

Win10 VMware Workstation 16 安装 macOS 记录

- 1. [一、简介](#)
- 2. [二、MacOS 安装](#)
- 3. [三、安装各类软件](#)
- 4. [四、扩容分区](#)
- 5. [五、关闭系统完整性保护](#)

一、简介

这里记录了我在 Win10 VMware workstation 上配置 macOS 虚拟机所踩过的坑点。

二、MacOS 安装

- 首先，下载 VMware 解锁 MacOS 选项的补丁。

“解锁 MacOS” 的这个说法其实个人感觉不是特别直接。
这个补丁的用途是**让 VMware 额外支持 MacOS**。

1	git clone git@github.com:BDISP/unlocker.git
---	---

之后，去任务管理器中，**强制退出所有 VMware 开头的进程**，防止补丁失败：

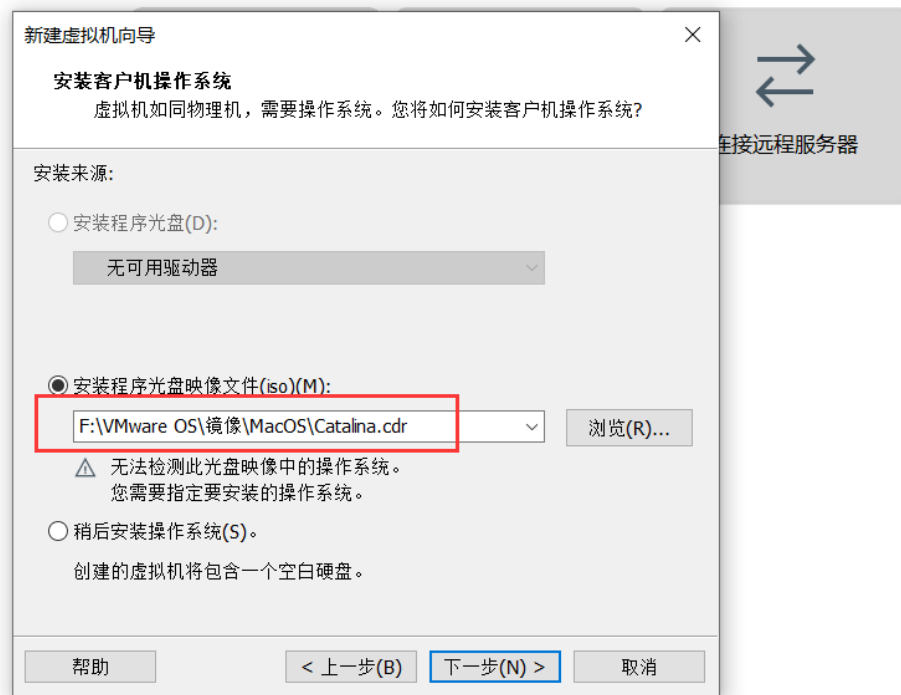
Usermode Font Driver Host	0%	0.1 MB	0 MB/秒	0 Mbps	0%
> Video Application (2)	0%	0.4 MB	0 MB/秒	0 Mbps	0%
> VMware Authorization Servi...	0%	0.6 MB	0 MB/秒	0 Mbps	0%
> VMware NAT Service (32 位)	0.1%	1.0 MB	0 MB/秒	0 Mbps	0%
> VMware USB Arbitration Ser...	0%	0.2 MB	0 MB/秒	0 Mbps	0%
> VMware VMnet DHCP servic...	0%	0.2 MB	0 MB/秒	0 Mbps	0%
VMware Workstation VMX	11.8%	27.3 MB	39.1 MB...	0 Mbps	0%
> vmware-hostd.exe (32 位)	0%	3.6 MB	0 MB/秒	0 Mbps	0%
Waves MaxxAudio Service A...	0%	5.0 MB	0 MB/秒	0 Mbps	0%

之后**管理员执行** `win-install.cmd`。执行时脚本会去 vmware 官网上下载一些东西，时间取决于网络条件。

执行完成后，**重启电脑**或手动去 **服务** 底下打开 **VMware NAT Service** 和 **VMware VMnet DHCP service** 服务，否则**虚拟机将无法连接网络**。

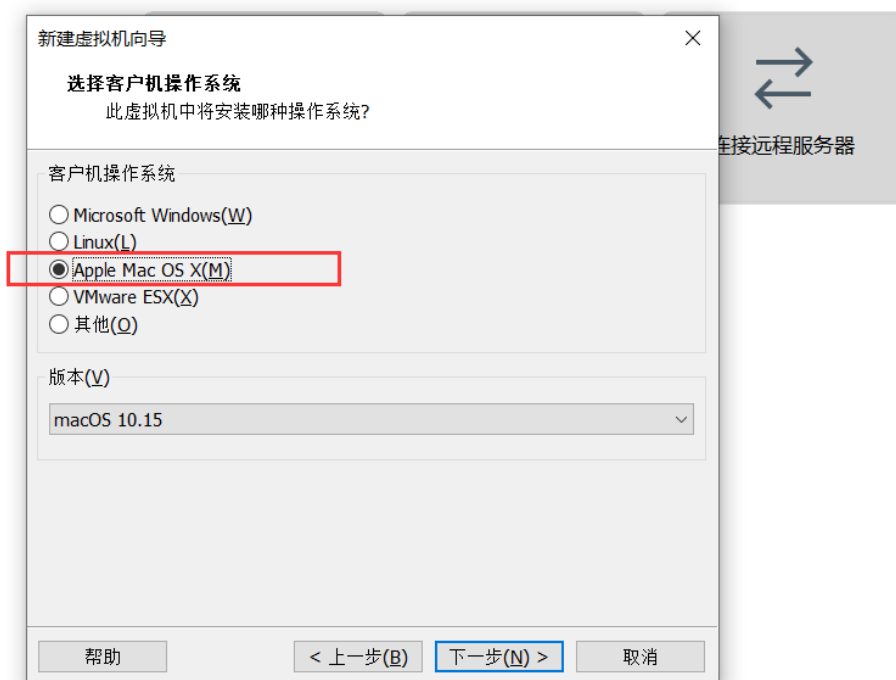
坑点：之前忘记重启网络服务了...

- 接着，去 VMware 上新建虚拟机，并指定光盘映像文件为下载下来的 ISO/CDR 文件：



然后选择 `Apple Mac OS X`，并一路 next 下去。磁盘大小建议 **至少分配70GB**。

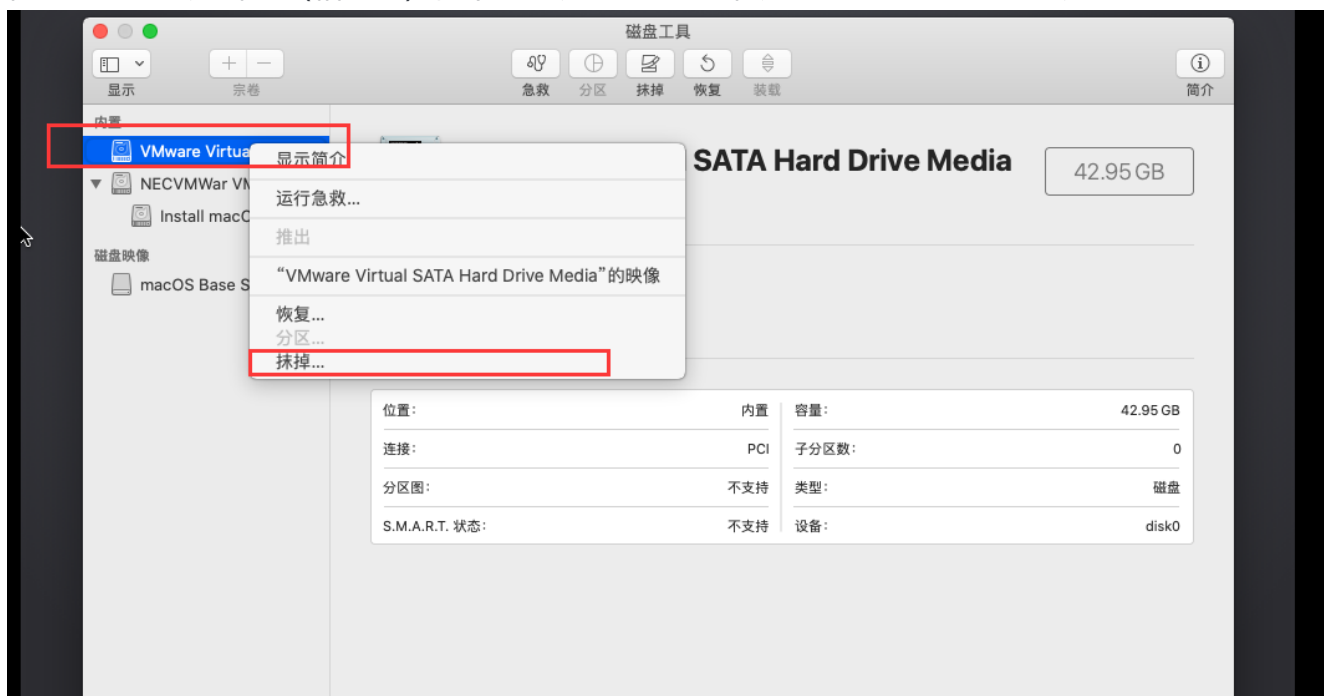
如果此时 VMware 里没有这个选项，则说明安装 VMware 补丁失败，需要重新安装最新版的补丁。



- 虚拟机建立好后，启动虚拟机。在**磁盘工具**处：



将 Vmware 磁盘抹掉（格式化），不然安装 macOS 时将无法访问到 Vmware 磁盘：



抹掉时改个磁盘名称就可以，其他的都不用动：

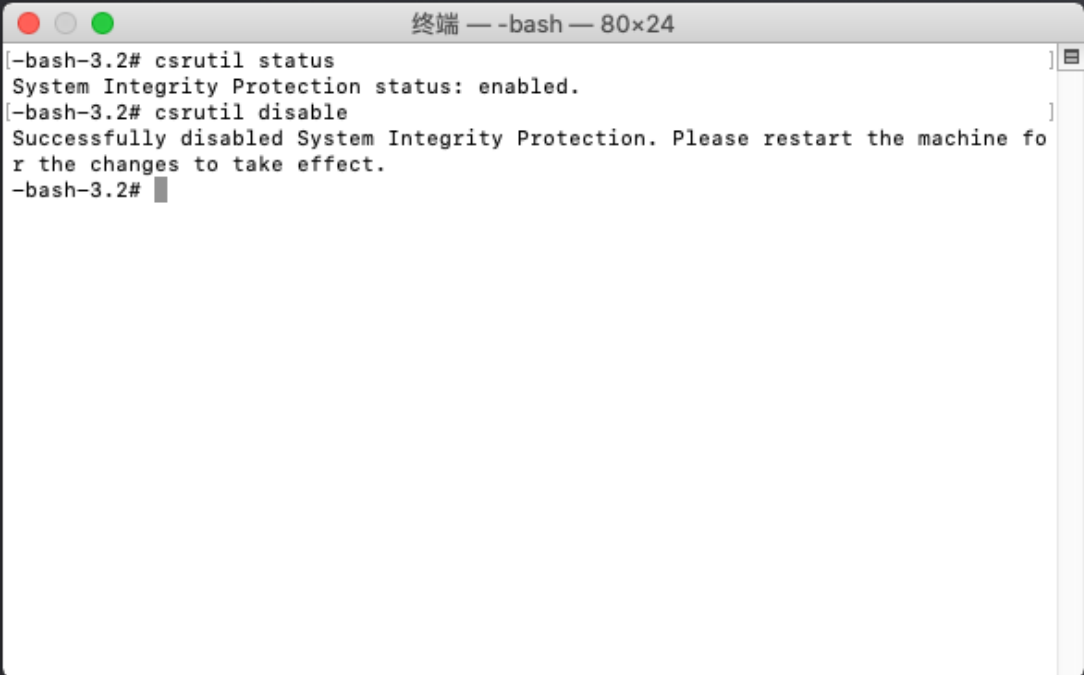


格式化磁盘后，在上方 **实用工具**->**终端**：



键入 `csrutil disable` 禁用系统完整性保护：

因为系统完整性保护会限制 root 权限的行为。



```
终端 — -bash — 80x24
[-bash-3.2# csrutil status
System Integrity Protection status: enabled.
[-bash-3.2# csrutil disable
Successfully disabled System Integrity Protection. Please restart the machine fo
r the changes to take effect.
-bash-3.2#
```

之后键入 `csrutil authenticated-root disable` 以关闭 Authenticated-root 保护。该保护会使得 MacOS 在引导期间，将一个被加密签名后的只读根文件系统快照挂载进根目录，因此我们需要禁用它以便于修改根路径或系统路径下的文件等。

如果还是不行，则在 MacOS 安装完成后，执行 `sudo mount -uw /` 试试，注意该指令只在本次开机时有效，下次开机需要重新设置。

接下来照常安装 MacOS 即可。

- MacOS 安装完成后。**不要马上启动！不要马上启动！不要马上启动！**

要先在该 MacOS 的 vmx 文件末尾追加 `smc.version = 0`，防止虚拟机出现错误。
追加完成后再启动。

- 启动新安装的 MacOS，之后一定要**立即升级当前安装的 MacOS 系统**（12GB左右）。因为 Apple 对远古版本的 MacOS 支持性非常低，就连安装软件都会有限制。

一定要在完成 MacOS 系统升级后，再去装各类软件以及 IDE 等等。

最好先安装当前远古版本 MacOS 系统的一些补丁，再去升级 MacOS 系统，不然可能有一定概率会升级失败。

我这边更新到的版本是 `macOS Monterey 12.0.1`。

三、安装各类软件

1. vmtools。右键虚拟机并点击 **安装 VMware Tools**，然后根据步骤一步步来就好。



2. AppStore 上安装

- 1. DevCleaner for xcode：释放 Xcode 缓存文件。
- 2. xcode（12GB左右）。不用多说。
- 3. 超级右键。扩展一下自己的右键菜单，使得支持**右键打开终端操作**。

3. 下载 [iStat Menus6](#)。这是 MacOS 上的一个系统监测软件，需要付费，可用序列号如下：

1	Email: 982092332@qq.com
2	SN: GAWAE-FCWQ3-P8NYB-C7GF7-NEDRT-Q5DTB-MFZG6-6NEQC-CRMUD-
3	8MZ2K-66SRB-SU8EW-EDLZ9-TGH3S-8SGA
4	ref: http://www.pc6.com/mac/111587.html

4. 安装homebrew 包管理器

安装 homebrew 时需要**多次输入密码**，切记别走开。

1	/bin/bash -c "(curl -fsSLhttps :
2	//raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)" <
3	br > cd"(brew --repo)"
4	git remote set-url origin https://mirrors.aliyun.com/homebrew/brew.git
5	git remote get-url origin
6	

如果发现 brew 安装有问题，无法搜索到任何软件包，则尝试运行 `brew doctor` 命令获取解决方案。

5. 设置双拼**自然码**方案。进入 **设置->键盘->输入法**，选择**简体双拼**，并在终端键入以下命令以启动**自然码**方案：

1	defaults write com.apple.inputmethod.CoreChineseEngineFramework shuangpinLayout 5
---	---

6. 安装 [VSCode for macOS](#)。下载后将其拖入**应用程序文件夹**下。

7. 安装 proxychain

1	brew install proxychains-ng
2	nano /usr/local/etc/proxychains.conf
3	proxychains4 curl -v google.com

8. 配置 git。

1	ssh-keygen
2	cat ~/.ssh/id_rsa.pub
3	git config --global user.name Kiprey
4	git config --global user.email Kiprey@qq.com
5	

9. 安装 ohmyzsh。

1	brew install wget
2	sh -c "\$(wget -O- https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"

之后安装常用插件

◦ autojump

执行以下命令下载：

1	git clone git://github.com/joelthelion/autojump.git
2	cd autojump
3	./install.py

之后 `nano ~/.zshrc`，将以下内容添加至文件末尾：

1	[[-s /Users/kiprey/.autojump/etc/profile.d/autojump.sh]] && source
2	/Users/kiprey/.autojump/etc/profile.d/autojump.sh
	autoload -U compinit && compinit -u

然后将 `autojump` 添加进 `.zshrc` 中的 plugin 字段中：

1	
2	
3	
4	plugins=(git autojump)
5	
6	

◦ zsh-autosuggestions 与 zsh-syntax-highlighting

下载:

1	git clone git://github.com/zsh-users/zsh-autosuggestions
2	\$ZSH_CUSTOM/plugins/zsh-autosuggestions
	git clone git://github.com/zsh-users/zsh-syntax-highlighting
	\$ZSH_CUSTOM/plugins/zsh-syntax-highlighting

将 `zsh-autosuggestions` 和 `zsh-syntax-highlighting` 添加进 `.zshrc` 中的 `plugin` 字段中:

1	plugins=(git autojump zsh-autosuggestions zsh-syntax-highlighting)

插件安装完成后, 最后执行 `source ~/.zshrc` 重新载入新的 `zsh` 配置以启动插件。

10. 安装 ShadowSocksR。下载地址: [shadowsocksX-NG-R - github](#), 支持订阅地址。

四、扩容分区

如果发现 MacOS 磁盘大小不够, 需要扩容一下虚拟磁盘, 则按照以下步骤进行:

1. 先去 Vmware 那里扩容一下磁盘
2. 在 MacOS 中, 执行 `diskutil list` 查看当前磁盘情况:

```
[kiprey@KipreydeMac Desktop % diskutil list
/dev/disk0 (internal, physical):
#:                      TYPE NAME                      SIZE      IDENTIFIER
0:      GUID_partition_scheme                      *75.2 GB   disk0
1:                  EFI EFI                          209.7 MB   disk0s1
2:                  Apple_APFS Container disk1          50.0 GB   disk0s2

/dev/disk1 (synthesized):
#:                      TYPE NAME                      SIZE      IDENTIFIER
0:      APFS Container Scheme -                      +50.0 GB   disk1
   Physical Store disk0s2
1:      APFS Volume Disk - 数据                      17.6 GB   disk1s1
2:      APFS Volume Preboot                          84.0 MB   disk1s2
3:      APFS Volume Recovery                          528.9 MB  disk1s3
4:      APFS Volume VM                               1.1 MB    disk1s4
5:      APFS Volume Disk                             11.2 GB   disk1s5
```

其中, `disk0` 为整个磁盘, 而 `disk0s2` 分区即 MacOS 此时使用的空间, 因此我们需要扩容 `disk0s2`。

3. 尝试扩展磁盘。

网络上都使用的是这个命令:

1	diskutil resizeVolume disk0s2 50GB

其中 `disk0s2` 为待扩容磁盘, `50GB` 为目标扩容大小。

但是由于本人的 `disk0s2` 为 `Apple_APFS` 类型, 因此上述命令不可使用。

需要使用以下命令:

1	diskutil apfs resizeContainer disk0s2 70GB

之后就开始扩容：

```
kiprey@KipreydeMac Desktop % diskutil apfs resizeContainer disk0s2 70GB
Started APFS operation
Aligning grow delta to 19,999,997,952 bytes and targeting a new physical store size of 69,999,996,928 bytes
Determined the maximum size for the targeted physical store of this APFS Container to be 74,951,143,424 bytes
Resizing APFS Container designated by APFS Container Reference disk1
The specific APFS Physical Store being resized is disk0s2
Verifying storage system
Using live mode
Performing fsck_apfs -n -x -l -S /dev/disk0s2
Checking the container superblock
Checking the EFI jumpstart record
Checking the space manager
Checking the space manager free queue trees
Checking the object map
Checking volume
Checking the APFS volume superblock
The volume Disk - 数据 was formatted by hfs_convert (1412.11.7) and last modified by apfs_kext (1412.141.1)
Checking the object map
Checking the snapshot metadata tree
Checking the snapshot metadata
Checking the extent ref tree
Checking the fsroot tree
Checking volume
Checking the APFS volume superblock
The volume Preboot was formatted by hfs_convert (1412.11.7) and last modified by apfs_kext (1412.141.1)
Checking the object map
Checking the snapshot metadata tree
Checking the snapshot metadata
Checking the extent ref tree
Checking the fsroot tree
Checking volume
Checking the APFS volume superblock
The volume Recovery was formatted by diskmanagementd (1412.11.7) and last modified by apfs_kext (1412.141.1)
Checking the object map
Checking the snapshot metadata tree
Checking the snapshot metadata
Checking the extent ref tree
Checking the fsroot tree
Checking volume
Checking the APFS volume superblock
The volume VM was formatted by diskmanagementd (1412.11.7) and last modified by apfs_kext (1412.141.1)
Checking the object map
Checking the snapshot metadata tree
Checking the snapshot metadata
Checking the extent ref tree
Checking the fsroot tree
Checking volume
Checking the APFS volume superblock
The volume Disk was formatted by diskmanagementd (1412.11.7) and last modified by apfs_kext (1412.141.1)
Checking the object map
Checking the snapshot metadata tree
Checking the snapshot metadata
Checking the extent ref tree
Checking the fsroot tree
```

4. 扩容成功

```
kiprey@KipreydeMac Desktop % diskutil list
/dev/disk0 (internal, physical):
#:          TYPE NAME                    SIZE      IDENTIFIER
0:          GUID_partition_scheme       *75.2 GB  disk0
1:          EFI EFI                     209.7 MB  disk0s1
2:          Apple_APFS Container disk1   70.0 GB   disk0s2

/dev/disk1 (synthesized):
#:          TYPE NAME                    SIZE      IDENTIFIER
0:          APFS Container Scheme -      +70.0 GB  disk1
              Physical Store disk0s2
1:          APFS Volume Disk - 数据       17.6 GB   disk1s1
2:          APFS Volume Preboot           84.0 MB   disk1s2
3:          APFS Volume Recovery          528.9 MB  disk1s3
4:          APFS Volume VM                1.1 MB    disk1s4
5:          APFS Volume Disk              11.2 GB   disk1s5
```

五、关闭系统完整性保护

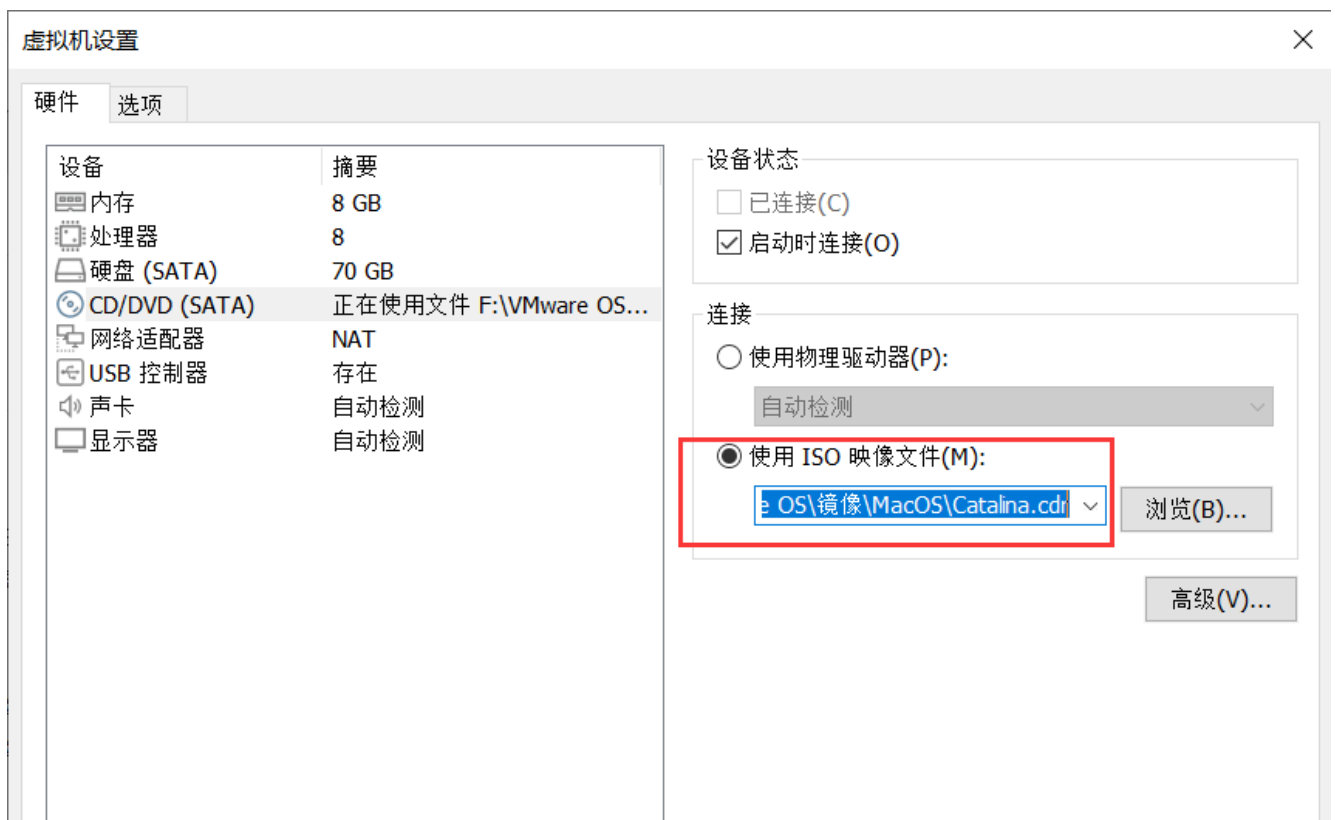
MacOS 中的系统完整性保护(SIP), 会限制住 root 用户的权限, 因此需要将其关闭。

见过用 root 权限 lladb attach 其他进程时, 被拒绝的快乐嘛...

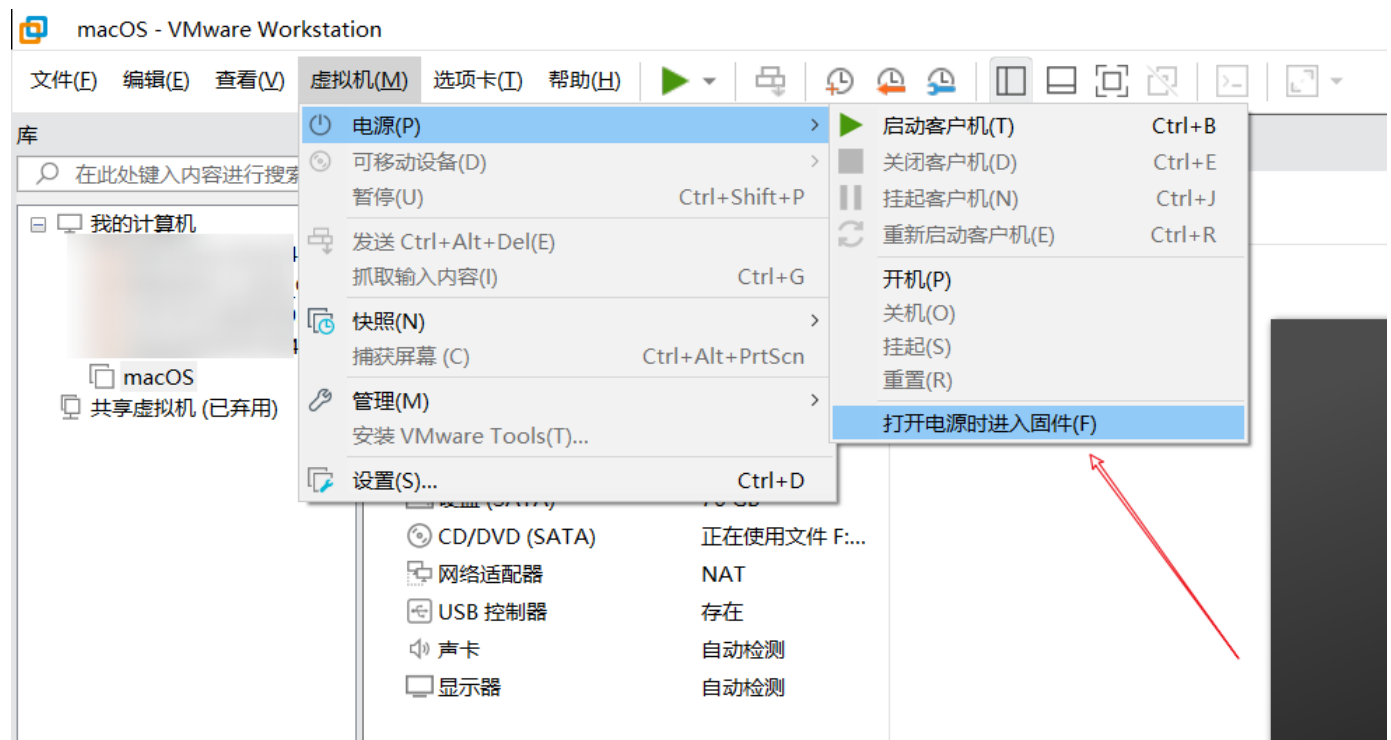
最简单的关闭方式, 莫过于上面在一开始安装时就将其关闭。

但要是当时安装时忘记关闭, 那么现在去关闭 SIP 就会稍微折腾一点...

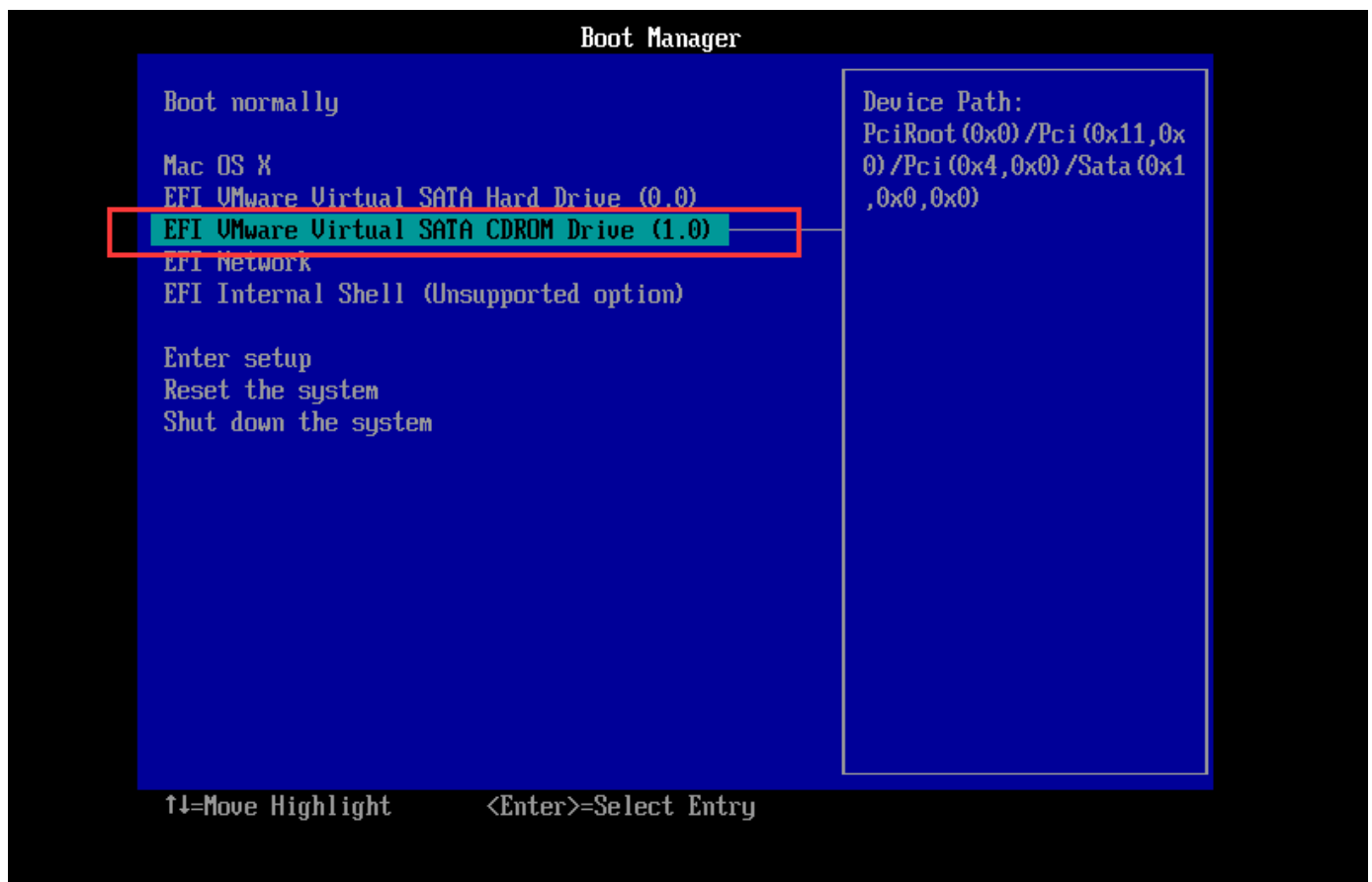
1. 设置虚拟机 CD/DVD 路径为原先的 MacOS 安装镜像：



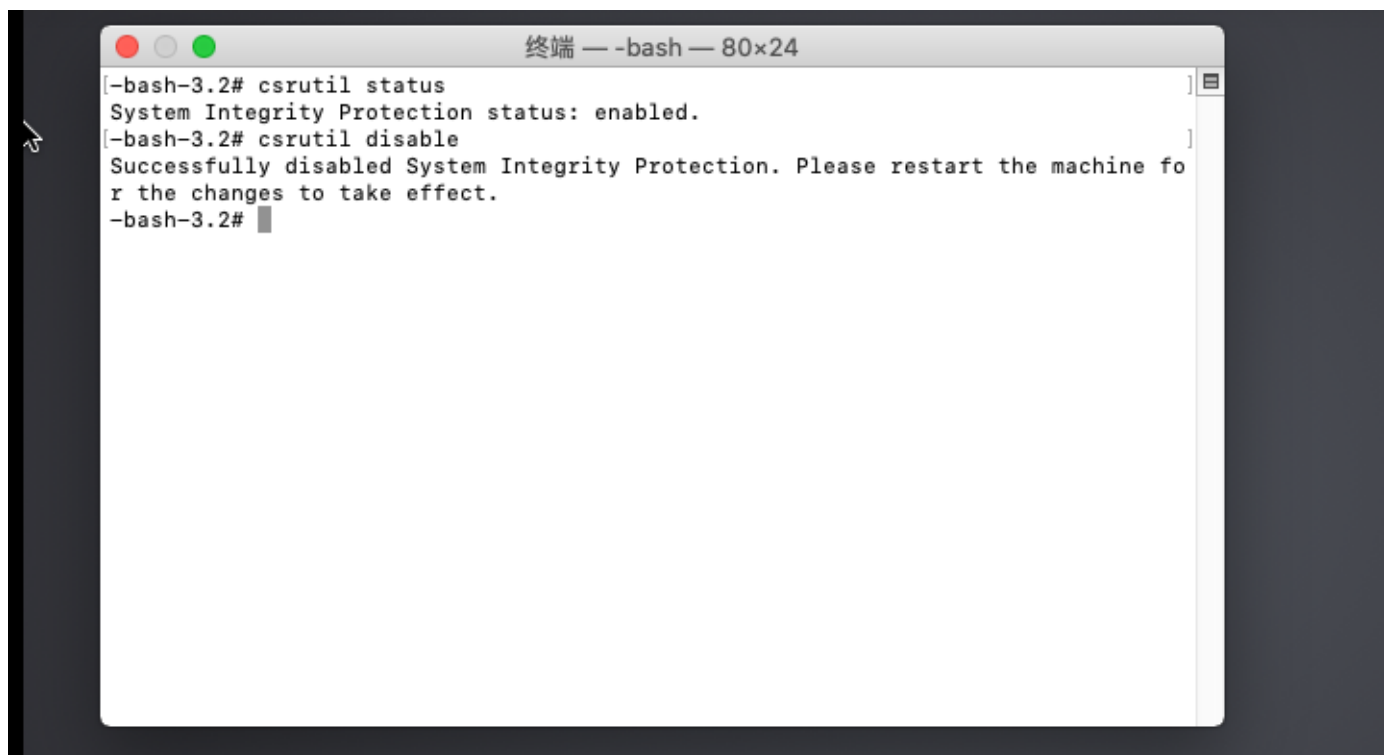
之后，进入虚拟机 BIOS



选择以 CD 为启动盘：



之后在启动后的界面，进入 **实用工具->终端** 下，键入 `csrutil disable` 命令并重启虚拟机，即可关闭 SIP。



- **版权声明：** 本博客所有文章除特别声明外，著作权归作者所有。转载请注明出处！

[上一篇](#)

[MacOSX Mach IPC 入门下一篇](#)

[Reversing.kr 刷题笔记 - 1](#)

