# fat32

实验报告：实现对FAT32磁盘的读写

#### 1. 实验目标

本实验的目标是在Ubuntu环境下，通过对FAT32磁盘进行读写操作，包括文件和目录的创建、删除、复制等操作。

#### 2. 实验工具

mkfs.vfat: 用于创建FAT32文件系统的工具。

mount: 用于挂载磁盘分区的工具。

umount: 用于卸载磁盘分区的工具。

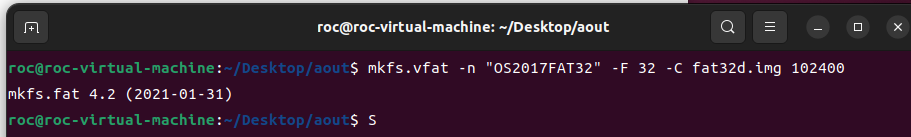
hexedit: 用于查看和编辑二进制文件的工具。

#### 3. 实验步骤

3.1 创建FAT32文件系统

使用mkfs.vfat命令创建一个FAT32文件系统的磁盘镜像文件：

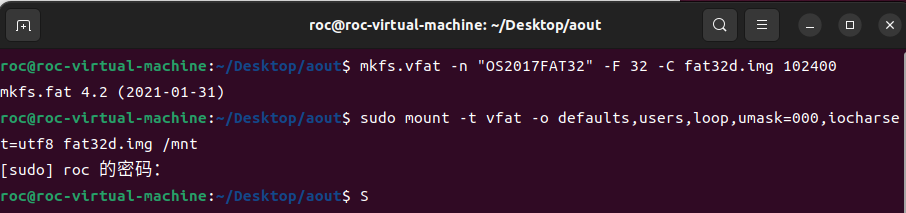
mkfs.vfat -n "OS2017FAT32" -F 32 -C fat32d.img 102400



3.2 挂载磁盘分区

使用mount命令将FAT32文件系统挂载到指定目录：

sudo mount -t vfat -o defaults,users,loop,umask=000,iocharset=utf8 fat32d.img /mnt

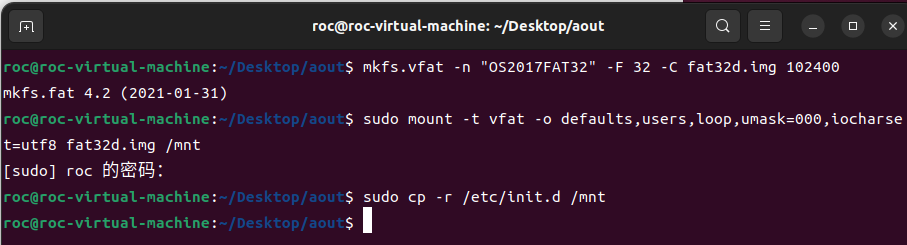


3.3 操作磁盘分区

在挂载后，可以使用常规文件操作命令进行文件和目录的创建、复制、删除等操作。例如：

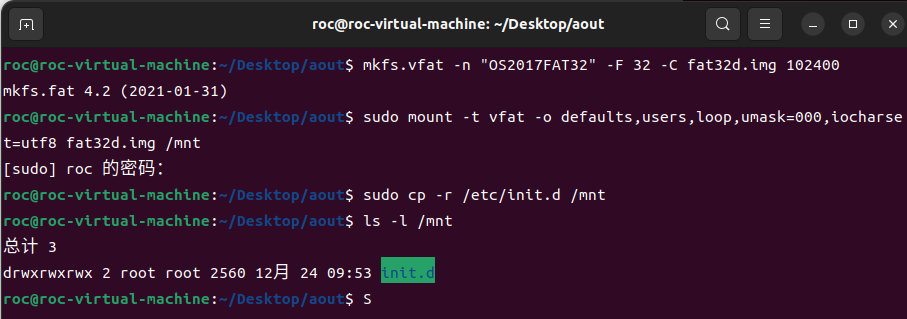
# 复制文件到挂载目录

sudo cp -r /etc/init.d /mnt



# 查看挂载目录内容

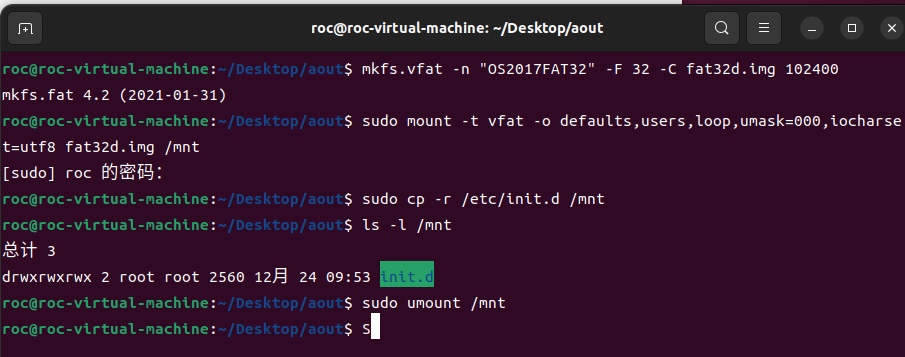
ls -l /mnt



3.4 卸载磁盘分区

使用umount命令卸载磁盘分区：

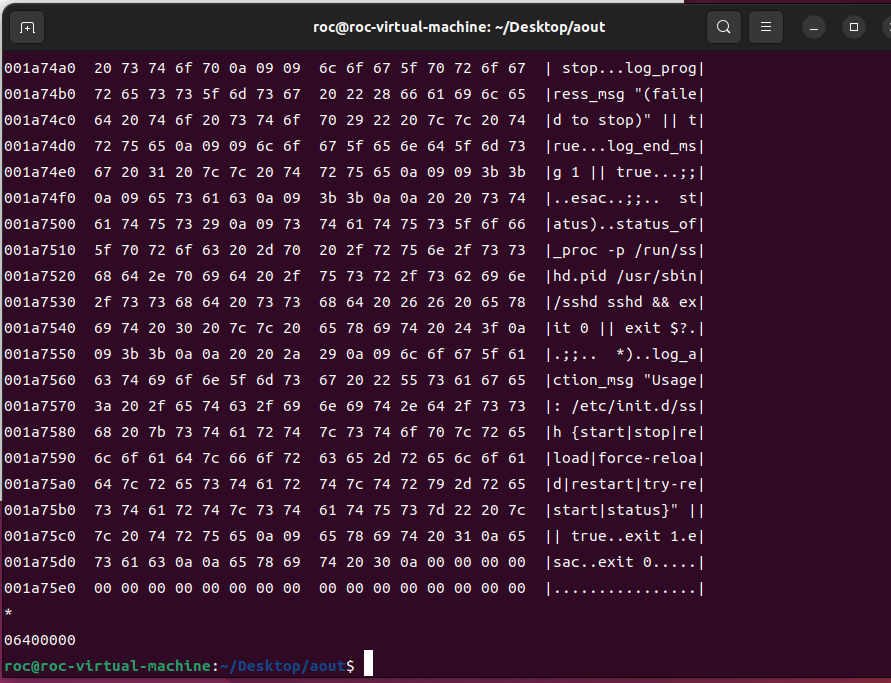
sudo umount /mnt



3.5 查看和编辑磁盘文件

使用hexedit命令查看和编辑磁盘文件，了解磁盘的结构和存储信息。

hexedit fat32d.img



#### 4. 文件系统结构

通过hexedit查看，可以了解FAT32文件系统的结构，包括引导扇区、文件系统信息扇区、FAT表、数据区等。

在FAT32文件系统中，FAT表（File Allocation Table）记录了文件和簇之间的分配关系。通常，FAT32包含两个FAT表，分别命名为FAT1和FAT2。以下是FAT表的结构：

4.1 结构概览：

FAT项大小： 每个FAT项通常是32位（4字节）大小。

FAT表项数量： FAT表的项数量与簇的数量相匹配。每个簇对应FAT表中的一个项。

FAT表的起始位置： FAT表紧随引导扇区之后，占用了若干扇区。

FAT表的作用： FAT表中的每一项都包含了下一个簇的编号或者特殊标记，指示了文件或目录中下一个数据部分的位置。

4.2具体内容：

簇的编号： FAT表中的每个项存储了下一个簇的编号。通过跟随FAT表中的项链，可以遍历整个文件或目录所涉及的簇。

特殊值： 有一些特殊的值表示了不同的状态，比如0x0代表簇为空闲，0xFFFFFFFF（或0xFFFFFFF8）表示簇的结束。

#### 5. 实验总结

通过本实验，成功实现了对FAT32磁盘的读写操作。掌握了文件系统的基本结构，学会了使用命令挂载和卸载磁盘分区，以及对磁盘进行操作，对文件系统的知识有了进一步的了解。