# Linux 101 分区 备份 批处理

- 文件系统是什么?
- 分区是什么?
- 如何玩分区?
- 文件权限 · 进阶
- 如何进行备份?
- 如何编写脚本进行批处理?

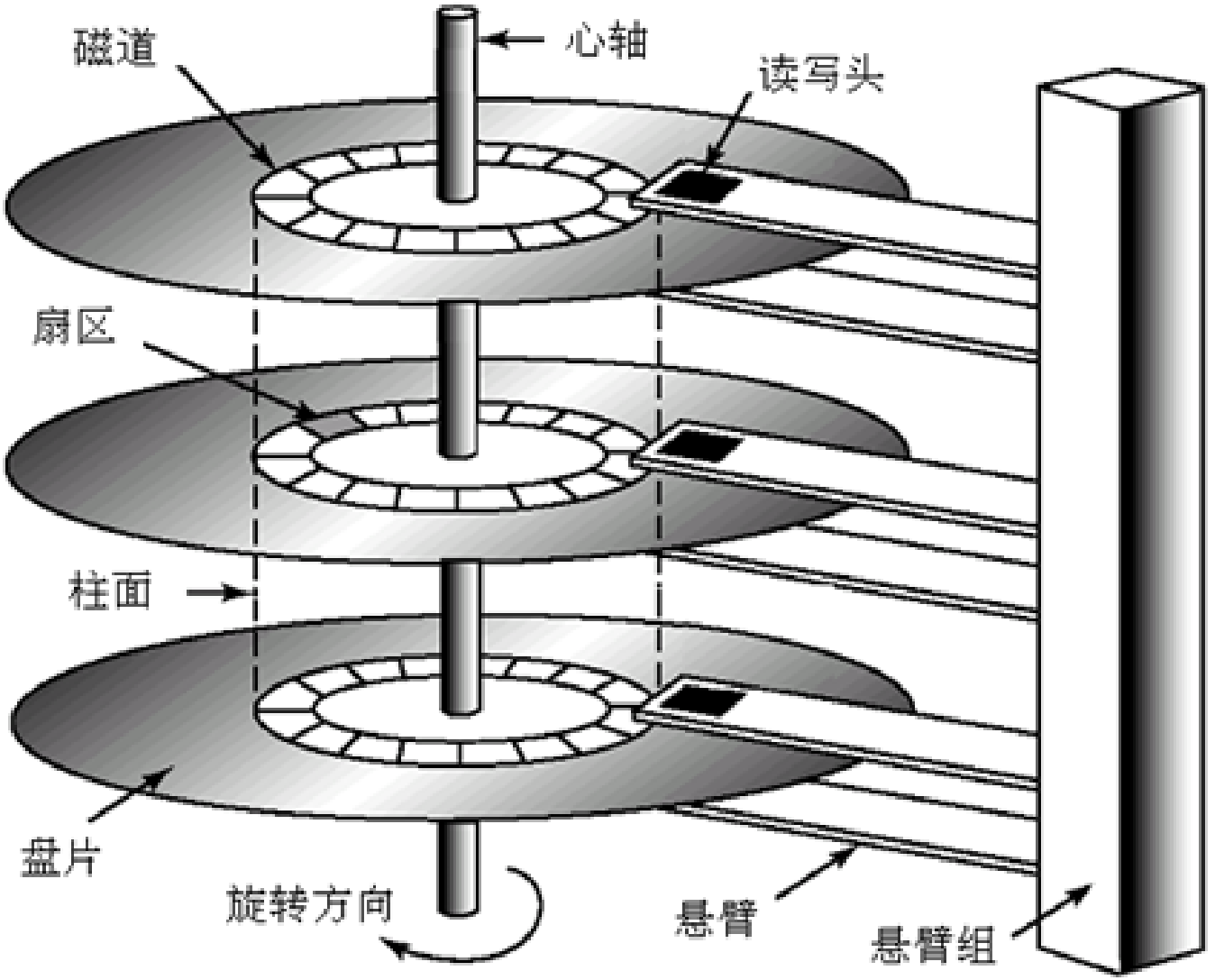
# FBI Warning

RTFM
Read The F**king Manual

STFW
Search The F**king Web

# 这是啥?

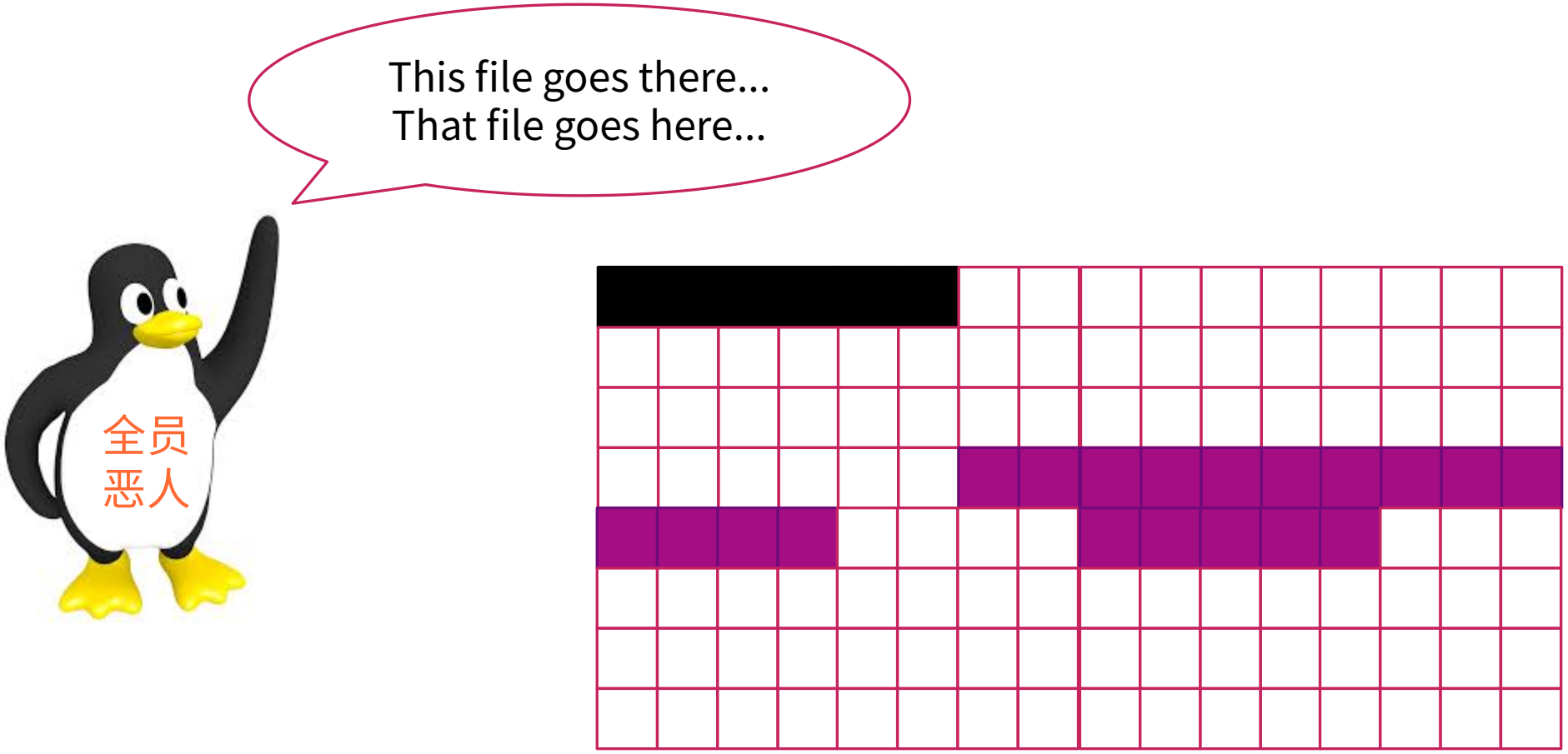
# emmmmmm

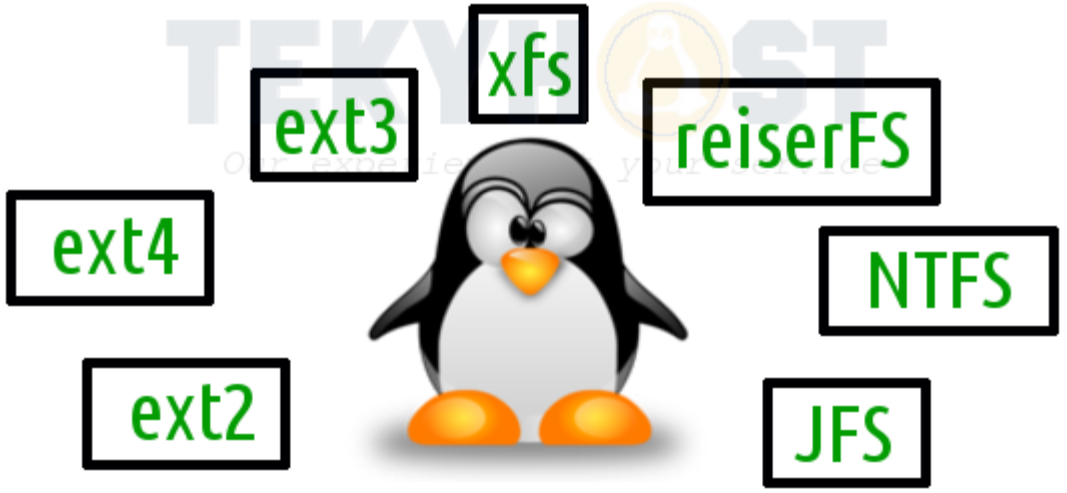


# 总之，操作系统看到的是

# 我的宝贵文件是怎么放在这 Chaos 里的?

# 文件系统




xfs
ext3
reiserFS
ext4
NTFS
ext2
JFS


NTFS or FAT 32?

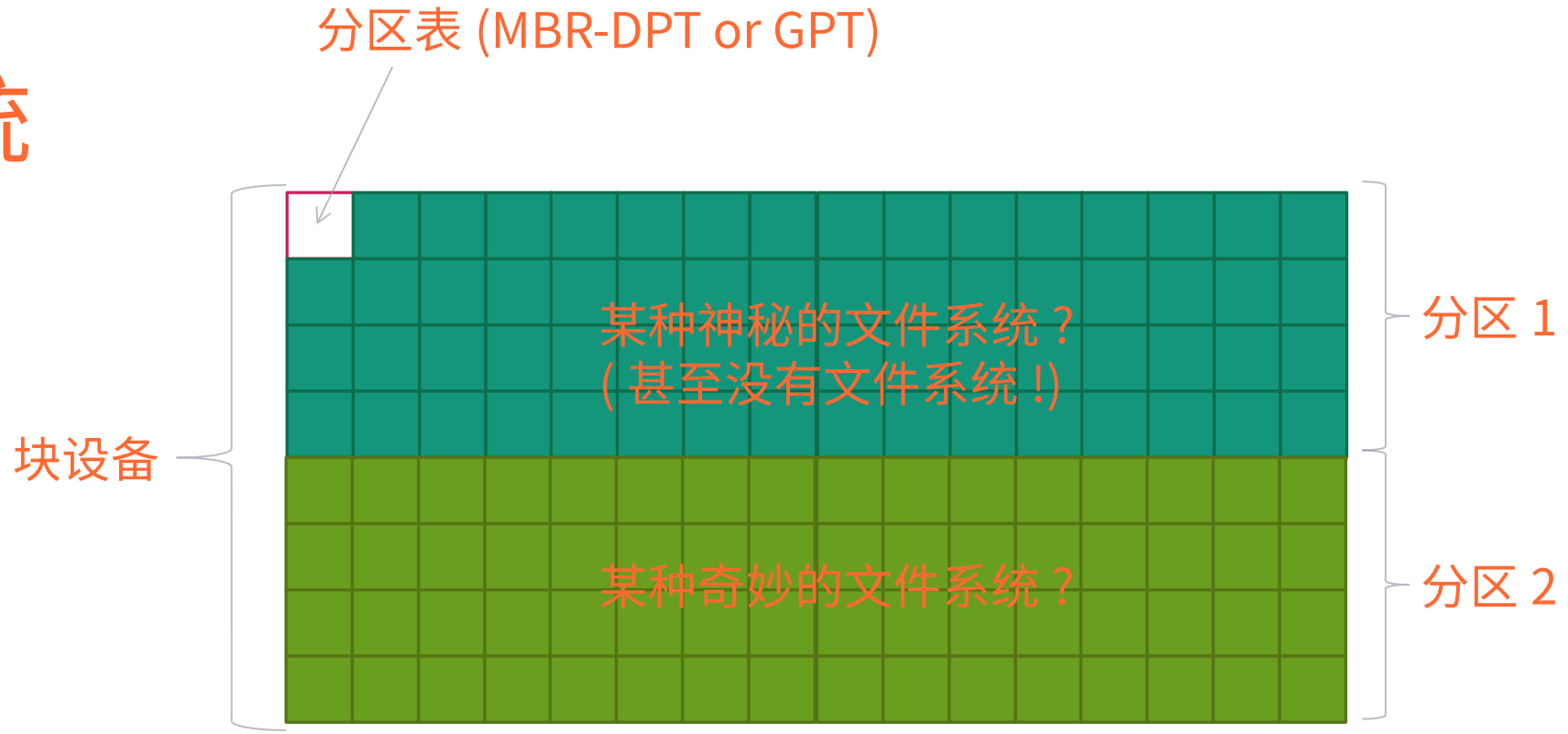

APFS vs HFS+

# 概念辨析

文件系统

分区

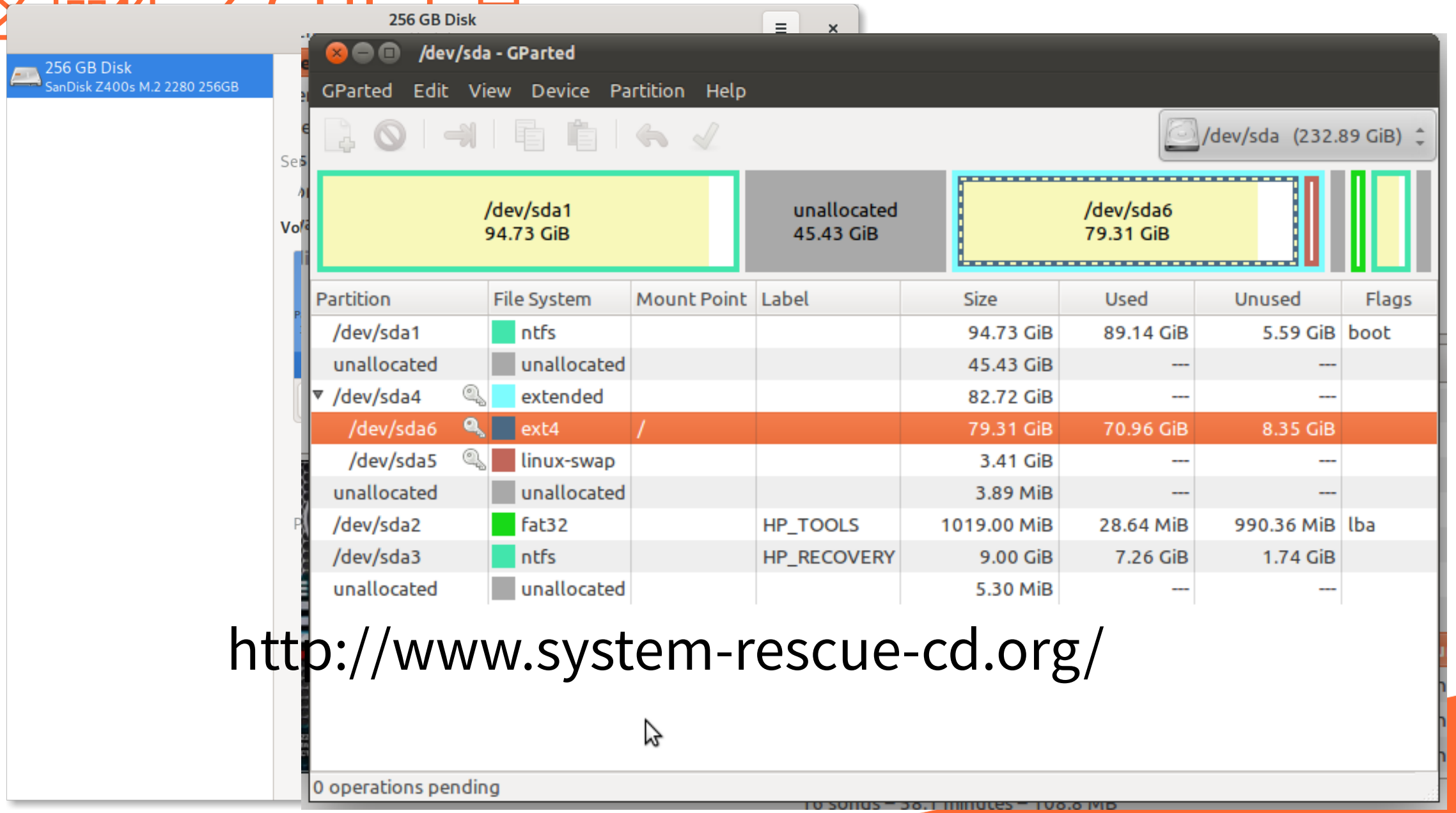
逻辑卷



# man 7 hier


file system??

# 分区操作之 GUI 工具




http://www.system-rescue-cd.org/

# 分区操作之 CLI 工具

sudo fdisk /dev/sda

Ctrl+C 取消

partprobe 重新扫描分区表

```
Help:

  GPT
   M   enter protective/hybrid MBR

  Generic
   d   delete a partition
   F   list free unpartitioned space
   l   list known partition types
   n   add a new partition
   p   print the partition table
   t   change a partition type
   v   verify the partition table
   i   print information about a partition

  Misc
   m   print this menu
   x   extra functionality (experts only)

  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file

  Save & Exit
   w   write table to disk and exit
   q   quit without saving changes

  Create a new label
   g   create a new empty GPT partition table
   G   create a new empty SGI (IRIX) partition table
   o   create a new empty DOS partition table
   s   create a new empty Sun partition table
```

# 文件系统的使用（挂载和卸载）

mount < 设备文件 > < 挂载点 > [ 其他参数 ]

必须事先作为文件夹存在

mount < 设备文件 >
mount < 挂载点 >

umount < 设备文件 >
umount < 挂载点 >

- 如果挂载点之前已经包含其他文件，挂载时会发生什么？

# fsck (file system check)

fsck < 设备文件 >
fsck < 挂载点 >

```
ksqsf@inspiron:~$ sudo fsck /tmp/disk
fsck from util-linux 2.33.1
e2fsck 1.44.5 (15-Dec-2018)
/tmp/disk: clean, 18/25688 files, 8912/102400 blocks
```

# /etc/fstab

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda2 during installation
UUID=ddc13a8f-3d9a-4ac0-8482-b0a572ea548d /               ext4    errors=remount-ro 0       1
# /boot/efi was on /dev/sda1 during installation
UUID=861A-60BC  /boot/efi       vfat    umask=0077      0       1
# swap was on /dev/sda3 during installation
UUID=ec3a4bd4-3e78-4c35-a58f-68f50432857e none            swap    sw              0       0
~
```

# 文件系统的创建

```
# mkfs.ext4 < 设备文件名 >
```

总之，就是 mkfs.XXXX 就对了
有一个 mkfs 程序，可以通过 -t 参数传入想要的 FS 类型

一个例子 (cwd=/tmp):

```
truncate disk --size 100M      # 创建一个 100M 的文件
sudo mkfs -t ext4 disk         # 在 disk 上建立 ext4
sudo mount disk /mnt           # 把 disk 挂载到 /mnt
```

# 文件系统的删除（？）

ご存知でしょうか ....
格式化

# 文件系统统计信息 (df)

```
ksqsf@inspiron:/$ df
Filesystem     1K-blocks      Used Available Use% Mounted on
udev             3916772        20   3916752   1% /dev
tmpfs             789208     31328    757880   4% /run
/dev/sda2      239260292 204384416  22652440  91% /
tmpfs            3946040     29888   3916152   1% /dev/shm
tmpfs               5120         4      5116   1% /run/lock
tmpfs            3946040         0   3946040   0% /sys/fs/cgroup
/dev/loop0         90624     90624         0 100% /snap/core/6964
/dev/loop1         91648     91648         0 100% /snap/core/6818
/dev/loop3        137216    137216         0 100% /snap/mathpix-snipping-tool/36
/dev/loop2        137216    137216         0 100% /snap/mathpix-snipping-tool/35
/dev/loop4         91392     91392         0 100% /snap/core/6673
/dev/sda1          98304     16799     81505  18% /boot/efi
tmpfs             789208     13468    775740   2% /run/user/1000
/dev/loop5         95054      1563     86323   2% /tmp/mnt
```

# 磁盘占用查看 (du)

```
ksqsf@inspiron:~/bin$ du -h ~/bin/
920K    /home/ksqsf/bin/dosbox/masm
924K    /home/ksqsf/bin/dosbox
7.7M    /home/ksqsf/bin/
```

```
ksqsf@inspiron:/tmp$ du -h --apparent-size disk
100M    disk
ksqsf@inspiron:/tmp$ du -h disk
424K    disk
```

# 高级文件属性

```
chattr   设置属性 +    -     -R      -p
lsattr   列出属性

'a' only append
'c' compress（不被 ext2/3/4 支持）
'd'  不被 dump
'C'  不使用 Copy on Write
'i' immutable （真·只读）
```

# 高级文件权限

SBIT: Sticky bit, 对文件无效，在目录上使用时，目录中的所有文件只能由其所有者删除或移动。如 /tmp 。 t = sbit+x, T = sbit. chmod o+t file

```
drwxrwxrwt   - root   24 5 月  23:54 tmp
```

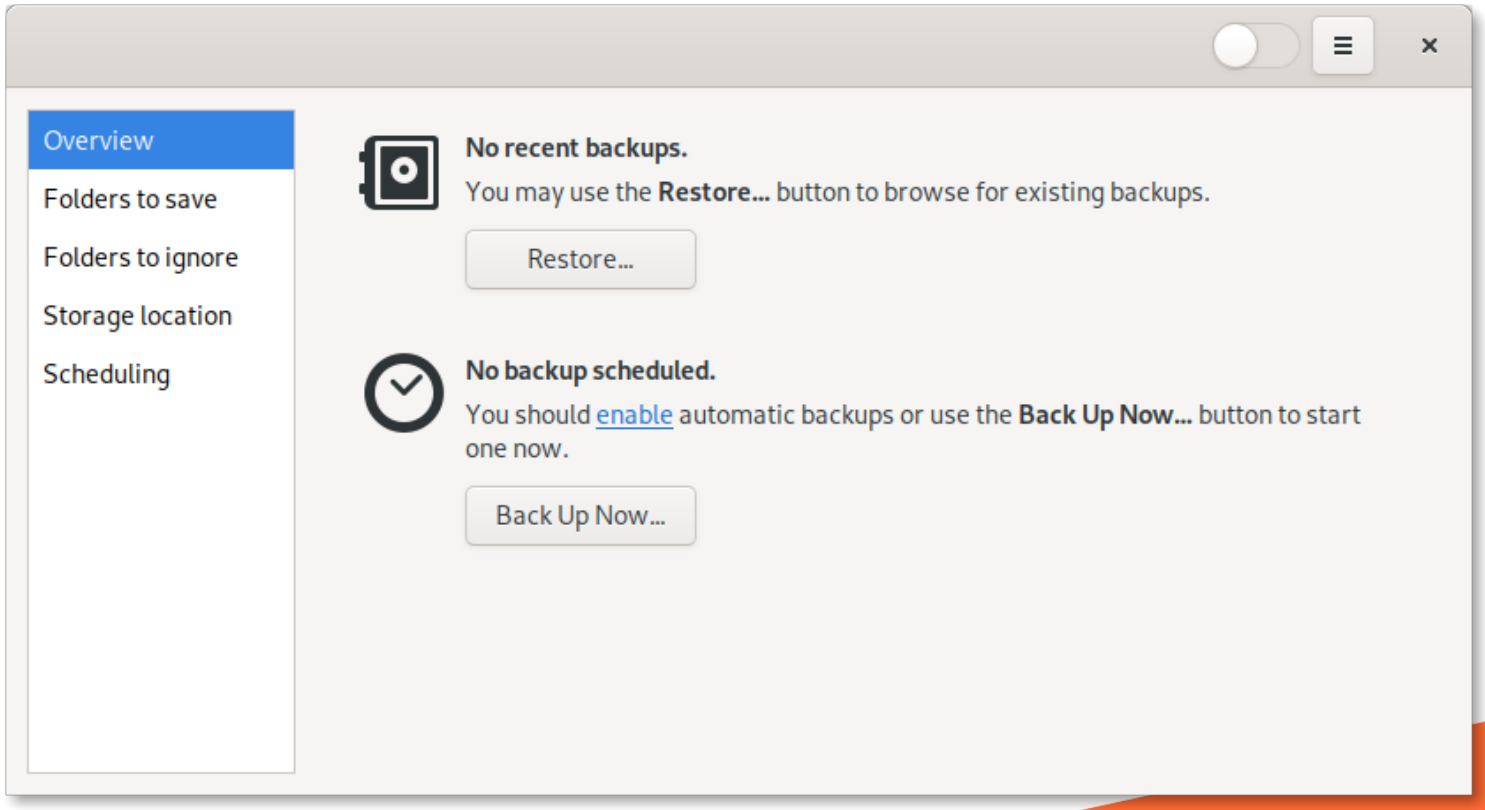
setuid: 可执行文件置有效用户为 owner 。 chmod u+s file
对目录无影响。

setgid: 可执行文件置有效组为 owner 。 chmod g+s file
目录内创建的文件属于父目录所属的组。

```
.rwsr-xr-x   157k root   13 1 月  3:10 sudo
```

# 备份

- rsync （高级 cp）
- dump
- Deja Dup & duplicity



# 备份与还原之 dump 与 restore

```
dump -0 -f <输出> <文件夹>


dump [-0123456789acLnSu] [-B records] [-b blocksize]
[-C cachesize] [-D dumpdates] [-d density] [-f file |
-P pipecommand] [-h level] [-s feet] [-T date]
filesystem

dump -W  哪些文件系统需要备份
dump -w  只显示 fstab 里的
```

# 备份与还原之 dump 与 restore

```
restore -f backup -i  #  交互式模式

cd /tmp/mp
restore -x -a -f /tmp/backup [ 文件列表 ]
```

# 备份与还原之 duplicity

```
duplicity /home/me sftp://me@lab/backup
duplicity full /home/me sftp://me@lab/backup
duplicity --full-if-older-than 1M ...

duplicity list-current-files sftp://me@lab/backup
duplicity remove-older-than <time> sftp://me@lab/backup

duplicity sftp://me@lab/backup /home/me [--force]
duplicity REMOTE LOCAL --file-to-restore FILE -t TIME
```

# find (1)

找到 90 天前的全部备份并删除

```
find backups                    #  目录存满了各种 full backup
   -name '?backup*'             #  文件名模式
   -type f                      #  普通文件， d 代表目录
   -mtime 90                    # 90 天前修改过 , ctime, atime
   -delete
```

# find (2)

把 180 天都没有访问过的备份文件移动到另外的地方

```
  find backups -name 'backup*' -atime 180 -print0
| sort -z | tee filelist
| xargs -r0 -I{} mv {} /old_backups/

find backups -name 'backup*' -atime 180
-exec mv {} /old_backups/ \;
```

# 自动备份脚本 maint.sh

```
CMD="$1"    #  子命令
SRC="$2"    #  待备份目录
DST="$3"    #  存放备份文件的目录

if [[ -z $CMD ]]; then
    echo "What do you want to do?"
    exit 1
elif [[ -z $SRC || -z $DST ]]; then
    echo "May I have your source and destination?"
    exit 1
else
    echo "Ok, let's do $CMD!"
fi
```

# 自动备份脚本 maint.sh

```
DATE=$(date -Iseconds)

case $CMD in
    autobackup | backup)
        dump -0 -f "$DST/backup-$DATE" "$SRC"
        echo "Full backup $DATE" >> $DST/log
        ;;
    autorestore | restore)
        cd $SRC
        find $DST -name 'backup-*' -type f -printf '%T@ %p\n' | sort -n
| tail -1 | cut -f2 -d" " | xargs -r -I{} restore -xa -f {}
        cd -
        ;;
```

# 自动备份脚本 maint.sh

```
    init)
        mkdir -p "$DST"
        touch "$DST/log"
        chattr +a "$DST/log"
        ;;
    clean)
        find "$DST" -name 'backup-*' -type f -mtime 30 -delete
        ;;
esac
```

写脚本的意义：

- 可以做成 cron 定时任务
- 封装后用起来简单

# Thank you!