Table of Contents

Part 1 of 2 - Explore all 232+ Linux commands at dargslan.com/learn/linux-commands
Each command includes syntax, options, practical examples with output, and pro tips.

$ blkid

Intermediate

Locate/print block device attributes

blkid displays block device attributes including filesystem type, UUID, label, and partition type. It identifies what type of data is on each partition.

blkid is essential for finding UUIDs needed for /etc/fstab entries. Using UUIDs instead of device names (/dev/sdX) ensures mounts are reliable ...

Options & Flags

`(no args)`	Show all block devices with attributes
`/dev/sdX`	Show attributes for specific device
`-o list`	Show in list format
`-s UUID`	Show only UUID
`-t TYPE=`	Search by attribute

Practical Examples

Example: Show all block devices

```
$ sudo blkid
/dev/sda1: UUID="abc-123" TYPE="ext4" LABEL="root"
/dev/sda2: UUID="def-456" TYPE="swap"
```

Lists all partitions with their UUID, type, and label.

Example: Get UUID for fstab

```
$ sudo blkid -s UUID -o value /dev/sdb1
a1b2c3d4-e5f6-7890-abcd-ef1234567890
```

Shows just the UUID value - perfect for pasting into /etc/fstab.

Example: List format

```
$ sudo blkid -o list
```

Shows device info in a readable table format.

Example: Find swap partitions

```
$ sudo blkid -t TYPE=swap
```

Finds all swap partitions on the system.

Example: Specific device

```
$ sudo blkid /dev/sda1
```

Shows attributes for a specific partition.

Tips & Best Practices

Pro Tip: Use UUIDs in /etc/fstab: UUID=xxx is more reliable than /dev/sdX in fstab. Device names can change; UUIDs never change.

Note: blkid vs lsblk -f: Both show filesystem info. blkid shows more attributes (PARTUUID, etc.). lsblk -f shows the device tree.

Warning: Requires root for all devices: Without root, blkid may not show all devices. Use sudo for complete information.

$ dd

Advanced

Convert and copy data at block level

dd copies and converts data at the block level. It reads from an input source, optionally transforms the data, and writes to an output destination. dd works with raw devices, making it essential for disk imaging, bootable USB creation, and low-level data operations.

dd operates below the filesys...

Options & Flags

`if=`	Input file/device
`of=`	Output file/device
`bs=`	Block size
`count=`	Number of blocks to copy
`status=progress`	Show progress during copy
`conv=`	Conversion options

Practical Examples

Example: Write ISO to USB

```
$ sudo dd if=ubuntu.iso of=/dev/sdb bs=4M status=progress
```

Creates a bootable USB drive from an ISO file. VERIFY /dev/sdb is correct!

Example: Create disk image

```
$ sudo dd if=/dev/sda of=/backup/disk.img bs=4M status=progress
```

Creates a complete byte-for-byte disk image.

Example: Benchmark disk speed

```
$ dd if=/dev/zero of=testfile bs=1M count=1024 status=progress
1073741824 bytes copied, 2.5 s, 429 MB/s
```

Writes 1GB of zeros to measure write speed.

Example: Wipe disk

```
$ sudo dd if=/dev/zero of=/dev/sdb bs=4M status=progress
```

Overwrites entire disk with zeros. DESTRUCTIVE!

Example: Create swap file

```
$ sudo dd if=/dev/zero of=/swapfile bs=1M count=2048
```

Creates a 2GB file filled with zeros for use as swap.

Tips & Best Practices

Warning: VERIFY the target device: dd with wrong of= DESTROYS data permanently. Triple-check with lsblk before running dd to a device. There is no undo.

Pro Tip: Use status=progress: Always add status=progress to see transfer speed and progress. Without it, dd runs silently until complete.

Note: Block size matters: Larger block sizes (bs=4M or bs=1M) are much faster than the default 512 bytes. Always specify bs for better performance.

$ df

Beginner

Report filesystem disk space usage

df (disk free) displays the amount of disk space used and available on mounted filesystems. It shows each mounted filesystem with its total size, used space, available space, usage percentage, and mount point.

df is the primary tool for monitoring disk usage at the filesystem level. It answers t...

Options & Flags

`-h`	Human-readable sizes (KB, MB, GB)
`-T`	Show filesystem type
`-i`	Show inode usage instead of block usage
`-t`	Show only specific filesystem type
`-x`	Exclude filesystem type
`--total`	Show grand total

Practical Examples

Example: Human-readable disk usage

```
$ df -h
Filesystem  Size  Used  Avail  Use%  Mounted on
/dev/sda1   500G  120G  356G   25%   /
/dev/sdb1   1.0T  800G  200G   80%   /data
```

Shows disk usage for all filesystems in human-readable format.

Example: With filesystem type

```
$ df -Th
```

Includes the filesystem type (ext4, xfs, etc.) in the output.

Example: Check inode usage

```
$ df -ih
```

Shows inode usage - important when you have many small files.

Example: Exclude virtual filesystems

```
$ df -h -x tmpfs -x devtmpfs -x squashfs
```

Shows only real disk filesystems.

Example: Check specific path

```
$ df -h /var/log
```

Shows the filesystem containing /var/log.

Tips & Best Practices

Warning: Watch for 90%+ usage: When filesystems reach 90%+ usage, performance degrades and applications may fail. Set up alerts at 80%.

Pro Tip: Inodes can run out too: df -i checks inode usage. You can have free space but no inodes (too many small files). Containers and mail servers are common victims.

Note: df shows mounted filesystems: df only shows mounted filesystems. Unmounted partitions are not visible. Use lsblk to see all block devices.

$ du

Beginner

Estimate file and directory space usage

du (disk usage) estimates file and directory space usage. Unlike df which shows filesystem-level usage, du shows how much space individual directories and files consume.

du is the go-to tool for finding what is consuming disk space. Combined with sort, it quickly identifies the largest directori...

Options & Flags

`-h`	Human-readable sizes
`-s`	Show only total for each argument
`-a`	Show all files, not just directories
`-c`	Show grand total
`--max-depth`	Limit directory depth
`-x`	Stay on same filesystem
`--exclude`	Exclude pattern

Practical Examples

Example: Find largest directories

```
$ du -sh /home/* | sort -rh | head -10
50G\t/home/developer\n12G\t/home/admin\n2G\t/home/guest
```

Shows the 10 largest home directories.

Example: Directory summary

```
$ du -sh /var/log/ /var/www/ /tmp/
2.5G\t/var/log/\n15G\t/var/www/\n500M\t/tmp/
```

Shows total size for each specified directory.

Example: First level breakdown

```
$ du -h --max-depth=1 / 2>/dev/null | sort -rh | head
50G\t/var\n20G\t/home\n5G\t/usr
```

Shows space usage for top-level directories.

Example: Find large files

```
$ du -ah /var/ | sort -rh | head -20
```

Finds the 20 largest files and directories under /var.

Example: Current directory usage

```
$ du -sh .
3.2G\t.
```

Shows total size of the current directory.

Tips & Best Practices

Pro Tip: du -sh * | sort -rh: This one-liner quickly shows the largest items in the current directory, sorted by size. Essential for disk cleanup.

Note: du vs ls -l: du shows actual disk usage (allocated blocks). ls -l shows file size. A sparse file may show 1GB in ls but 0 in du.

Warning: Can be slow: du traverses all files. On large filesystems, du / can take minutes. Use ncdu for interactive, cached exploration.

$ fdisk

Advanced

Partition table manipulator for Linux

fdisk is a dialog-driven partition table manipulator. It creates, deletes, resizes, and manages disk partitions. fdisk supports MBR and GPT partition tables.

fdisk is essential for disk preparation - creating partitions before formatting them with mkfs. It can display current partition layouts, ...

Options & Flags

`-l`	List partition tables for all devices
`-l /dev/sdX`	List partitions for specific device
`/dev/sdX`	Enter interactive mode for device

Practical Examples

Example: List all partitions

```
$ sudo fdisk -l
Disk /dev/sda: 500 GiB
Device      Start       End  Size Type
/dev/sda1    2048   1050623  512M EFI System
/dev/sda2 1050624 976773119  465G Linux filesystem
```

Shows partition tables for all disks.

Example: List specific disk

```
$ sudo fdisk -l /dev/sdb
```

Shows partitions for a specific disk.

Example: Interactive partitioning

```
$ sudo fdisk /dev/sdb
```

Opens interactive mode. Commands: n=new, d=delete, p=print, w=write, q=quit.

Example: Create partition (interactive)

```
$ # In fdisk: n  (new partition) p  (primary) 1  (number)    (default start) +100G  (size) w  (write)
```

Steps to create a 100GB primary partition.

Example: Check partition alignment

```
$ sudo fdisk -l /dev/sda | grep -i sector
```

Verifies partition alignment for optimal performance.

Tips & Best Practices

Warning: Write only when sure: fdisk changes are not applied until you press 'w' (write). Press 'q' to quit without changes. Double-check with 'p' before writing.

Pro Tip: Use parted for GPT: fdisk handles GPT on modern Linux, but parted/gdisk are purpose-built for GPT and offer more features.

Note: Partition after fdisk: After creating a partition with fdisk, format it with mkfs: sudo mkfs.ext4 /dev/sdb1, then mount it.

$ fsck

Advanced

Check and repair a Linux filesystem

fsck (filesystem check) verifies and repairs Linux filesystem integrity. It checks for and fixes errors such as orphaned inodes, bad blocks, incorrect link counts, and directory structure problems.

fsck should only be run on unmounted filesystems or filesystems mounted read-only. Running fsck on...

Options & Flags

`-y`	Automatically answer yes to all repairs
`-n`	No changes - just check
`-f`	Force check even if filesystem seems clean
`-t`	Specify filesystem type
`-p`	Automatically repair safe problems
`-C`	Show progress bar

Practical Examples

Example: Check filesystem

```
$ sudo fsck /dev/sdb1
/dev/sdb1: clean, 1234/65536 files, 45678/262144 blocks
```

Checks the filesystem for errors.

Example: Force check

```
$ sudo fsck -f /dev/sdb1
```

Forces a full check even if the filesystem appears clean.

Example: Auto-repair

```
$ sudo fsck -y /dev/sdb1
```

Automatically fixes all detected errors.

Example: Check without changes

```
$ sudo fsck -n /dev/sdb1
```

Reports errors without making any repairs.

Example: Check root filesystem

```
$ sudo touch /forcefsck && sudo reboot
```

Forces a root filesystem check on next boot.

Tips & Best Practices

Warning: NEVER run on mounted filesystem: Running fsck on a mounted read-write filesystem can cause severe data corruption. Unmount first or boot into single-user mode.

Pro Tip: Journaling reduces need: ext4 and xfs have journaling that recovers from most crash scenarios. fsck is rarely needed on modern systems.

Note: Exit codes: 0=no errors, 1=errors corrected, 2=reboot needed, 4=errors uncorrected, 8=operational error.

$ lvm

Advanced

Logical Volume Manager for flexible disk management

The LVM (Logical Volume Manager) suite provides flexible, dynamic disk management for Linux systems. LVM adds an abstraction layer between physical disks and filesystems, allowing you to resize volumes on the fly, span multiple disks, create snapshots, and manage storage without unmounting or reb...

Options & Flags

`pvcreate`	Initialize a disk/partition for LVM
`vgcreate`	Create a volume group from physical volumes
`lvcreate -L SIZE -n NAME VG`	Create a logical volume
`lvextend -L +SIZE`	Extend a logical volume
`lvreduce -L SIZE`	Shrink a logical volume
`pvs / vgs / lvs`	List PVs, VGs, or LVs with summary
`vgextend`	Add a disk to an existing volume group
`lvcreate --snapshot`	Create a snapshot of a logical volume
`lvcreate --thin`	Create a thin-provisioned volume
`pvmove`	Move data between physical volumes

Practical Examples

Example: Create LVM from scratch

```
$ sudo pvcreate /dev/sdb1 && sudo vgcreate data_vg /dev/sdb1 && sudo lvcreate -L 50G -n data_lv data_vg && sudo mkfs.ext4 /dev/data
```

Complete LVM setup: initialize PV, create VG, create 50GB LV, format with ext4, and mount.

Example: Extend a volume online

```
$ sudo lvextend -L +20G /dev/data_vg/data_lv && sudo resize2fs /dev/data_vg/data_lv
```

Add 20GB to a mounted ext4 volume - no downtime required. Use xfs_growfs for XFS.

Example: Add a new disk to volume group

```
$ sudo pvcreate /dev/sdc1 && sudo vgextend data_vg /dev/sdc1 && sudo lvextend -l +100%FREE /dev/data_vg/data_lv && sudo resize2fs
```

Add a new disk, extend the VG, use all free space to extend the LV, and grow the filesystem.

Example: View LVM status

```
$ sudo pvs && echo "---" && sudo vgs && echo "---" && sudo lvs
```

Quick overview of all physical volumes, volume groups, and logical volumes.

Example: Create LVM snapshot

```
$ sudo lvcreate --snapshot -L 5G -n data_snap /dev/data_vg/data_lv
```

Create a 5GB snapshot of data_lv. The snapshot captures the point-in-time state for backup or testing.

Tips & Best Practices

Pro Tip: Always leave free space in VG: Keep 5-10% of VG space free for snapshots and emergency extensions. Use lvs -o +lv_size,vg_free to check available space.

Warning: Shrinking is dangerous: Shrinking LVs requires unmounting and is risky (data loss if done incorrectly). Always backup first. Shrink filesystem FIRST (resize2fs), then shrink LV (lvreduce). XFS cannot be shrunk.

Note: Growing is always safe: Extending LVs is safe and can be done online (mounted). lvextend + resize2fs (ext4) or xfs_growfs (XFS) with zero downtime.

Pro Tip: LVM thin provisioning: Thin pools allocate space on demand: create a 100GB thin pool, create 500GB of thin LVs - actual disk used is only what data is written. Perfect for development VMs.

$ mkfs

Advanced

Build a filesystem on a device partition

mkfs (make filesystem) creates a filesystem on a disk partition or device. It formats the raw storage so it can store files and directories. mkfs is actually a frontend - the real work is done by type-specific tools like mkfs.ext4, mkfs.xfs, etc.

mkfs is used after partitioning with fdisk to pre...

Options & Flags

`-t`	Specify filesystem type
`.ext4`	Create ext4 filesystem (most common)
`.xfs`	Create XFS filesystem
`.vfat`	Create FAT32 filesystem
`-L`	Set volume label
`-n`	Dry run (some implementations)

Practical Examples

Example: Create ext4 filesystem

```
$ sudo mkfs.ext4 /dev/sdb1
```

Formats partition as ext4 - the most common Linux filesystem.

Example: Create with label

```
$ sudo mkfs.ext4 -L "BackupDisk" /dev/sdb1
```

Creates ext4 filesystem with a volume label.

Example: Create XFS

```
$ sudo mkfs.xfs /dev/sdb1
```

Creates an XFS filesystem - good for large files and high performance.

Example: Create FAT32 for USB

```
$ sudo mkfs.vfat -F 32 /dev/sdc1
```

Creates FAT32 - compatible with Windows, macOS, and Linux.

Example: Full workflow

```
$ sudo fdisk /dev/sdb  # Create partition\nsudo mkfs.ext4 /dev/sdb1\nsudo mount /dev/sdb1 /mnt/data
```

Complete disk setup: partition, format, mount.

Tips & Best Practices

Warning: DESTROYS ALL DATA: mkfs completely erases the partition. Triple-check the device name with lsblk before running. There is no undo.

Pro Tip: ext4 for most Linux use: ext4 is the best default choice - mature, well-supported, good performance, and journaling for reliability.

Note: After mkfs: After formatting, mount the filesystem: sudo mount /dev/sdb1 /mnt/data. Add to /etc/fstab for permanent mount.

$ mount

Intermediate

Mount a filesystem

mount attaches a filesystem to the directory tree. It makes a storage device (disk, partition, USB drive, network share) accessible at a specified directory (mount point).

mount is essential for accessing additional disks, USB drives, network filesystems (NFS, CIFS/SMB), disk images, and remote ...

Options & Flags

`-t`	Specify filesystem type
`-o`	Mount options (ro, rw, noexec, etc.)
`-a`	Mount all filesystems in /etc/fstab
`--bind`	Bind mount (mount directory to another location)
`-l`	Show labels
`-r`	Mount read-only

Practical Examples

Example: Mount a partition

```
$ sudo mount /dev/sdb1 /mnt/data
```

Mounts the partition at /mnt/data.

Example: Mount USB drive

```
$ sudo mount /dev/sdc1 /mnt/usb
```

Mounts a USB flash drive.

Example: Mount read-only

```
$ sudo mount -o ro /dev/sdb1 /mnt/backup
```

Mounts the partition read-only for safety.

Example: Mount NFS share

```
$ sudo mount -t nfs server:/export/data /mnt/nfs
```

Mounts a remote NFS filesystem.

Example: Show mounted filesystems

```
$ mount | column -t
```

Lists all currently mounted filesystems in readable format.

Tips & Best Practices

Pro Tip: Use /etc/fstab for permanent mounts: mount commands are temporary. Add entries to /etc/fstab for mounts that persist across reboots.

Warning: Unmount before removing: Always unmount (umount /mnt/point) before removing a USB drive or disconnecting storage to prevent data loss.

Note: findmnt is better for listing: findmnt shows mounted filesystems in a cleaner tree format than mount. Use findmnt -t ext4 to filter by type.

$ ncdu

Beginner

Interactive disk usage analyzer with ncurses interface

ncdu (NCurses Disk Usage) is an interactive disk usage analyzer with a text-based UI. It scans directories and presents results in a navigable, sorted view, making it easy to find and clean up large files and directories.

ncdu is far more efficient than running du repeatedly. It scans once and l...

Options & Flags

`-q`	Quiet mode (no refresh during scan)
`-x`	Stay on same filesystem
`-e`	Enable shell extension (delete files)
`-o`	Export scan results to file
`-f`	Load previously saved scan
`--exclude`	Exclude pattern

Practical Examples

Example: Scan and explore

```
$ ncdu /
```

Scans the root filesystem and opens interactive browser.

Example: Stay on one filesystem

```
$ ncdu -x /
```

Scans only the root filesystem, excluding mounted drives.

Example: Scan specific directory

```
$ ncdu /var/log
```

Scans and shows usage for the log directory.

Example: Export scan

```
$ ncdu -o /tmp/scan.json /home/
```

Saves scan results to a file for later viewing.

Example: Load saved scan

```
$ ncdu -f /tmp/scan.json
```

Opens a previously saved scan without re-scanning.

Tips & Best Practices

Pro Tip: Navigation keys: Arrows to navigate, Enter to enter directory, d to delete, n to sort by name, s to sort by size, q to quit.

Note: Export for remote servers: Scan a server and save results: ncdu -o- / | gzip > scan.gz. View locally: zcat scan.gz | ncdu -f-

Warning: Not always installed: ncdu is not installed by default. Install with: apt install ncdu, yum install ncdu, or brew install ncdu.

Ready for more? Explore 200+ professional IT eBooks

Go deeper with comprehensive guides, hands-on projects, and real-world examples

dargslan.com/books