

Linux Command Flags Reference Guide

System Commands

File Operations

ls (List Directory Contents)

bash

ls [OPTIONS] [FILE...]

Common Flags:

- **-l** : Long format (detailed information)
- **-a** : Show all files (including hidden files starting with .)
- **-A** : Show all files except . and ..
- **-h** : Human readable sizes (with -l)
- **-R** : Recursive listing
- **-t** : Sort by modification time
- **-r** : Reverse order
- **-S** : Sort by file size
- **-1** : One file per line
- **-d** : List directories themselves, not their contents
- **-i** : Show inode numbers
- **-F** : Add indicators (*/= > @|)
- **-G** : Colorize output
- **--color=auto** : Colorize when output is terminal

Examples:

bash

ls -la # Long format with hidden files

ls -lhS # Long format, human readable, sorted by size

ls -ltr # Long format, all files, sorted by time (newest last)

cp (Copy Files/Directories)

bash

Common Flags:

- `-r, -R` : Copy directories recursively
- `-i` : Interactive (prompt before overwrite)
- `-f` : Force (overwrite without prompt)
- `-v` : Verbose (show files being copied)
- `-p` : Preserve attributes (permissions, timestamps)
- `-a` : Archive mode (equivalent to `-dpR`)
- `-u` : Update (copy only when source is newer)
- `-n` : No-clobber (don't overwrite existing files)
- `-l` : Create hard links instead of copying
- `-s` : Create symbolic links instead of copying
- `-b` : Make backup of existing files
- `-t DIR` : Copy all sources into directory

Examples:

bash

```
cp -r /source/dir /dest/dir  # Copy directory recursively
cp -av file1 file2 /backup/  # Archive copy with verbose output
cp -u *.txt /backup/         # Update copy (only newer files)
```

`mv` (Move/Rename Files)

bash

`mv [OPTIONS] SOURCE DEST`

Common Flags:

- `-i` : Interactive (prompt before overwrite)
- `-f` : Force (overwrite without prompt)
- `-v` : Verbose (show files being moved)
- `-n` : No-clobber (don't overwrite existing files)
- `-u` : Update (move only when source is newer)
- `-b` : Make backup of existing files

- `-t DIR` : Move all sources into directory

Examples:

bash

```
mv -i oldname newname    # Interactive rename
mv -v *.txt /documents/  # Verbose move multiple files
```

`rm` (Remove Files/Directories)

bash

```
rm [OPTIONS] FILE...
```

Common Flags:

- `-r, -R` : Remove directories recursively
- `-f` : Force (ignore nonexistent files, no prompts)
- `-i` : Interactive (prompt before each removal)
- `-I` : Prompt once before removing more than 3 files
- `-v` : Verbose (show files being removed)
- `--preserve-root` : Don't remove / (default)
- `--no-preserve-root` : Don't treat / specially
- `-d` : Remove empty directories

Examples:

bash

```
rm -rf /temp/olddir    # Force remove directory recursively
rm -i *.log             # Interactive removal of log files
rm -v file1 file2       # Verbose removal
```

`mkdir` (Make Directories)

bash

```
mkdir [OPTIONS] DIRECTORY...
```

Common Flags:

- `-p` : Create parent directories as needed
- `-m MODE` : Set permissions (like chmod)

- `-v` : Verbose (show directories being created)

Examples:

```
bash

mkdir -p /path/to/deep/directory # Create nested directories
mkdir -m 755 newdir              # Create with specific permissions
mkdir -pv project/{src,docs,tests} # Create multiple dirs with verbose
```

Process Management

`ps` (Process Status)

```
bash

ps [OPTIONS]
```

Common Flags:

- `aux` : Show all processes for all users
- `-e` : Show all processes
- `-f` : Full format listing
- `-l` : Long format
- `-u USER` : Show processes for specific user
- `-p PID` : Show specific process
- `--forest` : Show process tree
- `-o FORMAT` : Custom output format
- `-C COMMAND` : Show processes running specific command
- `--sort` : Sort by column

Examples:

```
bash

ps aux          # All processes, all users
ps -ef          # All processes, full format
ps -u username  # Processes for specific user
ps --forest     # Process tree view
ps -o pid,ppid,cmd # Custom format
```

`kill` (Terminate Processes)

```
bash
```

```
kill [OPTIONS] PID...
```

Common Flags:

- `-s SIGNAL` : Send specific signal
- `-SIGNAL` : Send signal (shorthand)
- `-l` : List available signals
- `-9` : SIGKILL (force kill)
- `-15` : SIGTERM (graceful termination, default)
- `-1` : SIGHUP (hang up)
- `-2` : SIGINT (interrupt)
- `-STOP` : Stop process
- `-CONT` : Continue stopped process

Examples:

```
bash
```

```
kill 1234      # Graceful termination
```

```
kill -9 1234   # Force kill
```

```
kill -STOP 1234 # Stop process
```

```
kill -l        # List all signals
```

`top` (Dynamic Process Viewer)

```
bash
```

```
top [OPTIONS]
```

Common Flags:

- `-d SECONDS` : Update delay
- `-p PID` : Monitor specific process
- `-u USER` : Monitor specific user
- `-H` : Show threads
- `-i` : Don't show idle processes
- `-c` : Show command line
- `-b` : Batch mode (for scripts)

- `-n NUM` : Number of iterations (with -b)
- `-s` : Secure mode

Interactive Commands (while running):

- `k` : Kill process
- `r` : Renice process
- `M` : Sort by memory usage
- `P` : Sort by CPU usage
- `q` : Quit

`htop` (Enhanced top)

```
bash  
htop [OPTIONS]
```

Common Flags:

- `-d DELAY` : Update delay
- `-u USER` : Show only user's processes
- `-p PID` : Show only specified PIDs
- `-s COLUMN` : Sort by column
- `-t` : Tree view
- `-C` : No color

System Information

`uname` (System Information)

```
bash  
uname [OPTIONS]
```

Common Flags:

- `-a` : All information
- `-s` : Kernel name
- `-n` : Node name (hostname)
- `-r` : Kernel release
- `-v` : Kernel version

- `-m` : Machine architecture
- `-p` : Processor type
- `-i` : Hardware platform
- `-o` : Operating system

Examples:

```
bash
```

```
uname -a    # All system information
```

```
uname -r    # Kernel version only
```

`whoami` (Current User)

```
bash
```

```
whoami
```

No common flags - simply returns current username.

`date` (Display/Set Date)

```
bash
```

```
date [OPTIONS] [+FORMAT]
```

Common Flags:

- `-d STRING` : Display time described by string
- `-f FILE` : Process dates from file
- `-r FILE` : Display last modification time of file
- `-s STRING` : Set time (requires root)
- `-u` : Display UTC time
- `--iso-8601` : ISO 8601 format
- `--rfc-3339` : RFC 3339 format

Format Options:

- `+%Y` : Year (4 digits)
- `+%m` : Month (01-12)
- `+%d` : Day (01-31)
- `+%H` : Hour (00-23)

- `+%M` : Minute (00-59)
- `+%S` : Second (00-60)

Examples:

bash

```
date +%Y-%m-%d      # 2024-07-27
date -d "tomorrow"   # Tomorrow's date
date -r file.txt      # File modification time
```

`uptime` (System Uptime)

bash

```
uptime [OPTIONS]
```

Common Flags:

- `-p` : Pretty format (human readable)
- `-s` : System boot time
- `-h` : Help

Text Processing Commands

Viewing Commands

`cat` (Concatenate and Display)

bash

```
cat [OPTIONS] [FILE...]
```

Common Flags:

- `-n` : Number all lines
- `-b` : Number non-empty lines
- `-s` : Suppress repeated empty lines
- `-E` : Show line endings (\$)
- `-T` : Show tabs as ^I
- `-A` : Show all non-printing characters
- `-v` : Show non-printing characters except tabs and newlines

Examples:

bash

```
cat -n file.txt      # Display with line numbers
cat -A file.txt      # Show all special characters
cat file1 file2      # Concatenate multiple files
```

less (Pager Program)

bash

```
less [OPTIONS] [FILE...]
```

Common Flags:

- `-N` : Show line numbers
- `-S` : Don't wrap long lines
- `-i` : Case-insensitive search
- `-I` : Case-insensitive search (ignore case completely)
- `-F` : Quit if entire file fits on screen
- `-X` : Don't clear screen on exit
- `-r` : Display raw control characters
- `-R` : Display ANSI color escape sequences
- `+G` : Start at end of file
- `+/PATTERN` : Start at first occurrence of pattern

Interactive Commands:

- `/pattern` : Search forward
- `?pattern` : Search backward
- `n` : Next search result
- `N` : Previous search result
- `g` : Go to beginning
- `G` : Go to end
- `q` : Quit

more (Simple Pager)

bash

```
more [OPTIONS] [FILE...]
```

Common Flags:

- `-d` : Display help instead of bell
- `-f` : Count logical lines instead of screen lines
- `-l` : Don't pause after form feed
- `-p` : Don't scroll, clear screen and display
- `-c` : Don't scroll, paint from top
- `-s` : Squeeze multiple blank lines
- `-u` : Suppress underline
- `+NUM` : Start at line number
- `+/STRING` : Start at first occurrence of string

`head` (Display First Lines)

```
bash
```

```
head [OPTIONS] [FILE...]
```

Common Flags:

- `-n NUM` : Show first NUM lines (default 10)
- `-c NUM` : Show first NUM bytes
- `-q` : Never print headers
- `-v` : Always print headers
- `-NUM` : Show first NUM lines (shorthand)

Examples:

```
bash
```

```
head -20 file.txt    # First 20 lines
```

```
head -c 100 file.txt # First 100 bytes
```

```
head -n 5 *.log     # First 5 lines of all log files
```

`tail` (Display Last Lines)

```
bash
```

```
tail [OPTIONS] [FILE...]
```

Common Flags:

- `-n NUM` : Show last NUM lines (default 10)
- `-c NUM` : Show last NUM bytes
- `-f` : Follow file changes (keep reading)
- `-F` : Follow file changes, reopen if file rotated
- `-q` : Never print headers
- `-v` : Always print headers
- `-s SECONDS` : Sleep seconds between iterations with `-f`
- `--pid=PID` : Terminate after process PID dies
- `-NUM` : Show last NUM lines (shorthand)

Examples:

```
bash

tail -f /var/log/syslog  # Follow log file
tail -50 error.log       # Last 50 lines
tail -c 1024 file.bin    # Last 1024 bytes
```

Text Processing

`grep` (Pattern Matching)

```
bash

grep [OPTIONS] PATTERN [FILE...]
```

Common Flags:

- `-i` : Case-insensitive
- `-v` : Invert match (show non-matching lines)
- `-r, -R` : Recursive search in directories
- `-n` : Show line numbers
- `-l` : Show only filenames with matches
- `-L` : Show only filenames without matches
- `-c` : Count matching lines
- `-w` : Match whole words only
- `-x` : Match whole lines only
- `-A NUM` : Show NUM lines after match
- `-B NUM` : Show NUM lines before match

- `-C NUM` : Show NUM lines around match
- `-E` : Extended regex (same as egrep)
- `-F` : Fixed strings (same as fgrep)
- `-o` : Show only matching part
- `--color` : Colorize matches
- `-m NUM` : Stop after NUM matches
- `-s` : Silent mode (suppress errors)
- `-H` : Always print filename
- `-h` : Never print filename

Examples:

bash

```
grep -rn "error" /var/log/      # Recursive search with line numbers
grep -i "warning" *.log        # Case-insensitive in log files
grep -A 5 -B 5 "exception" app.log # Show context around matches
grep -c "ERROR" *.log          # Count matches in each file
```

`sed` (Stream Editor)

bash

```
sed [OPTIONS] 'SCRIPT' [FILE...]
```

Common Flags:

- `-n` : Suppress automatic printing
- `-e SCRIPT` : Add script to commands
- `-f SCRIPT_FILE` : Read script from file
- `-i[SUFFIX]` : Edit files in-place (optional backup)
- `-r, -E` : Use extended regular expressions
- `-s` : Consider files as separate
- `--follow-symlinks` : Follow symbolic links

Common Commands:

- `s/OLD/NEW/` : Substitute first occurrence
- `s/OLD/NEW/g` : Substitute all occurrences
- `d` : Delete line

- `p` : Print line
- `a\TEXT` : Append text
- `i\TEXT` : Insert text
- `c\TEXT` : Change line to text

Examples:

bash

```
sed 's/old/new/g' file.txt      # Replace all 'old' with 'new'
sed -i 's/foo/bar/g' *.txt     # In-place replacement
sed -n '10,20p' file.txt       # Print lines 10-20
sed '/pattern/d' file.txt      # Delete lines matching pattern
```

`awk` (Pattern Processing Language)

bash

```
awk [OPTIONS] 'PROGRAM' [FILE...]
```

Common Flags:

- `-F FS` : Field separator
- `-v VAR=VALUE` : Set variable
- `-f PROGRAM_FILE` : Read program from file
- `-W` : Enable warnings
- `-c` : Use POSIX mode
- `-V` : Print version

Built-in Variables:

- `NR` : Current record number
- `NF` : Number of fields in current record
- `FS` : Field separator
- `RS` : Record separator
- `OFS` : Output field separator
- `ORS` : Output record separator
- `$0` : Entire record
- `$1, $2, ...` : Individual fields

Examples:

bash

```
awk '{print $1}' file.txt      # Print first column
awk -F: '{print $1}' /etc/passwd # Use colon as separator
awk 'NR==5' file.txt          # Print 5th line
awk '{sum+=$1} END {print sum}' # Sum first column
```

sort (Sort Lines)

bash

```
sort [OPTIONS] [FILE...]
```

Common Flags:

- `-r` : Reverse order
- `-n` : Numeric sort
- `-h` : Human-readable numbers (1K, 1M, etc.)
- `-k FIELD` : Sort by specific field/column
- `-t CHAR` : Field separator
- `-u` : Unique (remove duplicates)
- `-f` : Case-insensitive
- `-b` : Ignore leading blanks
- `-d` : Dictionary order
- `-i` : Ignore non-printing characters
- `-M` : Month sort
- `-R` : Random sort
- `-o FILE` : Output to file
- `-c` : Check if already sorted
- `-m` : Merge already sorted files
- `-s` : Stable sort
- `-V` : Version sort

Examples:

bash

```
sort -nr numbers.txt      # Numeric reverse sort
sort -k2 -t: /etc/passwd  # Sort by 2nd field, colon separator
sort -u names.txt         # Sort and remove duplicates
sort -h sizes.txt         # Sort human-readable sizes
```

uniq (Remove Duplicate Lines)

bash

```
uniq [OPTIONS] [INPUT [OUTPUT]]
```

Common Flags:

- `-c` : Count occurrences
- `-d` : Show only duplicate lines
- `-u` : Show only unique lines
- `-i` : Case-insensitive
- `-f NUM` : Skip first NUM fields
- `-s NUM` : Skip first NUM characters
- `-w NUM` : Compare only first NUM characters
- `--group` : Group identical lines

Examples:

bash

```
sort file.txt | uniq -c    # Count occurrences of each line
uniq -d duplicates.txt    # Show only duplicate lines
uniq -u file.txt          # Show only unique lines
```

Network Commands

ping (Network Connectivity Test)

bash

```
ping [OPTIONS] DESTINATION
```

Common Flags:

- `-c COUNT` : Send only COUNT packets
- `-i INTERVAL` : Wait INTERVAL seconds between packets

- `-s SIZE` : Packet size in bytes
- `-t TTL` : Set Time To Live
- `-w TIMEOUT` : Timeout in seconds
- `-W TIMEOUT` : Time to wait for response
- `-f` : Flood ping (requires root)
- `-q` : Quiet output (summary only)
- `-v` : Verbose output
- `-4` : Use IPv4 only
- `-6` : Use IPv6 only
- `-I INTERFACE` : Use specific interface
- `-p PATTERN` : Fill packet with pattern
- `-R` : Record route
- `-a` : Audible ping
- `-n` : Numeric output only (no DNS)
- `-D` : Print timestamps

Examples:

bash

```
ping -c 4 google.com      # Send 4 packets
ping -i 0.5 -c 10 8.8.8.8 # 0.5 second interval, 10 packets
ping -s 1000 example.com  # Use 1000-byte packets
ping -q -c 100 server.com # Quiet mode, 100 packets
```

`curl` (Transfer Data)

bash

```
curl [OPTIONS] [URL...]
```

Common Flags:

- `-o FILE` : Write output to file
- `-O` : Write output to file (use remote filename)
- `-L` : Follow redirects
- `-I` : Head request only (headers)
- `-X METHOD` : HTTP method (GET, POST, PUT, DELETE)
- `-H "HEADER"` : Add custom header

- `-d DATA` : Send data in POST request
- `-F "FIELD=VALUE"` : Send form data
- `-u USER:PASS` : Authentication
- `-k` : Ignore SSL certificate errors
- `-s` : Silent mode
- `-S` : Show errors even in silent mode
- `-v` : Verbose output
- `-w FORMAT` : Custom output format
- `-c FILE` : Cookie jar file
- `-b COOKIES` : Send cookies
- `-A "USER_AGENT"` : Set user agent
- `-e REFERER` : Set referer
- `-T FILE` : Upload file
- `-C -` : Resume transfer
- `--connect-timeout SEC` : Connection timeout
- `-m SEC` : Maximum time for operation
- `-r RANGE` : Download range
- `--limit-rate RATE` : Limit transfer rate
- `-4` : Use IPv4 only
- `-6` : Use IPv6 only
- `--proxy PROXY` : Use proxy
- `-j` : Junk cookies (don't save)

Examples:

bash

```
curl -O https://example.com/file.zip # Download file
curl -L -o page.html https://site.com # Follow redirects, save as page.html
curl -X POST -d "data=value" api.com # POST request with data
curl -H "Authorization: Bearer token" api.com # Custom header
curl -I https://example.com # Get headers only
curl -s -o /dev/null -w "%{http_code}" url # Get status code only
```

wget (Web Downloader)

bash

Common Flags:

- `-O FILE` : Save as filename
- `-P DIR` : Save to directory
- `-c` : Continue partial download
- `-r` : Recursive download
- `-l DEPTH` : Maximum recursion depth
- `-k` : Convert links for local viewing
- `-p` : Download page prerequisites
- `-m` : Mirror (equivalent to `-r -N -l inf --no-remove-listing`)
- `-np` : No parent directories
- `-nH` : Don't create host directories
- `--cut-dirs=NUM` : Ignore NUM directory components
- `-A LIST` : Accept file types
- `-R LIST` : Reject file types
- `-D LIST` : Domain list
- `--exclude-domains=LIST` : Exclude domains
- `-w SECONDS` : Wait between retrievals
- `--random-wait` : Random wait (0.5wait to 1.5wait)
- `--limit-rate=RATE` : Limit download rate
- `-t NUM` : Number of retries
- `-T TIMEOUT` : Read timeout
- `--connect-timeout=SEC` : Connection timeout
- `-q` : Quiet mode
- `-v` : Verbose mode
- `-d` : Debug mode
- `--progress=TYPE` : Progress indicator type
- `-N` : Timestamping (only newer files)
- `--no-cookies` : Don't use cookies
- `--load-cookies=FILE` : Load cookies from file
- `--save-cookies=FILE` : Save cookies to file
- `--header="HEADER"` : Add header

- `-U "USER_AGENT"` : Set user agent
- `--referer=URL` : Set referer
- `--post-data=STRING` : POST data
- `--post-file=FILE` : POST data from file
- `-4` : Use IPv4 only
- `-6` : Use IPv6 only
- `--proxy=on/off` : Use proxy
- `--no-check-certificate` : Don't verify SSL certificates

Examples:

bash

```
wget -c -t 5 https://example.com/large-file.zip # Resume with 5 retries
wget -r -l 2 -k -p https://site.com           # Mirror site, 2 levels deep
wget -P /downloads/ -A "*.pdf" -r site.com    # Download only PDFs
wget --limit-rate=200k https://example.com/file # Limit to 200 KB/s
wget -q --show-progress file.zip             # Quiet with progress bar
```

ssh (Secure Shell)

bash

```
ssh [OPTIONS] [user@]hostname [command]
```

Common Flags:

- `-p PORT` : Connect to port
- `-i FILE` : Use private key file
- `-l USER` : Login as user
- `-o OPTION` : Set option
- `-v` : Verbose mode
- `-q` : Quiet mode
- `-f` : Go to background after authentication
- `-N` : Don't execute remote command
- `-T` : Disable pseudo-terminal allocation
- `-t` : Force pseudo-terminal allocation
- `-X` : Enable X11 forwarding
- `-Y` : Enable trusted X11 forwarding

- `-A` : Enable authentication agent forwarding
- `-g` : Allow remote hosts to connect to forwarded ports
- `-L [bind_address:]port:host:hostport` : Local port forwarding
- `-R [bind_address:]port:host:hostport` : Remote port forwarding
- `-D [bind_address:]port` : SOCKS proxy
- `-C` : Enable compression
- `-c CIPHER` : Select cipher
- `-F FILE` : Use alternative config file
- `-J [user@]host[:port]` : Jump host
- `-W HOST:PORT` : Forward stdin/stdout to host:port
- `-4` : Use IPv4 only
- `-6` : Use IPv6 only
- `-1` : Force protocol version 1
- `-2` : Force protocol version 2

Examples:

bash

```
ssh user@server.com           # Basic connection
ssh -p 2222 user@server.com    # Custom port
ssh -i ~/.ssh/mykey user@server.com # Use specific key
ssh -L 8080:localhost:80 server.com # Local port forwarding
ssh -R 9000:localhost:3000 server.com # Remote port forwarding
ssh -J jumphost user@target    # Through jump host
ssh -X user@server.com         # With X11 forwarding
```