

# Inhaltsverzeichnis

Wochenaufgabe 1:.....	1
Wochenaufgabe 2:.....	2
Wochenaufgabe 4:.....	2
Wochenaufgabe 5:.....	2
Wochenaufgabe 6:.....	2
Wochenaufgabe 7:.....	3
Wochenaufgabe 8:.....	3

## Wochenaufgabe 1:

SSH	Secure shell Einloggen auf einem entfernten Rechner Ssh -p 8080 <u>DEINACCOUNT@hopper.hs-bremerhaven.de</u>
pwd	Print working directory Gibt das aktuelle Arbeitsverzeichnis aus
cd	Wechselt das aktuelle Arbeitsverzeichnis Cd /tmp (relativer Pfad) Cd ../USER/step/ (absoluter Pfad)
mkdir	Make directory Erstelle ein Verzeichnis
rmdir	Leere Verzeichnisse entfernen Rmdir [OPTION] Verzeichnisname
rm	Remove Dateien oder Verzeichnisse entfernen
ls	List Listet den Inhalt des aktuellen Verzeichnisses auf
man	Manual Gibt eine Hilfe zu jedem Linux-Kommando
cat	Concatenate Gib den Inhalt der Datei aus Cat /home/step2022/data/user.txt
tree	Tree Gibt den Verzeichnisbaum des aktuellen oder des übergebenen Verzeichnisses grafisch aus
sort	Sort Gibt die angegebene Datei sortiert aus. Sort -n datei.txt (-n steht für number) Sortiere numerisch Sort -k 3 datei.txt (-k steht für key) Sortiere alphanumerisch nach der 3. Spalte
grep	Sucht und markiert Zeilen, die das Muster erhalten. Grep „robert“ data.txt
>	Ausgabeumleitung in Datei Die Ausgabe eines jeden Programmes, das etwas ausgibt, kann umgeleitet werden in eine Datei. ls > list.txt  >> Datei umlenken ohne sie zu überschreiben.

## Wochenaufgabe 2:

head	gibt die ersten 10 Zeilen einer Datei aus. (Standardmäßig)
tail	gibt die letzten 10 Zeilen einer Datei aus. (Standardmäßig)
cut	schneidet Felder (oder Spalten) aus allen Zeilen aus
nl	Numberline fügt Zeilennummern am Anfang jeder Zeile hinzu nl /home/step2022/data/user.txt
	Pipe fügt die Ausgabe des einen Programmes in die Eingabe des anderen Programmes ein cat datei.txt   cut -d " " -f 1,2,4   grep " stud "   sort -k 1 > erg.txt
bash	Skript ausführen
chmod	Skript ausführbar machen
uniq	Filter alle doppelten Zeilen unter der Bedingung, dass die Zeilen sortiert sind.

## Wochenaufgabe 4:

echo	Schreibt die übergebenen Argumente auf Standard-Out.
sed	Streaming Editor nimmt als Argument eine Anweisung, die oft die Form hat: 's/suchmuster/ersetzung/g'
tidy	tidy HTML-Dateien prüfen auf Korrektheit

## Wochenaufgabe 5:

read	Das Programm read liest eine Eingabezeile aus der Standardeingabe und verteilt sie wortweise auf die angegebenen Variablen
------	--

## Wochenaufgabe 6:

date	gibt das aktuelle Datum aus.
------	------------------------------

## Wochenaufgabe 7:

Wildcards vs. RegEx	
Wildcards sind weniger mächtig, dafür aber etwas natürlicher zu formulieren und finden sich in der Bash, in Datenbanken und bei Suchmaschinen. Lest Euch die Seite <a href="https://linuxhint.com/bash_wildcard_tutorial/">https://linuxhint.com/bash_wildcard_tutorial/</a> durch.	Regex (reguläre Ausdrücke) sind ein sehr mächtiges Konzept, das in vielen Bereichen der Informatik benötigt wird. Allerdings ist es nicht immer ganz so leicht, den richtigen Ausdruck zu finden.
<pre>echo /home/step2022/aufgaben/*.txt ls /home/step2022/aufgaben/*.txt ls /home/step2022/aufgaben/   grep 'aufgabe-[0-9]*\.txt'</pre>	

## Wochenaufgabe 8:

Wir haben dieses Mal wieder die vorige Wochenaufgabe ausführlich besprochen und uns mit den unterschiedlichen Strategien zum Schreiben in Dateien beschäftigt.		
Wenn wir direkt im Homeverzeichnis schreiben, dann können bestimmte Operationen schon einmal langsamer sein als andere. Beachtet den Unterschied!		
<pre>cat datei.txt   \ while read line; do echo "\$line" &gt;&gt; ziel.txt done</pre>	<pre>cat datei.txt   \ while read line; do echo "\$line" done &gt;&gt; ziel.txt</pre>	
Zudem ist es ein Unterschied, ob ihr ...		
<p>ein Shellprogramm mit den imperativen Elementen (while, if, echo ...) schreibt,</p> <pre>cat datei.txt   \ while read felda feldb; do echo "\$felda" done</pre>	<p>ein passendes Programm findet (grep, sed, ...), bzw. eine Kombination davon, die durch die Pipe verknüpft sind</p> <pre>cut -d ' ' -f1 datei.txt</pre> <p>bzw.</p> <pre>cat datei.txt   \ cut -d ' ' -f1 datei.txt</pre>	<p>oder in jedem Schleifendurchlauf ein weiteres Programm aufruft, dessen Ergebnis Ihr beispielsweise durch Kommandosubstitution in eine Variable übergebt.</p> <pre>cat datei.txt   \ while read line; do feld=\$(echo "\$line"   cut -d ' ' -f1) echo "\$feld" done</pre>
Dazu gab es heute wieder zwei neue Programme und ein neues Konzept:		
<p>tr translate/delete</p> <p>die Folge des ersten Argumentes wird jeweils in die Folge des zweiten Argumentes übersetzt:</p> <pre>echo "abc"   tr 'ab' 'AB' =&gt; ABc echo "abcd"   tr 'a-z' 'A-Z' =&gt; ABCD</pre> <p>Schaut in jedem Fall in die Man-Page und findet heraus, wie Ihr mit tr einzelne Zeichen löscht und dann das Komplement (alle Zeichen, die nicht in der aufgezählten Menge sind) löscht.</p> <p>tr ist immer dann notwendig, wenn der Zeilenumbruch ("\n") ersetzt werden soll - das geht mit sed nämlich nicht.</p>	<p>Brace Expansion</p> <p>In der Bash lassen sich als Kurzform von \$(seq 10) mit geschweiften Klammern Produktionen erzeugen:</p> <pre>echo {1..9} =&gt; 1 2 3 4 5 6 7 8 9</pre> <p>Das werdet Ihr im Internet in vielen Beispielen finden. Zusätzlich lassen sich damit kombinierte Wörter erzeugen</p> <pre>echo {0..1}{0..1}{0..1}{0..1} 0000 0001 0010 0011 0100 ...</pre> <p>Schaut in der Man-Page zur bash unter dem Abschnitt brace-Expansion nach. Oft benötigt man Zahlenfolgen mit führendem Nullen:</p> <pre>echo {001..100} 001 002 ...</pre> <pre>seq -w 1 100 001 002</pre>	<p>Curl</p> <p>"there is nothing that curl can't do ..."</p> <p>Letztlich ist curl ein Programm, um den Inhalt einer URL im Web abzurufen oder Dateien hochzuladen (neben all den anderen Protokollen). Damit lassen sich Automatisierungen bis hin zu Lasttests und Korrektheitstest für Webseiten bauen.</p> <pre>curl https://informatik.hs-bremerhaven.de/step2022/stundenplan/data/step1.csv</pre>