

[Home](#)

einfach • klar • benutzerfreundlich

Suchen:

[Linux Magazin](#)[Linux User](#)[Easy Linux](#)[Ubuntu User](#)[International](#)[Linux Community](#)[Home / Artikel / ausgabe / 2005 / 01](#)Erschienen in EasyLinux [01/2005](#) » [Ausgabe bestellen](#)**Shell-Arbeiten leicht gemacht**

Befehlsgewaltig

von Heike Jurzik

In der Rubrik Guru-Training und den Shell-Tipps geben wir regelmäßig Hilfestellung zum Arbeiten auf der Shell -- Zeit, das Wissen einmal zu vertiefen und mehr über die Bash selbst zu lernen. Wir zeigen, wie Sie die Kommandozeile effizienter nutzen.

Die Bash vermittelt als Kommandozeileninterpreter zwischen Ihnen und dem Linux-System. Im Klartext heißt das: Sie geben einen Befehl am Prompt ein, und die Bash interpretiert das Kommando und gibt den Auftrag weiter. Darüber hinaus wertet die Bash auch so genannte Platzhalter und Variablen aus, steht also als "Übersetzer" hilfreich zur Seite.

Viele Kommandozeilen-Profis schätzen an der Shell vor allem, dass sie Befehle miteinander kombinieren und Programmausgaben umleiten können. So schauen Sie eine lange Meldungsflut beispielsweise in Betrachtern wie *less* und *more* seitenweise an. Ebenso ist es möglich, ganze Ausgaben durch eine einfache Umleitung in einer anderen Datei abzuspeichern.

Mit welchen Tricks und Kniffen Sie auf der Bash leichter und effizienter arbeiten, zeigen wir in diesem Artikel. Dabei geht es zunächst um vergangene Kommandos und wie Sie die Bash-History durchforsten. Danach zeigen wir Tastaturkombinationen, mit denen Sie Tipparbeit sparen. Zu guter Letzt erfahren Sie, wie Sie Programmausgaben umleiten und Befehle miteinander verketteten.

Vergangenes

Praktischerweise "merkt" sich die Bash, welche Befehle Sie zuletzt am Prompt eingegeben haben. Drücken Sie beispielsweise die Taste [Pfeil hoch], gehen Sie in dieser "History" (wörtlich: Geschichte) zurück, und der zuletzt getippte Befehl erscheint am Prompt. Drücken Sie mehrmals die Taste [Pfeil hoch], wandern Sie weiter zurück; entsprechend gehen Sie mit der Taste [Pfeil runter] wieder nach vorne. Ertönt ein Piepton, haben Sie das Ende der Liste erreicht. Wie lang die History ist, erfahren Sie auf der Kommandozeile, indem Sie den Inhalt der Umgebungsvariable *HISTSIZE* mit Hilfe des *echo*-Befehls anzeigen:

```
$ echo $HISTSIZE
1000
```

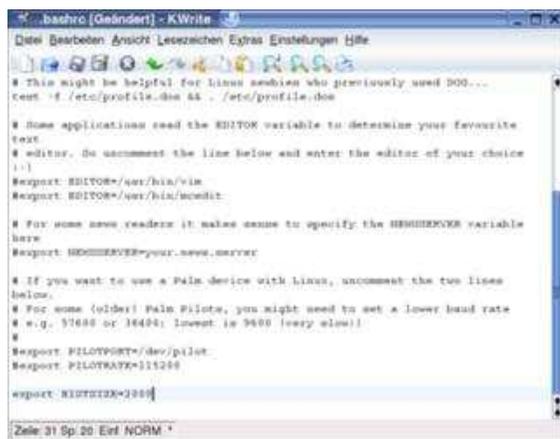
Die Ausgabe verrät: Insgesamt 1000 Einträge können Sie speichern. Wem die Liste nicht lang genug ist, kann durch Neudefinieren der Umgebungsvariable *HISTSIZE* mehr Platz für die History schaffen. Das geht wie folgt:

1. Öffnen Sie mit einem Editor die Datei *.bashrc* in Ihrem Home-Verzeichnis, z. B. über KDEs Schnellstarter [Alt-F2] und Eingabe von

```
kwrite ~/.bashrc
```

2. Gehen Sie bis zum Dateiende und fügen Sie eine neue Zeile ein; um die History auf 2000 Einträge zu vergrößern, lautet der Eintrag:

```
export HISTSIZE=2000
```



3. Speichern Sie die Änderung mit *Datei / Speichern* und verlassen Sie den Editor.
4. Damit die Änderungen greifen, melden Sie sich ab und wieder an.

In welcher Datei die Bash ihre History ablegt, erfahren Sie ebenfalls auf der Kommandozeile:

```
$ echo $HISTFILE
/home/huhn/.bash_history
```

Die Bash-History liegt also in der versteckten Datei *.bash_history* im eigenen Home-Verzeichnis -- dass diese versteckt ist, erkennen

Sie daran, dass ihr Dateiname mit einem Punkt anfängt. Wenn Sie die Datei mit einem Text-Editor betrachten, sehen Sie sämtliche zuletzt ausgeführten Kommandos.

Gesucht, gefunden

Um einen Befehl in der History wiederzufinden, müssen Sie allerdings nicht dauernd hoch- und runterblättern: Die Bash bietet eine eingebaute Suchfunktion. Anstatt so lange [Pfeil hoch] und [Pfeil runter] zu tippen, bis Sie das gesuchte Kommando gefunden haben, rufen Sie über die Tastenkombination [Strg-R] die Befehlssuche auf. Der Prompt verändert sich und zeigt an:

```
(reverse-i-search)` `:
```

Am Doppelpunkt beginnen Sie nun, das gesuchte Kommando einzugeben. Dabei vervollständigt die Shell das Kommando automatisch. Um beispielsweise nach dem letzten `scp`-Befehl zu suchen, geben Sie nur die ersten zwei Buchstaben (`sc`) ein -- schon hat die Bash ein Kommando aufgespürt und zeigt den kompletten Befehl an (Abbildung 1). Sie müssen hier nicht zwingend mit den ersten Buchstaben des Kommandos anfangen; auch mit Teilstücken aus der Mitte des Befehls funktioniert die Suchfunktion problemlos.



Abb. 1: Mit der Tastenkombination [Strg-R] stöbern Sie in der Shell-History und spüren vergangene Befehle auf.

Um den ersten gefundenen Befehl direkt auszuführen, drücken Sie [Eingabe]; alternativ wählen Sie [Esc], wenn Sie das Kommando noch verändern möchten. Der Befehl erscheint dann wie frisch eingetippt am Prompt, und Sie können ihn verändern oder ergänzen. Ist der angezeigte Befehl hingegen noch nicht der richtige Treffer, suchen Sie weiter: Drücken Sie so lange erneut [Strg-R], bis der richtige Befehl auftaucht.

Besser tippen

Neben der Kombination [Strg-R] für das Stöbern in der History hat die Bash noch einige weitere praktische Tastenkombinationen zu bieten, die das Kommandozeilenleben leichter machen. In ein paar Fingerübungen zeigen wir jetzt, wie Sie die Tipparbeit verringern können. Die Tabelle *Tastaturfunktionen in der Bash* zeigt darüber hinaus die wichtigsten Kombinationen.

Dass Sie mit den Tasten [Pfeil hoch] und [Pfeil runter] in der History blättern, wissen Sie schon. Ebenso verwenden Sie die anderen beiden Pfeiltasten, um nach rechts oder links in der Befehlseingabe zu wandern. Da die Entfernung zwischen dem ersten und dem

letzten Zeichen -- gerade bei langen Kommandos -- ziemlich groß ist, bietet die Bash eine Abkürzung, um an den Anfang ([Strg-A]) oder ans Ende ([Strg-E]) der Eingabezeile zu springen. Natürlich geht es auch wortweise nach links ([Alt-B], "B" wie back: zurück) oder rechts ([Alt-F], "F" wie forward: vorwärts).

Zeichen löschen Sie mit den Tasten [Rückschritt] (nach links) und [Entf] (nach rechts). Auch hier gibt es wieder schnellere Wege: [Strg-K] löscht von der aktuellen Position aus bis zum Ende der Zeile, [Strg-U] bis zum Anfang (also bis zum Prompt); ein ganzes Wort nach links löschen Sie mit [Strg-W] und auf manchen Rechnern auch mit [Alt-Rückschritt].

Tutto Completo

Eine besondere Funktion hat die Tabulator-Taste. Mit der so genannten Tab-Completion (auf Deutsch etwa: Vervollständigung mit [Tab]) ersparen Sie sich jede Menge Tipperei, denn die [Tab]-Taste ergänzt Ihre Eingaben. Um mit dem Kommando `ls` beispielsweise den Inhalt des Verzeichnisses `/etc/X11/xserver/` anzuzeigen, geben Sie nicht umständlich jeden Buchstaben ein -- schreiben Sie einfach

```
$ ls /e
```

und drücken Sie dann [Tab]. Die Bash vervollständigt direkt zu `/etc/`. Auch das nächste Verzeichnis ist schon nach dem großen `X` eindeutig und wird mit einem Druck auf [Tab] zu `X11` komplettiert -- am Prompt steht jetzt also:

```
$ ls /etc/X11/
```

Nach Eingabe des nächsten Buchstabens ist das Verzeichnis noch nicht eindeutig -- darum piept es, wenn Sie `x` eingeben und [Tab] drücken. Drücken Sie einfach ein zweites Mal [Tab], erhalten Sie eine Auswahl mit allen Möglichkeiten:

```
$ ls /etc/X11/x
xdm  xinit  xkb  xserver  xsm
```

Geben Sie einfach weitere Buchstaben des Verzeichnisnamens ein -- sobald die Eingabe eindeutig ist, vervollständigt der Druck auf [Tab]. Dieses Feature funktioniert nicht nur mit Pfadnamen: Geben Sie am Prompt einen Buchstaben ein und drücken zweimal [Tab], zeigt die Bash alle Befehle an, die mit diesem Zeichen beginnen. Falls die Liste zu lang ist, fragt die Shell vorsichtshalber nach, ob Sie wirklich alle Kommandos sehen wollen (Abbildung 2).

Eine Sache sollten Sie bei der Verwendung des Größerzeichens zur Umleitung allerdings beachten: Dieser Operator leitet nicht nur in die Datei um, sondern überschreibt eine eventuell vorhandene Datei mit gleichem Namen einfach.

Sicherer ist es, das doppelte Größerzeichen `>>` zu verwenden: Wenn die Zieldatei noch nicht existiert, wird sie angelegt; besteht die Datei schon, hängt die Shell die Programmausgabe einfach an:

```
ls /etc >> etc_inhalt.txt
```

Verkettung von Kommandos

Ein nützliches Shell-Feature ist die Verkettung mehrerer Befehle. Um zwei oder mehr Kommandos nacheinander auszuführen, können Sie diese hintereinander am Prompt aufschreiben und dabei durch ein Semikolon `;` voneinander abtrennen. Erst ganz am Schluss drücken Sie [Eingabe], dann führt die Bash alle Befehle aus. Mit der Kommandoreihe

```
cd Documents; ls > ausgabe.txt; cd -
```

wechseln Sie zunächst in das Unterverzeichnis *Documents*, führen in diesem das Kommando *ls* aus, leiten die Ausgabe in die Datei *ausgabe.txt* um und gehen anschließend mit *cd -* ins vorherige Verzeichnis zurück.

Besonders praktisch ist für eine lange Befehlskette das Programm *sleep*: Es wartet einen genau definierten Zeitraum, bevor es sich selbst beendet -- in der Zwischenzeit tut es einfach nichts, es "schläft". So legen Sie die Shell mit

```
sleep 10
```

für zehn Sekunden schlafen. Eingereicht in eine ganze Kommandoliste sorgt *sleep* für die nötigen Pausen:

```
sleep 15; import -window root screen.png
```

Hier legt *sleep* zunächst eine 15 Sekunden lange Pause ein; danach erstellt das Programm *import* mit den Parametern *-window root* einen Screenshot vom kompletten Bildschirm und speichert diesen (im PNG-Format) in der Datei *screen.png*. So haben Sie genug Zeit, die Fenster auf dem Bildschirm anzuordnen, bevor *import* den Bildschirm knipst.

Pipes

Statt in eine Datei können Sie die Ausgabe eines Kommandos auch direkt ohne Zwischenschritt an einen weiteren Befehl übergeben, dazu verwenden Sie das Pipe-Zeichen (den senkrechten Strich `|`, den Sie über die Tastenkombination [AltGr-<] erreichen). Ist etwa die Ausgabe von *ls* zu lang, so dass Sie aus dem Terminal-Fenster heraus scrollt, leiten Sie das überlange Listing mit dem Pipe-Zeichen an *less* weiter:

```
ls | less
```

Es ist sogar möglich, mehrere dieser Pipes nacheinander zu verwenden -- Ihrer Phantasie sind hier keine Grenzen gesetzt. Wer z. B. alle laufenden Prozesse eines Systems anzeigen (*ps x*), diese Ausgabe direkt nach der Zeichenkette *huhn* durchsuchen und das

Ergebnis wiederum mit `less` betrachten möchte, hängt alles hintereinander:

```
ps x | grep huhn | less
```

Fazit

Auch wenn die Arbeit auf der Kommandozeile für viele Linux-Einsteiger ungewöhnlich und mitunter kompliziert erscheint, erledigen Sie viele Dinge einfach schneller mit der Shell. Wenn Sie die hier gezeigten Funktionen und Fingerübungen ein bisschen trainieren, arbeiten Sie effizienter und lernen die Bash besser kennen -- so werden künftige Artikel in dieser Rubrik besser verständlich und machen mehr Spaß. (hge)

Dieser Online-Artikel kann Links enthalten, die auf nicht mehr vorhandene Seiten verweisen. Wir ändern solche "broken links" nur in wenigen Ausnahmefällen. Der Online-Artikel soll möglichst unverändert der gedruckten Fassung entsprechen.



[Druckerfreundliche Version](#) | [Feedback zu dieser Seite](#) | [Datenschutz](#) | © 2018 [COMPUTEC MEDIA GmbH](#) | Last modified:
2007-04-05 11:10

[Nutzungsbasierte Onlinewerbung](#)

[\[Linux-Magazin\]](#) [\[LinuxUser\]](#) [\[Linux-Community\]](#) [\[Admin-Magazin\]](#) [\[Ubuntu User\]](#) [\[Smart Developer\]](#) [\[Linux Events\]](#) [\[Linux Magazine\]](#) [\[Ubuntu User\]](#) [\[Admin Magazine\]](#) [\[Smart Developer\]](#) [\[Linux Magazine Poland\]](#) [\[Linux Community Poland\]](#) [\[Linux Magazine Brasil\]](#) [\[Linux Magazine Spain\]](#) [\[Linux Technical Review\]](#)