## Verzeichnisse und Kommandos

Der BASH-Prompt ist wie folgt aufgebaut: *Benutzername@Hostname:$ORT*, wobei *$ORT* das aktuelle Arbeitsverzeichnis ist. In diesem Fall ist es ~, was eine Abkürzung für das Homeverzeichnis darstellt.

Mit dem Befehl:

`pwd`

kannst du dir jederzeit den genauen Pfad ausgeben lassen, in dem du dich befindest. **pwd** steht bezeichnenderweise für **p**rint **w**orking **d**irectory.

Wenn du **pwd** im Homeverzeichnis des Benutzers root eingibst, sollte die Ausgabe wie folgt aussehen:

**/root**

### ls

Mit dem Befehl:

`ls`

kannst du dir den Verzeichnisinhalt ausgeben lassen. Da das Heimatverzeichnis des Benutzers **root** bisher leer ist, wird keine Ausgabe erscheinen. Doch aufgepasst: Es gibt unter Linux sogenannte versteckte Dateien. Diese werden mit einem Punkt vorangestellt. Ein einfaches **ls** zeigt die versteckten Dateien nicht an.

Bevor es weitergeht, melde dich bitte mithilfe des `logout` Kommandos von der aktuellen *tty* ab und am Login-Prompt erneut als Benutzer **root** mit dem von dir vergebenen Passwort an.

Versuche es nun noch einmal, aber diesmal mit dem Befehl:

`ls -al`

Die Ausgabe sollte dabei wie folgt aussehen:

```
root@darkstar:~# ls -al
total 24
drwx--x---  3 root root 4096 Mar 10 11:23 ./
drwxr-xr-x 22 root root 4096 Mar  9 10:01 ../
-rw-------  1 root root   20 Mar 10 10:16 .bash_history
drwx------  3 root root 4096 Mar 10 10:18 .cache/
-rw-r--r--  1 root root  886 Mar 10 11:21 .xinitrc
-rwxr-xr-x  1 root root  886 Mar 10 11:21 .xsession*
root@darkstar:~# /etc/rc.d/rc.gpm stop
Stopping gpm...
root@darkstar:~#
```

Doch woher kommen diese Dateien und was bedeuten sie? Und was hat es mit den Zahlen und Zeichen auf sich?

## Dateiberechtigungen

Schauen wir uns einmal eine der Dateien an:

> `-rw-------` `1 root root 20 Mar 10 10:16 .bash_history`

Der erste Bereich `-rw-------` gibt die Rechte der Datei an. Diese werden von Linux in Oktalform verwaltet (4-2-1). Das hört sich erst einmal kompliziert an, ist aber ganz einfach, wenn du es verstanden hast. Das allererste `-` gibt den Typ an. In diesem Falle ist er nicht speziell definiert worden. Würde es sich um ein Verzeichnis handeln, würde dort *d* für Directory stehen.

Danach folgt ein `rw-`. Es gibt drei Blöcke, mit jeweils drei Werten. Die Blöcke stehen für **Owner** / **Group** und **Others**, also dem Dateieigentümer, der Gruppe und allen Anderen. Die drei Werte innerhalb der Blöcke stehen für:

- r = read (lesen) – Oktal: 4
- w = write (schreiben) – Oktal: 2
- x = execute (ausführen) – Oktal: 1

Für die Datei *.bash_history* sind nur die Werte `r` und `w` für den *Owner* definiert worden: `-rw-------`. Der Eigentümer der Datei kann diese also lesen und schreiben:

**Owner**: `rw-` = 4+2+0 = 6 / **Group**: `---`: 0+0+0 = 0 / **Others**: `---`: 0+0+0 = 0 oder zusammengesetzt: **600**

Falls der Eigentümer die Datei auch ausführen können soll, sähe die Definition wie folgt aus:

> `-rwx------`

Oktal wäre dies **Owner**: `rwx` = 4+2+1 = 7 / **Group**: `---`: 0+0+0 = 0 / **Others**: `---`: 0+0+0 = 0 oder zusammengesetzt: **700**.

Kommen wir nochmals auf das Beispiel zurück:

> `-rw-------` `1` **`root root`** `Mar 10 10:16 .bash_history`

Der Eigentümer und die Gruppe sind beide **root root**. Falls auch die Gruppe und alle Anderen lesend auf die Datei zugreifen können soll, würden die Rechte wie folgt dargestellt: `-rw-r--r--`.

Oktal wäre dies **Owner**: `rw-` = 4+2+0 = 6 / **Group**: `r--`: 4+0+0 = 4 / **Others**: `r--`: 4+0+0 = 4 oder zusammengesetzt: **644**.

Diese Rechte könntest du testweise mit folgendem Befehl ändern:

```
chmod 644 .bash_history
```

Wenn du nun nochmals ein **ls -al** eingibst, könnte die Ausgabe jetzt wie folgt aussehen:

> `-rw-r--r--` `1 root root 20 Mar 10 10:16 .bash_history`

Ändere die Rechte der Datei daraufhin bitte wieder zurück auf den Ursprungswert:

```
chmod 600 .bash_history
```

Du hast wahrscheinlich beobachtet, dass sich der Zahlenwert hinter der *Owner* und *Group* Definition erhöht hat, ebenso die Änderungszeit. Das hat damit zu tun, dass die History Datei in der Zwischenzeit angewachsen ist.

> `-rw-r--r-- 1 root root` **`64`** `Mar 10` **`10:20`** `.bash_history`

## History

Die versteckte Datei **.bash_history** enthält eine Historie der vom Benutzer eingegebenen Befehle. Nach dem ersten Login musstest du dich einmal abmelden und wieder anmelden, damit diese erzeugt worden ist. Ausgeben lassen kannst du sie dir einfach mit dem Befehl:

```
history
```

Dort sollte auch deine Eingabe von **pwd** mit einer Nummer davor aufgeführt werden. Möchtest du dieses Kommando erneut ausführen, kannst du einfach **!$NUMMER** eingeben, also das Ausrufezeichen, direkt gefolgt von der angegebenen Nummer, zum Beispiel **!3**, wobei du die 3 natürlich durch die entsprechende Nummer ersetzen musst.

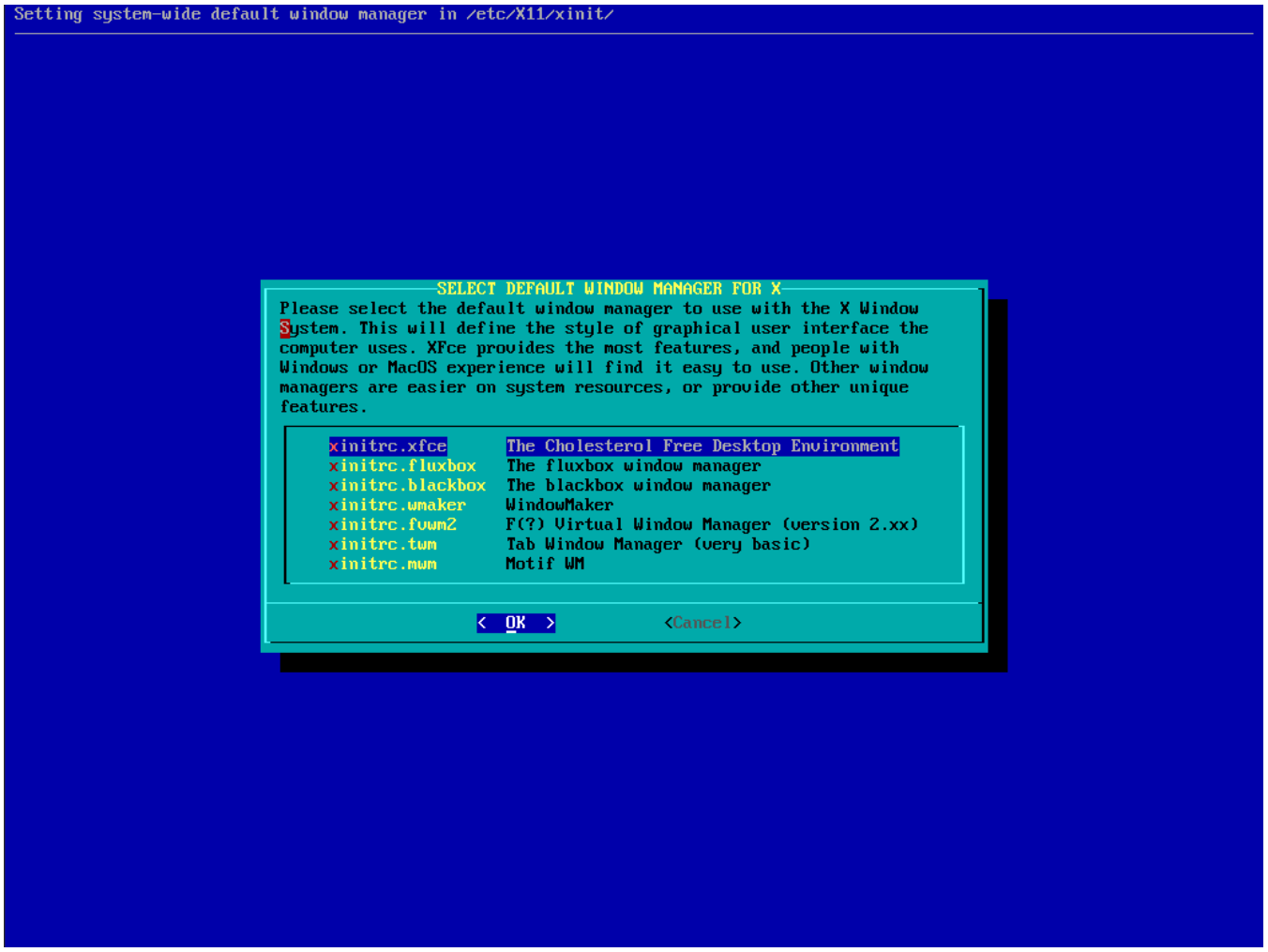
Du kannst in der BASH auch ganz einfach mit den Pfeil-hoch, Pfeil-runter Tasten die History durchblättern. Enter löst dann jeweils den gefundenen Befehl aus.

Mithilfe von **Ctrl+R** kannst du die History durchsuchen. Gebe einfach **Ctrl+R** gefolgt von dem Suchbegriff ein, zum Beispiel *pwd*. Wenn du **Ctrl+R** erneut eingibst, wird der nächste Treffer angezeigt (falls es einen weiteren Treffer gibt). Enter löst auch hier den Befehl aus.

Der Befehl **history -c** löscht den gesamten Inhalt der History.

## xwmconfig

Die beiden weiteren versteckten Dateien, die du auf dem Bildschirmfoto gesehen hast (*.xinitrc* und *.xsession*), wurden während der Installation vom **setup** Kommando erstellt, welches wiederum den Befehl **xwmconfig** ausgeführt hat. Rufe **xwmconfig** erneut auf. Die Anzeige sollte dir von der Installation vertraut sein:




Belasse die Einstellung bitte auf xinitrc.xfce.

## Systemdienste

Im Bildschirmfoto, das die *ls -al* Ausgabe anzeigt, hast du vielleicht bemerkt, dass ein weiterer Befehl ausgeführt worden ist:

```
/etc/rc.d/rc.gpm stop
```

**/etc/rc.d/rc.gpm** ist der Befehl und **stop** der Parameter.

Damit wurde der Dienst gpm, der für die Maus in der Konsole zuständig ist, beendet. Dieser Befehl wurde nur für den Kurs ausgeführt, damit auf den folgenden Anzeigen der Mauscursor nicht mehr erscheint. Du kannst es dennoch gerne einmal testen. Der Mauszeiger sollte, nachdem du den Befehl ausgeführt hast, verschwinden. Beim nächsten Systemstart würde er allerdings wieder angezeigt. Mithilfe des folgenden Befehls kannst du den Dienst wieder starten:

```
/etc/rc.d/rc.gpm start
```

Falls du nicht möchtest, dass ein Dienst beim Systemstart gestartet wird, kannst du unter Slackware einfach die Startdatei als nicht-ausführbar markieren.

Schaue dir zuvor einmal die Datei mit dem **ls -al** Kommando an.

```
ls -al /etc/rc.d/rc.gpm
```




du siehst, dass die Datei für jeden ausführbar ist, beschreiben darf allerdings nur der *Owner*, in diesem Falle root.:

> `-rwxr-xr-x`

Oktal wäre dies **Owner**: `rwx` = 4+2+1 = 7 / **Group**: `r-x` : 4+0+1 = 5 / **Others**: `r-x` : 4+0+1 = 5 oder zusammengesetzt: **755**.

Der **chmod** Befehl bietet einen einfachen Parameter, um eine Datei nicht mehr als ausführbar zu markieren:

```
chmod -x /etc/rc.d/rc.gpm
```

Dies entfernt alle *executable* Rechte von der Datei. Gebe erneut **ls -al** ein, um die Änderungen zu prüfen:

```
ls -al /etc/rc.d/rc.gpm
```

> `-rw-r--r--`

Somit wird der Dienst beim Systemstart nicht mehr gestartet. Möchtest du, dass er wieder gestartet werden soll, führe den **chmod** Befehl einfach mit **+x** aus:

```
chmod +x /etc/rc.d/rc.gpm
```