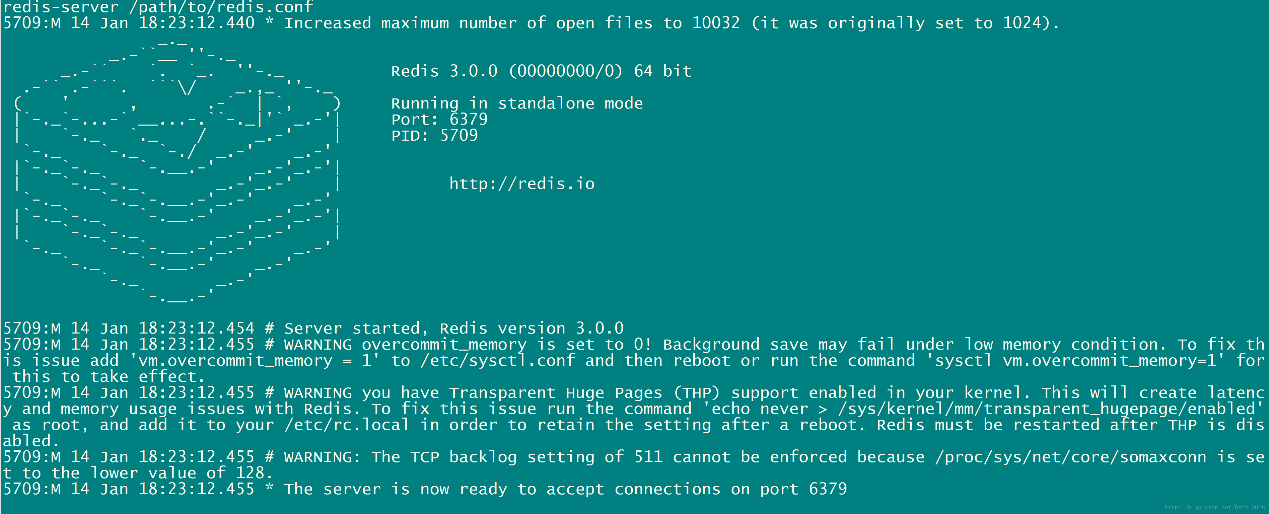
7 进入redis目录，可以看到bin文件夹



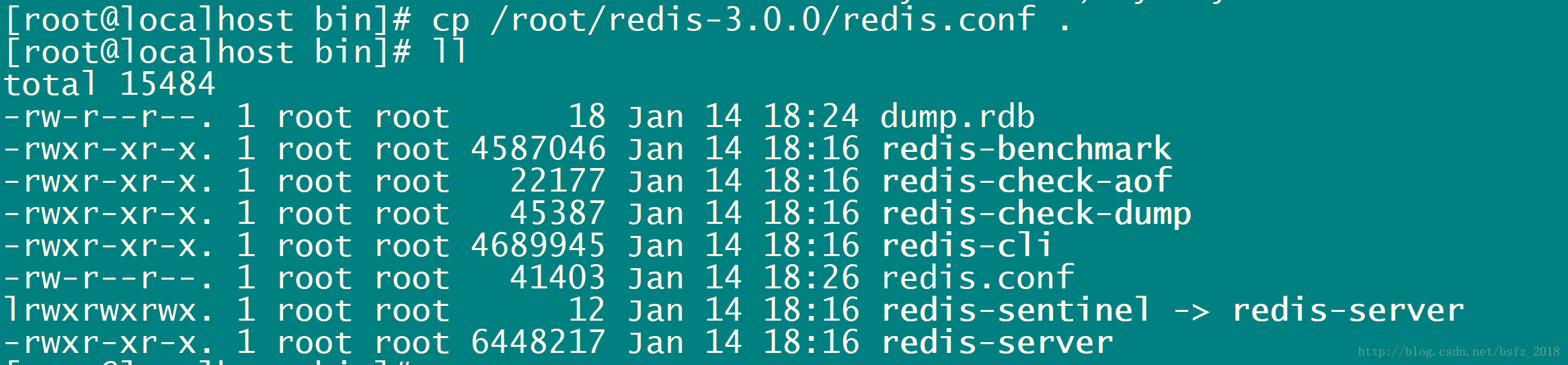
8 进入bin文件夹，可以看到redis的服务端，客户端



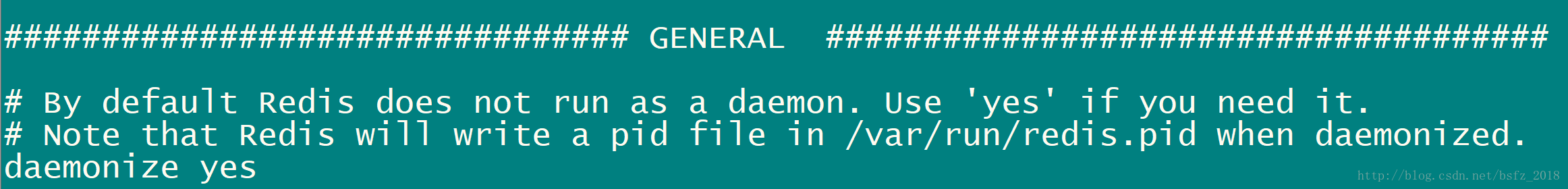
9 直接使用./redis-server启动redis服务端，出现下图即为启动成功



10 因为这样的前端启动方式会占用终端，因此不建议采用，建议采用后端启动的方式。使用ctrl+c退出，并将原用户宿主目录下的redis配置文件redis.conf拷贝到本路径



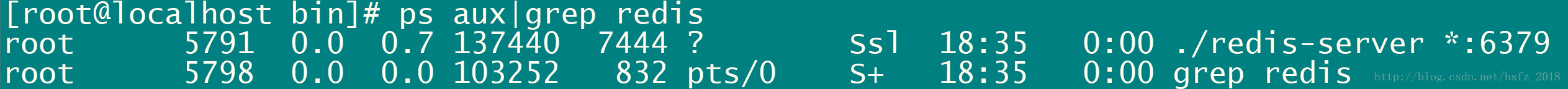
11 修改该配置文件，更改daemon选项为yes，即支持后台启动方式



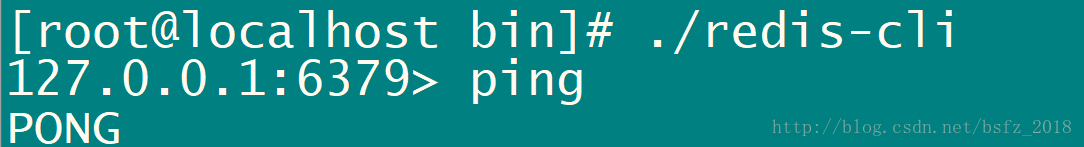
12 修改完毕按esc，然后输入:wq保存并退出，然后输入命令用指定的redis配置文件启动redis

https://img-blog.csdn.net/20180115103504214?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvYnNmel8yMDE4/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast

13 查看后台进程可以看到redis已经成功启动



14 使用redis客户端进行连接测试，可以看到已经成功



redis.conf配置文件配置项解析

daemonize：如需要在后台运行，把该项的值改为yes

pdifile：把pid文件放在/var/run/redis.pid，可以配置到其他地址

bind：指定redis只接收来自该IP的请求，如果不设置，那么将处理所有请求，在生产环节中最好设置该项

port：监听端口，默认为6379

timeout：设置客户端连接时的超时时间，单位为秒

loglevel：等级分为4级，debug，revbose，notice和warning。生产环境下一般开启notice

logfile：配置log文件地址，默认使用标准输出，即打印在命令行终端的端口上

database：设置数据库的个数，默认使用的数据库是0

save：设置redis进行数据库镜像的频率

rdbcompression：在进行镜像备份时，是否进行压缩

dbfilename：镜像备份文件的文件名

dir：数据库镜像备份的文件放置的路径

slaveof：设置该数据库为其他数据库的从数据库

masterauth：当主数据库连接需要密码验证时，在这里设定

requirepass：设置客户端连接后进行任何其他指定前需要使用的密码

maxclients：限制同时连接的客户端数量

maxmemory：设置redis能够使用的最大内存

appendonly：开启appendonly模式后，redis会把每一次所接收到的写操作都追加到appendonly.aof文件中，当redis重新启动时，会从该文件恢复出之前的状态

appendfsync：设置appendonly.aof文件进行同步的频率

vm\_enabled：是否开启虚拟内存支持

vm\_swap\_file：设置虚拟内存的交换文件的路径

vm\_max\_momery：设置开启虚拟内存后，redis将使用的最大物理内存的大小，默认为0

vm\_page\_size：设置虚拟内存页的大小

vm\_pages：设置交换文件的总的page数量

vm\_max\_thrrads：设置vm IO同时使用的线程数量