**05-磁盘操作**

1. **VMware管理**

1.1 **虚拟机磁盘压缩管理**

1、打开Ubuntu虚拟机的终端

|  |
| --- |
| C++ sudo su #使用root权限 |

|  |
| --- |
| C++ cat /dev/zero > zero.fill |

#将占用空间却无法清理的东西转变成一个 zero.fill文件，这个过程的时间会比较长，最后出现：

cat: 写入错误: 设备上没有空间

|  |
| --- |
| C++ rm -f zero.fill |

#将这个zero.fill文件彻底删除

注意：在此过程中并不会增大虚拟机vmdk文件的容量，因此不用担心硬盘分区被挤爆。

|  |
| --- |
| C++ sudo shutdown now #关闭虚拟机 |

2、VMWare硬盘压缩

编辑虚拟机设置——硬件——硬盘（SATA），选择实用工具——压缩（C）

过一段时间，等压缩完毕后，将VMWare关闭。

重新打开VMWare，查看一下硬盘——容量——当前大小，应该会和虚拟机实际硬盘占用相差不大了。

我的Ubuntu虚拟机压缩前是29GB，压缩后仅剩8.4GB，整整回收了20GB空间。

其实，Linux虚拟机压缩按照这个方法是可靠的，不过安全起见，我在第一次做压缩操作的时候还是将整个虚拟机目录拷贝到另一个分区去做，最后确认没问题了才拷贝回来覆盖原来的目录。

1.2 **扩容工具**

Gparted

1.3 **fisk应用**

cd ~/build

#创建一个空镜像，实际大小根据你的roofs文件系统大小设定，可使用 du -h ~/build/linux-rootfs 命令查看

fallocate -l 1024M rootfs.img

#对空白镜像进行分区

fdisk rootfs.img

1. 输入o，对分区进行初始化，
2. 输入n，然后默认p回车，起始扇区默认，终止扇区输入+150M，表示第一个扇区150M大小（BOOT分区），
3. 输入t，然后输入c，设置第一个分区格式为W95 FAT32 (LBA)，
4. 输入n，然后全部默认敲击回车，将剩余空间全部设置为第二个分区，
5. 输入w，保存并退出。

#可使用命令查看新的分区

fdisk -l rootfs.img

#挂载分区

sudo kpartx -av rootfs.img

#格式化分区

sudo mkfs.vfat -n BOOT /dev/mapper/loop15p1

sudo mkfs.ext4 -F -L ROOTFS -O "^has\_journal" /dev/mapper/loop15p2

#挂载到电脑上

sudo mkdir {/mnt/loopXp1,/mnt/loopXp2}

sudo mount /dev/mapper/loop15p1 /mnt/loopXp1

sudo mount /dev/mapper/loop15p2 /mnt/loopXp2

#复制至boot分区

sudo cp -r firmware/boot/\* /mnt/loopXp1/

sudo cp linux/arch/arm64/boot/dts/broadcom/\*.dtb /mnt/loopXp1