Mounting File Systems Manually with `mount` and Loop Devices

Mounting file systems manually is a core task in Linux system administration. It allows a system to access storage devices or filesystems not automatically mounted at boot time. One powerful but less commonly discussed capability is mounting **loop devices**, which lets you mount disk images as if they were physical devices.

🔄 Loop Devices: Mounting Disk Images

A **loop device** is a pseudo-device that allows a file to be accessed as a block device. This is useful for mounting ISO files or raw disk images.

🔧 Use Case: Mounting an ISO File

1. **Create a mount point:**

```
bash sudo mkdir /mnt/iso
```

1. **Mount using `mount` with `-o loop`:**

```
bash sudo mount -o loop file.iso /mnt/iso
```

- `-o loop` tells the kernel to treat the file as a block device using a loop device.
- Contents of `file.iso` can now be accessed at `/mnt/iso`.
- **Unmount when done:**

```
bash sudo umount /mnt/iso
```

📁 Mounting File Systems Manually

Mounting a USB drive (e.g., `/dev/sdb1`):

```
sudo mkdir /mnt/usb
sudo mount /dev/sdb1 /mnt/usb
```

To **specify file system type** (optional):

```
sudo mount -t vfat /dev/sdb1 /mnt/usb  # For FAT32
```

Unmount:

```
sudo umount /mnt/usb
```

📚 Summary

Task	Command
Mount ISO using loop	`mount -o loop file.iso /mnt/iso`
Mount USB drive	`mount /dev/sdb1 /mnt/usb`
Mount specific partition in `.img`	`mount -o loop,offset=... disk.img /mnt`
Unmount any mount point	`umount /mnt/...`

Loop mounting is particularly useful for:

- Mounting OS images for inspection.
- Testing ISO contents before burning.
- Performing forensic analysis on disk images.