Einführung in Linux und Gaussian

Teil I
Linux-Einführung

Was ist Linux? Was ist eine Linux-Distribution?

- Der Begriff Linux bezeichnet den eigentlichen Betriebssystemkern (von Linus Torvalds).
- Dieser Betriebssystemkern (Kernel) vermittelt zwischen Treiberebene und Programmierschnittstellen (API).
- Eine Linux-Distribution besteht neben diesem Kernel aus einer Vielzahl von Programmpaketen sowie einer eigenen Paketverwaltung.
- Bekannte Distributionen sind Debian, Red Hat/Fedora, Ubuntu und openSUSE.
- Bestimmte Distributionen unterliegen vollständig der GPL (*General Public License*), z.B. Debian.

Das Herzstück für Anwender: die Shell

- Die Shell ist ein Kommandozeileninterpreter (Befehlszeileninterpreter).
- Ein sogenannter Prompt (eine Eingabeaufforderung) zeigt die Bereitschaft der Shell, ein Kommando entgegenzunehmen:

  `Benutzer@Computer:/PFAD/$ Kommando`

- Kommandos können Optionen und Parameter besitzen:

  `Kommando [OPTIONEN] PARAMETER`
  (z.B. `rm -r Verzeichnis`)

- Eine Eingabe wird mit Return bestätigt, die Eingabezeile sieht dann beispielsweise wie folgt aus:

  `Benutzer@Computer:~$ rm -r Verzeichnis`

Wichtige Kommandos I: Informationen zu Kommandos

- `man [OPTIONEN] KOMMANDO`

  listet die Optionen eines Kommandos auf und zeigt häufig auch mögliche Anwendungsbeispiele.
  (z.B. `man rm` oder `man -w rm`)

- `apropos STICHWORT`

  dient zum Auffinden von man-pages oder Kommandos.

- `info [OPTIONEN] KOMMANDO`

  zeigt, sofern info-pages für das Kommando vorhanden sind, Informationen an, die in einigen Fällen über den Inhalt der man-page hinausgehen.

- man-pages können eine große Hilfe sein, es benötigt allerdings etwas Zeit, sich in diesen zurechtzufinden.

- Sehr häufig hilft auch eine einfache Internetsuche.

Wichtige Kommandos II: Wildcards und Tricks

- Wildcards sind Platzhalter für Zeichen von Dateinamen:
  - `?` steht für ein beliebiges Zeichen.
  - `*` steht für eine beliebige Anzahl verschiedener Zeichen.
  - `[ ]` steht für eine Auswahl von Zeichen.
- Ein- und Ausgabeumleitung:
  - `<` Eingabe aus einer Datei.
  - `>` Ausgabe in eine Datei (überschreibt existierende Datei).
  - `>>` Ausgabe in eine Datei (hängt an existierende Datei an).
- Der Operator Pipe (`|`) leitet die Ausgabe von `KOMMANDO1` in die Eingabe von `KOMMANDO2`:

  `KOMMANDO1 | KOMMANDO2`

  (z.B. `dmesg | grep -n USB`)

Wichtige Kommandos III: Verzeichnis- und Dateiverwaltung

- `mkdir [OPTIONEN] VERZEICHNIS`

  erzeugt ein Verzeichnis mit dem gewählten Namen.

- `rm -r VERZEICHNIS`

  löscht das Verzeichnis sowie alle seine Inhalte (Dateien, sämtliche Unterverzeichnisse usw.)! Beim Aufruf dieses Kommandos ist allergrößte Vorsicht geboten!

- `ls [OPTIONEN]`

  gibt den Inhalt des aktuellen Verzeichnisses aus.

- `cd [OPTIONEN] [VERZEICHNIS]`

  wechselt in das angegebene Verzeichnis oder, falls nicht angegeben, in das eigene Heimatverzeichnis.

- Folgende Verzeichnis-Abkürzungen sind definiert:

  `~/` steht für das eigene Heimatverzeichnis.
  `./` steht für das aktuelle Verzeichnis.
  `../` steht für das übergeordnete Verzeichnis.

Wichtige Kommandos III: Beispiele und Übungen

- Das Verzeichnis Rechnungen soll erstellt werden:

  ```
  mkdir Rechnungen
  ```

- Das Verzeichnis Rechnungen und alle Inhalte sollen definitiv und unwiderruflich gelöscht werden:

  ```
  rm -r Rechnungen
  ```

- Es soll das Verzeichnis Rechnungen erstellt, in das Verzeichnis gewechselt und dessen Inhalt ausgegeben werden:

  ```
  mkdir Rechnungen && cd Rechnungen && ls -la
  ```

- Es soll im Verzeichnisbaum zwei Verzeichnisse nach oben gegangen werden:

  ```
  cd ../../
  ```

- Was passiert, wenn man `cd ./` eingibt?

Wichtige Kommandos IV: Ausgabe von Textdateien

- `cat [OPTIONEN] [DATEI1] [DATEI2] ...`
  zeigt den Inhalt der Datei(en) auf der Standardausgabe.
- `less [OPTIONEN] [DATEI1] [DATEI2] ...`
  zeigt den Inhalt der Datei(en) interaktiv.
- `head [OPTIONEN] [DATEI1] [DATEI2] ...`
  zeigt die ersten Zeilen der Datei(en).
- `tail [OPTIONEN] [DATEI1] [DATEI2] ...`
  zeigt die letzten Zeilen der Datei(en).

Wichtige Kommandos IV: Beispiele und Übungen

- Die Datei H2O.txt soll ausgegeben werden:

  ```
  cat H2O.txt
  ```

- Die Datei H2O.txt soll interaktiv ausgegeben werden:

  ```
  less H2O.txt
  ```

- Die ersten sechs Zeilen der Datei H2O.txt sollen ausgegeben werden:

  ```
  head -n 6 H2O.txt
  ```

- Die letzten drei Zeilen der Datei H2O.txt sollen ausgegeben werden:

  ```
  tail -n 3 H2O.txt
  ```

Wichtige Kommandos V: Dateien erstellen und verwalten

- `cp [OPTIONEN] DATEI ZIEL`

  kopiert eine Datei. Je nach Angabe des Ziels wird die Datei entweder unter anderem Namen in das aktuelle Verzeichnis oder mit gleichem oder anderem Namen in ein anderes Verzeichnis kopiert.

- `mv [OPTIONEN] DATEI ZIEL`

  verschiebt eine Datei und/oder benennt sie um. Je nach Angabe des Ziels wird die Datei im aktuellen Verzeichnis umbenannt oder mit gleichem oder anderem Namen in ein anderes Verzeichnis verschoben.

- `gedit [OPTIONEN] [DATEI1] [DATEI2] ...`

  ist ein einfacher grafischer Texteditor.

- `vim [OPTIONEN] [DATEI1] [DATEI2] ...`

  ist ein komplexer Texteditor, der im Terminal läuft.

Wichtige Kommandos V: Beispiele und Übungen

- Die Datei H2O.txt soll in das Heimatverzeichnis kopiert werden. Falls im Heimatverzeichnis bereits eine Datei mit diesem Namen vorhanden ist, soll nachgefragt werden, ob sie überschrieben werden soll:

  ```
  cp -i H2O.txt ~/
  ```

- Die Datei H2O.txt soll in H3O.txt umbenannt werden:

  ```
  mv H2O.txt H3O.txt
  ```

- Es sollen drei leere Dateien erzeugt und mit gedit geöffnet werden:

  ```
  touch 1.txt 2.txt 3.txt && gedit *.txt
  ```

- Schreiben Sie in die Datei 1.txt ein paar Sätze, in denen die Wörter Maus, Haus und Hausen vorkommen. Speichern Sie die Datei und benennen Sie sie dann in Hausordnung.txt um.

Wichtige Kommandos VI: Suchen von und in Dateien

- `locate [OPTIONEN] MUSTER`

  sucht auf Basis einer Datenbank nach Verzeichnissen oder Dateien, bei denen das angegebene Muster irgendwo im Pfadnamen vorkommt. Diese Suche ist sehr schnell, aber die Datenbank muss stets auf dem aktuellen Stand sein.

- `find [VERZEICHNIS] [SUCHAUSDRUCK]`

  ist ein sehr mächtiges Suchwerkzeug, welches neben dem eigentlichen Suchmuster eine Vielzahl weiterer Kriterien für das Durchsuchen des Verzeichnisbaums erlaubt.

- `grep [OPTIONEN] MUSTER [DATEI1] [DATEI2] ...`

  durchsucht den Inhalt von Dateien nach dem angegebenen Muster.

Wichtige Kommandos VI: Beispiele und Übungen

- Es soll global nach der Datei passwd gesucht werden:

  ```
  locate passwd
  ```

- Es soll im Verzeichnis /home nach der Datei Hausordnung.txt gesucht werden:

  ```
  find /home -type f -name ¨Hausordnung.txt¨
  ```

- Wechseln Sie in das Verzeichnis, in der sich die Datei Hausordnung.txt befindet, und suchen Sie nach diesen Mustern: `¨Haus¨`, `¨Haus*¨`, `¨Haus.*¨`, und `¨.aus¨`. (z.B. `grep ¨Haus¨ Hausordnung.txt`).

Teil II
Gaussian-Einführung

Was ist Gaussian?

- Gaussian ist ein Programmpaket für quantenchemische Berechnungen, welches in Fortran geschrieben wurde.
- Das Projekt wurde von John Anthony Pople gegründet.
- Gaussian gilt als einer der wichtigsten Vertreter unter den Quantenchemie-Programmen.
- Weitere bekannte Quantenchemie-Programmpakete sind Dalton, GAMESS, Molpro, NWChem, ORCA, Q-Chem und Turbomole.

Die Gaussian Input-Datei

`%chk=/Pfad/Dateiname.chk`	Checkpoint-Datei
`%nproc=2`	Zahl der Prozesse
	Leerzeile
`# RHF 6-31+G* opt`	Methode/Basissatz, Art der Rechnung
	Leerzeile
`Beliebiger Text`	Kommentar
	Leerzeile
`Ladung, Multiplizität`	
`Koordinaten`	
	Leerzeile
`Zusätzliche Einträge`	Variablen usw.
	Leerzeile

Die Gaussian Input-Datei

`0 1`	Ladung, Multiplizität
`C`	Darstellung des Moleküls als Z-Matrix
`H  1 r1`	
`Cl 1 r2 2 a3`	
`Cl 1 r2 2 a3 3 d4`	
`Cl 1 r2 2 a3 4 d4`	
	Leerzeile
`r1 = 1.089`	Variable: Abstand in Å
`r2 = 1.750`	Variable: Abstand in Å
`a3 = 109.471`	Variable: Winkel in Grad
`d4 = 120.000`	Variable: Diederwinkel in Grad
	Leerzeile

Moleküleditoren

- `molden`
  ist ein sehr einfacher, aber auch sehr effizienter Moleküleditor.

- `avogadro`
  ist ein optisch sehr ansprechender Moleküleditor.