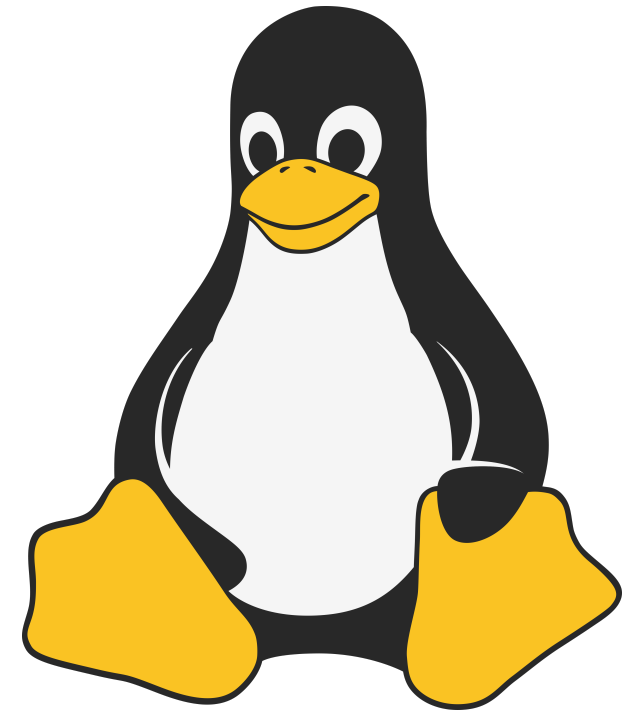


Mustersuche mit **grep**



Zeilen filtern mit `grep`

- Das Kommando `grep <pattern> [file ...]` (global regular expression print) filtert Zeilen aus einer (oder mehreren) Dateien heraus, die das angegebene Muster enthalten.
- Die Angabe eines Dateiarguments ist optional. Wird kein Dateiargument angegeben, wird die Standardeingabe (STDIN) nach dem Muster durchsucht und gefiltert.
- Das Muster kann ein einfacher Text oder ein regulärer Ausdruck sein. (Reguläre Ausdrücke werden in dieser Lektion nicht behandelt.)

Häufig verwendete Optionen

- **-v** (invert match): Zeilen ausgeben, die das Muster nicht enthalten.
- **-i** (ignore case): Groß- und Kleinschreibung ignorieren.
- **-n** (line number): Zeilennummer der Trefferzeile mit ausgeben.
- **-c** (count): Anzahl der Trefferzeilen ausgeben (ohne die Zeilen selbst).
- **-l** Namen der Dateien ausgeben, die mindestens eine Trefferzeile enthalten.
- **-r** (recursive): Rekursiv in den Dateien von Unterverzeichnissen suchen.

- `--color=auto`: Farbige Ausgabe der Treffer (nur bei der Ausgabe im Terminal).

Alias `grep` für `grep --color=auto`

- Alias-Definition: `alias grep='grep --color=auto'`
- Der Alias schaltet die farbige Hervorhebung der Treffer ein, ohne dass die Option `--color=auto` jedes Mal eingetippt werden muss.
- Definieren Sie den Alias und persistieren Sie ihn in der Datei `~/.bash_aliases`, bevor Sie das `grep`-Kommando in den folgenden Übungsbeispielen verwenden.
- Persistieren: `alias grep >> ~/.bash_aliases`

```
hermann@debian:~$ grep hermann /etc/passwd
hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
```

```
hermann@debian:~$ echo $USER
hermann
hermann@debian:~$ grep $USER /etc/passwd # variable substitution
hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
```

```
hermann@debian:~$ id -un
hermann
hermann@debian:~$ grep $(id -un) /etc/passwd # command substitution
hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
```

Ausgabe mit der Ausgabe des `id`-Kommandos zum Vergleich mit den folgenden `grep`-Beispielen:

```
hermann@debian:~$ id
uid=1000(hermann) gid=1000(hermann) Gruppen=1000(hermann),24(cdrom),...,27(sudo),...
```

Das folgende `grep`-Kommado zeigt die Gruppe `hermann` (Gruppenname im 1. Feld) sowie alle Gruppen, in denen der Benutzer `hermann` Mitglied ist (Liste der Benutzernamen einer Gruppe im 4. Feld).

```
hermann@debian:~$ grep $USER /etc/group | nl
 1  cdrom:x:24:hermann
 2  floppy:x:25:hermann
 3  sudo:x:27:hermann,harry,hermine,ron
 4  audio:x:29:pulse,hermann
 5  dip:x:30:hermann
 6  video:x:44:hermann
 7  plugdev:x:46:hermann
 8  users:x:100:hermann,harry,hermine,ron
 9  netdev:x:106:hermann
10  bluetooth:x:111:hermann
11  lpadmin:x:113:hermann
12  scanner:x:116:saned,hermann
13  hermann:x:1000:
```

```
hermann@debian:~$ grep $USER /etc/group | wc -l
13
```


Übungen 1

- Wie lässt sich das `grep`-Kommando aus dem obigen Beispiel so anpassen, dass nur die Primärgruppe des angemeldeten Benutzers ausgegeben wird?
- Wie lässt sich das `grep`-Kommando so anpassen, dass nur die Sekundärgruppen ausgegeben werden, in denen der angemeldete Benutzer Mitglied ist?

(Probieren Sie es aus, bevor Sie die Lösungen auf der nächsten Folie ansehen.)

```
hermann@debian:~$ grep $USER: /etc/group | nl
1  hermann:x:1000:
```

```
hermann@debian:~$ grep $USER /etc/group | grep -v $USER: | nl
1  cdrom:x:24:hermann
2  floppy:x:25:hermann
3  sudo:x:27:hermann,harry,hermine,ron
4  audio:x:29:pulse,hermann
5  dip:x:30:hermann
6  video:x:44:hermann
7  plugdev:x:46:hermann
8  users:x:100:hermann,harry,hermine,ron
9  netdev:x:106:hermann
10 bluetooth:x:111:hermann
11 lpadmin:x:113:hermann
12 scanner:x:116:saned,hermann
```

```
hermann@debian:~$ grep $USER /etc/group | grep -v $USER: | wc -l  
12
```

grep mit mehreren Dateiargumenten

- `grep` kann auch mit mehreren Dateiargumenten verwendet werden.
- In diesem Fall wird der Dateiname vor der Trefferzeile ausgegeben.

```
hermann@debian:~$ grep $USER: /etc/passwd /etc/group
/etc/passwd:hermann:x:1000:1000:Hermann Hueck,,,:/home/hermann:/bin/bash
/etc/group:hermann:x:1000:
```

Weitere **grep**-Beispiele

```
hermann@debian:~$ grep -n $USER: /etc/passwd /etc/group
/etc/passwd:33:hermann:x:1000:1000:Hermann Hueck,,,:/home/hermann:/bin/bash
/etc/group:63:hermann:x:1000:
```

```
hermann@debian:~$ grep -c $USER: /etc/passwd /etc/group
/etc/passwd:1
/etc/group:1
```

```
hermann@debian:~$ grep -c $USER /etc/passwd /etc/group
/etc/passwd:1
/etc/group:10
```

```
hermann@debian:~$ grep -l $USER: /etc/passwd /etc/group  
/etc/passwd  
/etc/group
```

Übungen 2

- Suche in Datei `/etc/fstab` nach Zeilen, die das Muster `uuid` enthalten. Ignoriere (`-i`) dabei die Groß- und Kleinschreibung.
- Suche rekursiv (`-r`) in allen Dateien des Verzeichnisses `/etc/apt` und dessen Unterverzeichnissen nach Zeilen, die das Muster `linux` enthalten. Ignoriere dabei (`-i`) die Groß- und Kleinschreibung.

```
hermann@debian:~$ grep -i uuid /etc/fstab | tr -s ' '
# device; this may be used with UUID= as a more robust way to name devices
UUID=1cbc22fc-4dad-4f2f-9e7d-d58329c86015 / ext4 errors=remount-ro 0 1
UUID=BA7C-C0C4 /boot/efi vfat umask=0077 0 1
UUID=ff32afed-4797-487a-becc-271b43edff14 none swap sw 0 0
```

- `tr -s ' '` ersetzt mehrere aufeinanderfolgende Leerzeichen durch ein einzelnes Leerzeichen.


```
hermann@debian:~$ grep -ir linux /etc/apt
/etc/apt/apt.conf.d/01autoremove: "^firmware-linux.*";
/etc/apt/apt.conf.d/01autoremove: "^linux-firmware$";
/etc/apt/apt.conf.d/01autoremove: "^linux-image-[a-z0-9]*$";
/etc/apt/apt.conf.d/01autoremove: "^linux-image-[a-z0-9]*-[a-z0-9]*$";
/etc/apt/apt.conf.d/01autoremove: "linux-.*";
/etc/apt/sources.list:#deb cdrom:[Debian GNU/Linux 12.7.0 _Bookworm_ ...
```

Mit der Option `-r` wird rekursiv in den Dateien des Verzeichnisses `/etc/apt` und dessen Unterverzeichnissen gesucht.