

Titel: Labor 3

Klasse: 3BHIF

Name: Haiden

Gruppe: 01

Aufgabe: 13.11.2019

Abgabe: 27.11.2019

Inhaltsverzeichnis

1	Head & Tail	1
1.1	Aufgabe 8.8 – Gezielt die 13.Zeile ausgeben	1
1.2	Aufgabe 8.9	1
2	Sort & Uniq	1
2.1	Aufgabe 8.12 – Sortieren der Teilnehmer0.dat.....	1
2.2	Aufgabe 8.13	2
2.3	Aufgabe 8.15	2
3	Cut & Paste	3
3.1	Aufgabe 8.16	3
3.2	Aufgabe 8.17	3
3.3	Aufgabe 8.18	4
3.4	Aufgabe 8.19	5
4	Diverse Fragen	5
4.1	Welche Bedeutung hat die Umgebungsvariable PATH	5
4.2	Unterscheidet sich die PATH Variable für die User schueler bzw.root?	5
4.3	Welche Bedeutung haben die Variablen UID, PWD, PS1, HOME?	5
4.3.1	UID	5
4.3.2	PWD	5
4.3.3	PS1	5
4.3.4	HOME	5
4.4	Welche Variablen sind gesetzt?	5
4.5	Wie können Sie eine Umgebungsvariable setzen?	6
4.6	Rufen Sie mit dem Befehl „bash“ eine „Kindshell auf. Ist ihre Umgebungsvariable noch gesetzt? Wie können Sie das erreichen?.....	6
4.7	Aufgabe 9.4 – Eltern und Kind-Shell.....	6
4.8	Welche Aufgabe haben die Befehle type, which und whereis (Testen!)?	7
4.8.1	TYPE	7
4.8.2	WHICH	7
4.8.3	WHERE IS	7

1 Head & Tail

1.1 Aufgabe 8.8 – Gezielt die 13. Zeile ausgeben



8.8 [!2] Wie würden Sie gezielt nur die 13. Zeile der Eingabe ausgeben?

```
schueler@Debian10nvs:~$ tail +13 gesamt.txt | head -1  
/usr/share/doc/debian/bug-mailserver-refcard.txt
```

1.2 Aufgabe 8.9



8.9 [3] Probieren Sie »tail -f« aus: Legen Sie eine Datei an und rufen Sie »tail -f« für diese Datei auf. Hängen Sie dann in einem anderen Fenster bzw. auf einer anderen virtuellen Konsole z. B. mit »echo >>« etwas an die Datei an und beobachten Sie die tail-Ausgabe. Wie sieht das ganze aus, wenn tail mehrere Dateien gleichzeitig beobachtet?

```
schueler@Debian10nvs:~/3b$ touch myfile  
schueler@Debian10nvs:~/3b$ tail -f myfile  
Hello World  
#Andere Konsole  
schueler@Debian10nvs:~/3b$ echo „Hello World“ >> myfile
```

2 Sort & Uniq

2.1 Aufgabe 8.12 – Sortieren der Teilnehmer0.dat



8.12 [!2] Sortieren Sie die Teilnehmerliste in `teilnehmer0.dat` (der Datei mit Doppelpunkten als Feldtrenner) nach den Vereinsnamen und innerhalb der Vereine nach den Nach- und Vornamen der Spieler (in dieser Reihenfolge).

```
schueler@Debian10nvs:~/3b$ sort -t: -k2,2 -k1,1 teilnehmer0.dat  
Flöttmann Fritz:Sportfreunde Renntal:217:Herren  
Fluschinsky Käthe:Sportfreunde Renntal:57:Damen  
Schwitz Sieglinde:Sportfreunde Renntal:93:Damen  
Langbein Leni:SV Schnaufenberg:55:Damen  
Schulz Hugo:SV Schnaufenberg:123:Herren  
Rasbichel Katja:TSV Lahmhausen:119:Damen  
Rasbichel Ulla:TSV Lahmhausen:117:Damen  
Schleicher Detlef:TSV Lahmhausen:13:Herren  
Springinsfeld Karlheinz:TV Jahnstein:154:Herren  
von Traben Gesine:TV Jahnstein:26:Damen
```

```
Zielinger Hannes:TV Jahnstein:45:Herren
```

2.2 Aufgabe 8.13



8.13 [3] Wie können Sie die Teilnehmerliste aufsteigend nach den Vereinsnamen und innerhalb der Vereine absteigend nach der Startnummer sortieren?
(Tipp: Dokumentation lesen!)



```
schueler@Debian10nvs:~/3b$ sort -t: -k2,2 -k3,3nr teilnehmer0.dat
Flöttmann Fritz:Sportfreunde Renntal:217:Herren
Schwitz Sieglinde:Sportfreunde Renntal:93:Damen
Fluschinsky Käthe:Sportfreunde Renntal:57:Damen
Schulz Hugo:SV Schnaufenberg:123:Herren
Langbein Leni:SV Schnaufenberg:55:Damen
Rasbichel Katja:TSV Lahmhausen:119:Damen
Rasbichel Ulla:TSV Lahmhausen:117:Damen
Schleicher Detlef:TSV Lahmhausen:13:Herren
Springinsfeld Karlheinz:TV Jahnstein:154:Herren
Zielinger Hannes:TV Jahnstein:45:Herren
von Traben Gesine:TV Jahnstein:26:Damen
```

2.3 Aufgabe 8.15



8.15 [2] Ein Verzeichnis enthält Dateien mit den folgenden Namen:

```
01-2002.txt 01-2003.txt 02-2002.txt 02-2003.txt
03-2002.txt 03-2003.txt 04-2002.txt 04-2003.txt
<<<<<<
11-2002.txt 11-2003.txt 12-2002.txt 12-2003.txt
```

Geben Sie ein `sort`-Kommando an, mit dem Sie die Ausgabe von `ls` in die »chronologisch richtige« Reihenfolge

```
01-2002.txt
02-2002.txt
<<<<<<
12-2002.txt
01-2003.txt
<<<<<<
12-2003.txt
```

bringen können.

```
schueler@Debian10nvs:~/3b/xxx$ ls | sort -t- -k2,2n -k1,1n
01-2010.txt
02-2010.txt
03-2010.txt
04-2010.txt
```

```
05-2010.txt
06-2010.txt
07-2010.txt
08-2010.txt
09-2010.txt
10-2010.txt
11-2010.txt
12-2010.txt
01-2011.txt
...
```

3 Cut & Paste

3.1 Aufgabe 8.16



8.16 [!2] Generieren Sie eine neue Version der Datei `teilnehmer.dat` (der mit der festen Spaltenbreite), in der die Startnummer und die Vereinszugehörigkeit nicht auftauchen.



```
schueler@Debian10nvs:~/3b$ cut -c1-24,50-60 teilnehmer.dat

Schulz Hugo                Herren
Schleicher Detlef          Herren
Flöttmann Fritz            Herren
Springinsfeld Karlheinz    Herren
von Traben Gesine          Damen
Rasbichel Ulla             Damen
Schwitz Sieglinde          Damen
Rasbichel Katja            Damen
Langbein Leni              Damen
Zielinger Hannes           Herren
Fluschinsky Käthe          Damen
```

3.2 Aufgabe 8.17



8.17 [!2] Generieren Sie eine neue Version der Datei `teilnehmer0.dat` (der mit den durch Doppelpunkt getrennten Feldern), in der die Startnummer und die Vereinszugehörigkeit nicht auftauchen.

```
schueler@Debian10nvs:~/3b$ cut -d: -f1,4 teilnehmer0.dat
Schulz Hugo:Herren
Schleicher Detlef:Herren
Flöttmann Fritz:Herren
Springinsfeld Karlheinz:Herren
von Traben Gesine:Damen
Rasbichel Ulla:Damen
Schwitz Sieglinde:Damen
Rasbichel Katja:Damen
Langbein Leni:Damen
Zielinger Hannes:Herren
Fluschinsky Käthe:Damen
```

3.3 Aufgabe 8.18



8.18 [3] Erzeugen Sie eine Version der Datei `teilnehmer0.dat`, bei der die Felder nicht durch Doppelpunkte, sondern durch die Zeichenkette `»,«` (Komma gefolgt von einem Leerzeichen) getrennt sind.

```
schueler@Debian10nvs:~/3b$ tr ":" ", \" < teilnehmer0.dat > new1.dat
schueler@Debian10nvs:~/3b$ cat new1.dat
Schulz Hugo,SV Schnaufenberg,123,Herren
Schleicher Detlef,TSV Lahmhausen,13,Herren
Flöttmann Fritz,Sportfreunde Renntal,217,Herren
Springinsfeld Karlheinz,TV Jahnstein,154,Herren
von Traben Gesine,TV Jahnstein,26,Damen
Rasbichel Ulla,TSV Lahmhausen,117,Damen
Schwitz Sieglinde,Sportfreunde Renntal,93,Damen
Rasbichel Katja,TSV Lahmhausen,119,Damen
Langbein Leni,SV Schnaufenberg,55,Damen
Zielinger Hannes,TV Jahnstein,45,Herren
Fluschinsky Käthe,Sportfreunde Renntal,57,Damen
```

3.4 Aufgabe 8.19



8.19 [3] Wie viele verschiedene Gruppen werden von Benutzern auf Ihrem System als primäre Gruppen benutzt? (Die primäre Gruppe eines Benutzers ist das vierte Feld in der Datei `/etc/passwd`.)

```
schueler@Debian10nvs:~$ cut -d: -f4 /etc/passwd | sort -k1,1n | uniq  
| wc -l
```

30 # Ausgabe des Befehls

4 Diverse Fragen

4.1 Welche Bedeutung hat die Umgebungsvariable PATH

In ihr stehen die Verzeichnisse, welche die ausführbaren Programme beinhalten. Sie wird von hinten nach vorne durchsucht.

4.2 Unterscheidet sich die PATH Variable für die User schueler bzw.root?

Ja, in der PATH-Variable von Root stehen Pfade für Systemprogramme, die ein normaler User nicht ausführen kann.

4.3 Welche Bedeutung haben die Variablen UID, PWD, PS1, HOME?

4.3.1 UID

Das ist der User Identifier, der jeden Benutzer mit einer einzigartigen Zahl identifiziert.

4.3.2 PWD

Hier steht der aktuelle Ordner, in dem man sich befindet.

4.3.3 PS1

Dies bezeichnet die Formatierung des Prompts, also z.B. `schueler@debian10nvs`.

4.3.4 HOME

Hier steht der absolute Verzeichnispfad vom Home-Verzeichnis des aktuell in der Shell angemeldeten Benutzers.

4.4 Welche Variablen sind gesetzt?

Mit `printenv` kann man herausfinden, welche Umgebungsvariablen gesetzt wurden bzw. gesetzt sind.

```
schueler@Debian10nvs:~$ printenv  
  
SHELL=/bin/bash  
  
SESSION_MANAGER=local/Debian10nvs:@/tmp/.ICE-unix/865,unix/Debian10nvs:/tmp/.ICE-unix/865  
  
WINDOWID=35651587  
  
...
```

4.5 Wie können Sie eine Umgebungsvariable setzen?

Mit dem Befehl `export NEWVAR=NEWVALUE` setzt man eine neue Variable. Achtung, zwischen Variablenname und Wert, der gesetzt werden soll, darf KEIN Leerzeichen sein.

Beispiel:

```
schueler@Debian10nvs:~$ export Name=huber
schueler@Debian10nvs:~$ echo $Name
huber
```

4.6 Rufen Sie mit dem Befehl „bash“ eine „Kindshell“ auf. Ist ihre Umgebungsvariable noch gesetzt? Wie können Sie das erreichen?

```
schueler@Debian10nvs:~$ export Name=huber
schueler@Debian10nvs:~$ echo $Name
huber
schueler@Debian10nvs:~$ bash
schueler@Debian10nvs:~$ echo $Name
huber
schueler@Debian10nvs:~$
```

Wie man am obigen Beispiel sehen kann, bleibt die Umgebungsvariable in der Kind-Shell erhalten. Möchte man dies erreichen, muss man eine Umgebungsvariable statt einer normalen Variable setzen, da nur Umgebungsvariablen vererbt werden.

4.7 Aufgabe 9.4 – Eltern und Kind-Shell



9.4 [!2] Was passiert, wenn Sie eine Umgebungsvariable im Kindprozess ändern? Betrachten Sie die folgende Kommandosequenz:

<code>\$ export bla=fasel</code>	<i>bla ist Umgebungsvariable</i>
<code>\$ bash</code>	<i>Neue Shell (Kindprozess)</i>
<code>\$ echo \$bla</code>	
<code>fasel</code>	<i>Umgebungsvariable wurde vererbt</i>
<code>\$ bla=blubb</code>	<i>Neuer Wert</i>
<code>\$ exit</code>	<i>Zurück in die Elter-Shell</i>
<code>\$ echo \$bla</code>	<i>Was kommt hier heraus??</i>

```
schueler@Debian10nvs:~$ export bla=fasel
schueler@Debian10nvs:~$ bash
schueler@Debian10nvs:~$ echo $bla
fasel
schueler@Debian10nvs:~$ bla=blubb
schueler@Debian10nvs:~$ echo $bla
blubb
schueler@Debian10nvs:~$ exit
```



```
exit
schueler@Debian10nvs:~$ echo $bla
fasel
schueler@Debian10nvs:~$
```

Wenn man die Umgebungsvariable im Kind-Prozess verändert, ändert sich der Wert nur in der geerbten Variable, allerdings nicht in der Eltern-Shell, wie man an der obigen, beispielhaften Terminalausgabe sehen kann.

4.8 Welche Aufgabe haben die Befehle type, which und whereis (Testen!)?

4.8.1 TYPE

Mit dem Type-Befehl kann man herausfinden, um was es sich für eine Art Befehl handelt, d.h. ob er in der Shell eingebaut ist oder als Binärprogramm vorliegt.

```
root@Debian9:~# type cd
cd is a shell builtin
root@Debian9:~# type vi
vi is /usr/bin/vi
```

4.8.2 WHICH

Mit dem Befehl which kann man herausfinden, in welchem Ordner das Binärprogramm liegt. Der Befehl sucht in den Ordnern, welche in der PATH Variable festgelegt sind, nach dem Programm.

```
root@Debian9:~# which cd
root@Debian9:~# which cp
/bin/cp
root@Debian9:~# which ls
/bin/ls
root@Debian9:~#
```

4.8.3 WHERE IS

Mit dem Where-Is Befehl kann man herausfinden, wo sich der Speicherort des Binärprogramms sowie seine Dokumentation in der Form einer Man-Page findet.

```
root@Debian9:~# whereis cp
cp: /bin/cp /usr/share/man/man1/cp.1.gz
root@Debian9:~# whereis nano
nano: /bin/nano /usr/share/nano /usr/share/man/man1/nano.1.gz
/usr/share/info/nano.info.gz
root@Debian9:~#
```