

Propädeutikum

Programmierung in der Bioinformatik

Einführung in die Shell

Thomas Mauermeier

23.10.2018

Ludwig-Maximilians-Universität München

Arbeit im CIP-Pool

- Arbeitsplatz sauber halten!
- Unterhaltungen → Aufenthaltsraum
- Beschädigungen melden
- Computer will nicht?
 - LAN-Kabel?
 - Monitor?
 - Anderer Rechner?
- Infos zum CIP-Pool/RGB:
<https://www.rz.ifi.lmu.de/>

Login am Rechner

- Zugangsdaten: CIP-Kennung
(habt ihr in O-Phase angelegt)

Keine Kennung?

- RGB-Sprechstunde:
Mo. bis Fr. von 14:00–17:00
Raum L U113
Oettingenstr. 67
- (amtlichen) Lichtbild- und
Studentenausweis mitnehmen

Achtung

Es wird leider keinen ausführlichen Support für Remotezugriff geben

Wichtig für:

- Leute die ihren eigenen Rechner mitbringen wollen
- Leute die von Zuhause aus Zugriff auf CIP-Pool wollen

Freischaltung via CipConf-Tool (<https://conf.cip.ifi.lmu.de/>), unter *Change remote connection config* den Zugriff erlauben.

Wer bereits mit eigenem Laptop und ssh arbeiten will: `remote.cip.ifi.lmu.de`

Zugriff läuft dann über:

ssh (*secure shell*)

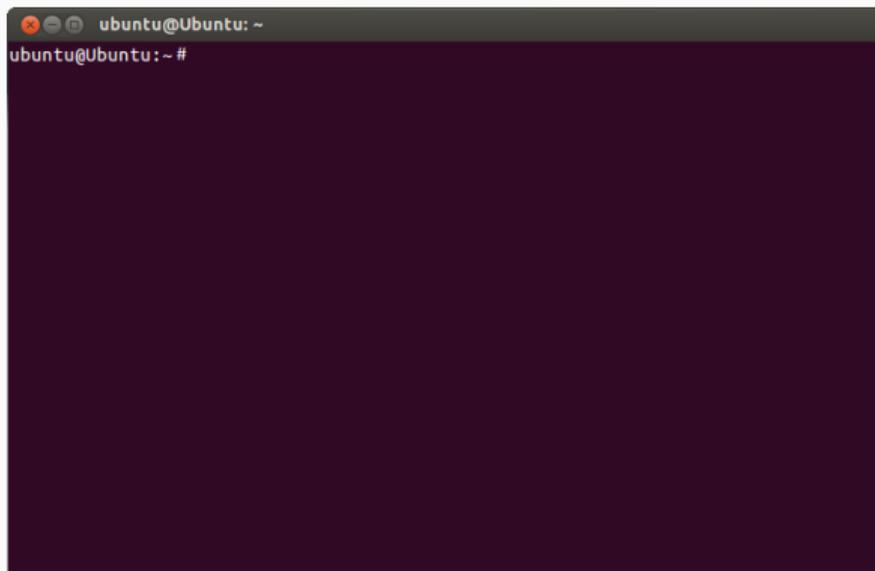
- Kommandozeile (Shell)
- *keine* grafische Oberfläche
- `https://www.rz.ifi.lmu.de/infos/ssh_de.html`

rdp (*remote desktop protocol*)

- grafische Oberfläche
- überträgt den Desktop
- `https://www.rz.ifi.lmu.de/infos/rdp_de.html`

Shell?

- *Shell* meint eine User-Schnittstelle zum Betriebssystem
- Kann sowohl grafische Oberfläche als auch Kommandozeile sein
- Hier nur Kommandozeile gemeint (Command line interface, “Terminal”, ...)

A screenshot of a terminal window. The title bar at the top reads "ubuntu@Ubuntu: ~". The terminal content shows the prompt "ubuntu@Ubuntu:~#" on the first line, with a dark purple background and a light purple cursor.

Warum mit Shell umgehen lernen?

mächtig sehr flexibel, viele Funktionen

schnell schneller Einzeiler vs. langwieriges Skript schreiben,
auch kein herumklicken in Menüs

portabel oft Arbeit auf externen Rechnern mittels ssh (kein GUI!), ziemlich
ähnlich bis identisch auf unixoiden Systemen (Linux, BSD, macOS, ...)

Last but not least: Wird in diversen Vorlesungen, Praktika, etc. sowieso verlangt, dass man (mindestens) einen groben Überblick hat.

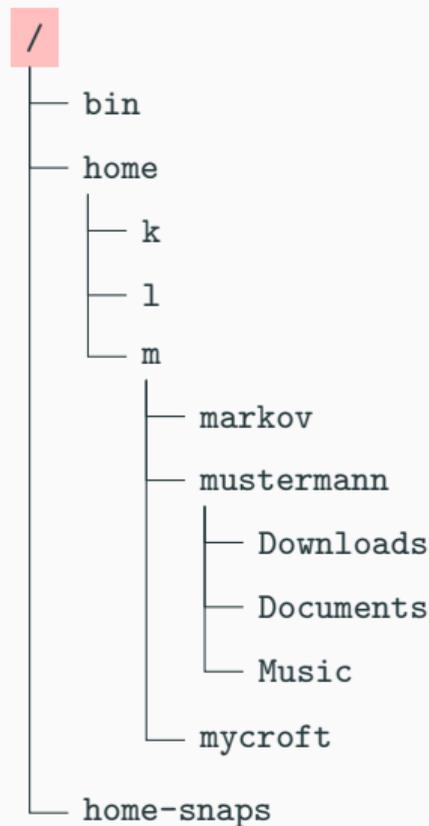
```
mustermann@hallimasch.cip.ifi.lmu.de : ~ $
```

- Euer **Username**
- Der **Hostname**, also auf welchem Rechner ihr eingeloggt seid
- Das **Arbeitsverzeichnis**, also der Pfad des Verzeichnisses in dem ihr gerade seid

```
/home/m/mustermann/Documents/
```

Wurzelverzeichnis (root directory)

Oberstes Verzeichnis in der Verzeichnishierarchie (“der Anfang”)

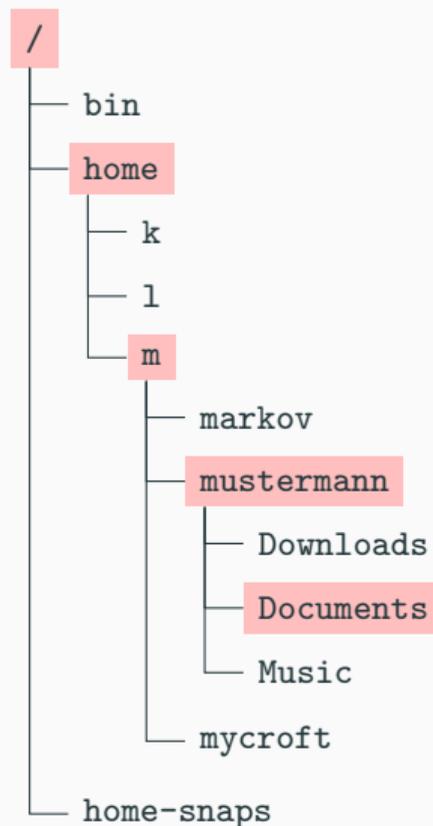


Pfade – absoluter Pfad

```
/home/m/mustermann/Documents/
```

absoluter Pfad

Voller Pfad vom Wurzelverzeichnis aus;
muss also immer mit einem / beginnen



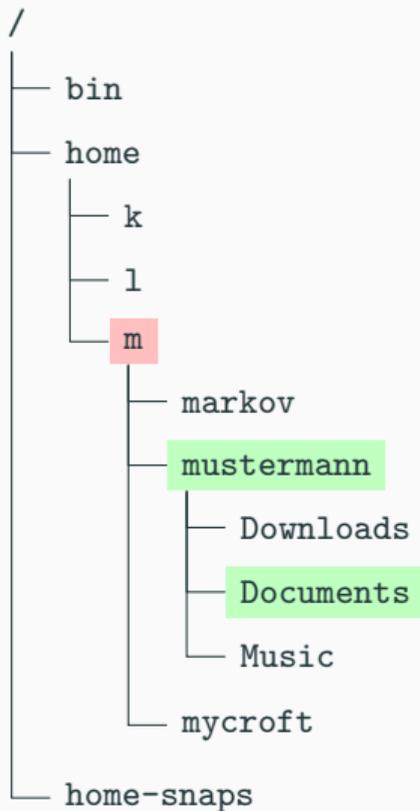
Pfade – Arbeitsverzeichnis & relativer Pfad

Angenommen ihr befindet euch gerade in:

`/home/m/`

Arbeitsverzeichnis (working directory)

Verzeichnis in dem man sich aktuell befindet (also “arbeitet”)



Pfade – Arbeitsverzeichnis & relativer Pfad

Angenommen ihr befindet euch gerade in:

```
/home/m/
```

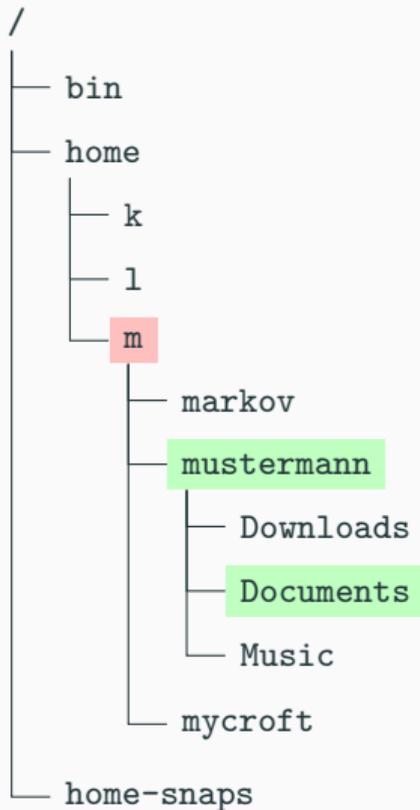
Arbeitsverzeichnis (working directory)

Verzeichnis in dem man sich aktuell befindet (also “arbeitet”)

Dann ist z.B. `mustermann/Documents/`

relativer Pfad

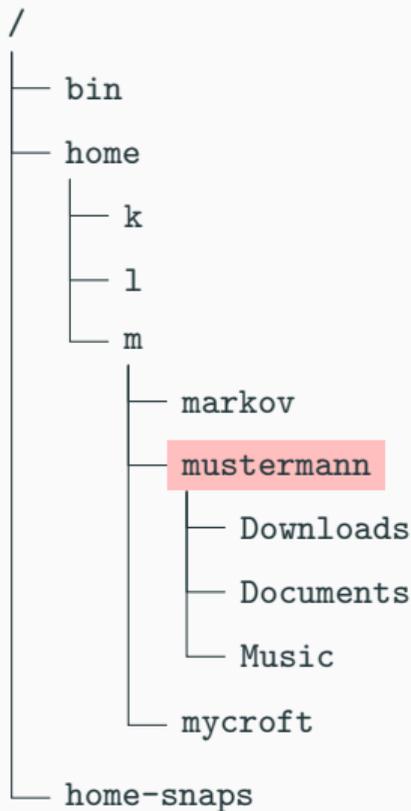
Pfad der vom Arbeitsverzeichnis aus gesehen beschrieben wird



```
/home/m/mustermann/
```

Benutzerverzeichnis (home)

Euer Verzeichnis auf das ihr weitgehend freie Zugriffsrechte habt (heisst aber auch: ihr könnt euch aussperren!)



```
/home/m/mustermann /
```

Benutzerverzeichnis (home)

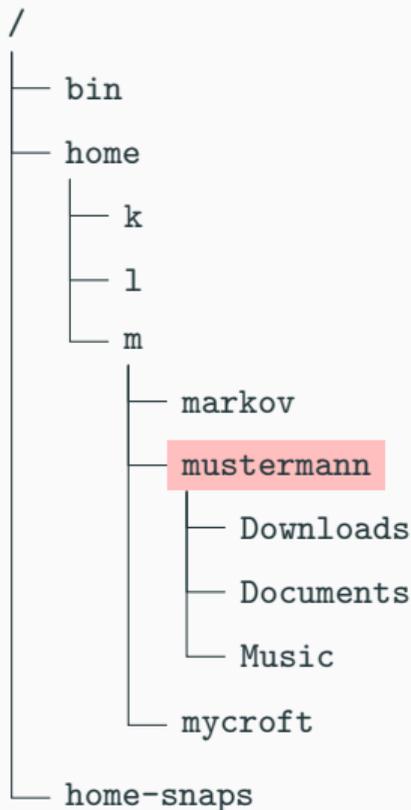
Euer Verzeichnis auf das ihr weitgehend freie Zugriffsrechte habt (heisst aber auch: ihr könnt euch aussperren!)

Kürzel für häufig genutzte Pfade:

~ Benutzerverzeichnis (home)

. Arbeitsverzeichnis

.. Übergeordnetes Verzeichnis



```
mustermann@hallimasch.cip.ifi.lmu.de: ~/Documents $ pwd
```

```
/home/m/mustermann/Documents
```

```
mustermann@hallimasch.cip.ifi.lmu.de: ~/Documents $ cd ../Downloads
```

```
mustermann@hallimasch.cip.ifi.lmu.de: ~/Downloads $ pwd
```

```
/home/m/mustermann/Downloads
```

- Das **Arbeitsverzeichnis** wird in der Prompt angezeigt
- `pwd` zeigt uns einen **absoluten Pfad** an
- `../Downloads` ist ein **relativer Pfad** der *ausgehend vom aktuellen Arbeitsverzeichnis* in ein anderes Verzeichnis zeigt

Syntax von Shell-Befehlen

Meistens in etwa diese Form, aber nicht immer:

`<Befehl> <Flags> <Argumente>`

Befehl Name des Befehls

Flags Optionen die dem Befehl übergeben werden

z.B. `-h` oder `--help` für Kurzhilfe zum Befehl

Argumente Der "Input" den der Befehl bekommt

z.B. Ein Pfad, eine Adresse, ein String ...

Beispiel `ls -la`

- Befehl `ls` für Auflisten von Dateien im Arbeitsverzeichnis
- Flag `-la` kombiniert `-l` (Listenformat) und `-a` (versteckte Dateien anzeigen)
- Argument bleibt leer

RTFM

Read the friendly (?) manual

- `man <Befehl>` zeigt ausführliche Anleitung (“manpage”, von *manual*)
- `apropos <Stichwort>` sucht in allen manpages nach nach Stichwort
- `-h` oder `--help` sind oft Flags für Kurzanleitungen zum Befehl
- Suchmaschine
- Ihr werdet noch oft Dokumentationen lesen müssen – am besten jetzt angewöhnen, aufmerksames Lesen erspart oft langes Grübeln und Rad neu erfinden

Shell-Befehle – Dateisystem

<code>pwd</code>	Arbeitsverzeichnis ausgeben	print working directory
<code>cd</code>	Wechselt das Arbeitsverzeichnis	change directory
<code>ls</code>	Zeigt Inhalt des Arbeitsverzeichnisses	list
<code>tree</code>	Wie <code>ls</code> , Ausgabe in Baumformat	–
<code>cp</code>	Kopie von Dateien/Verzeichnissen	copy
<code>mv</code>	Verschieben/Umbenennen von Dateien	move
<code>ln</code>	Verknüpfung (Link) erstellen	link
<code>touch</code>	Datei erstellen (oder Zugriffsstempel ändern)	–
<code>mkdir</code>	Erstellt Verzeichnis	make directory
<code>rm</code>	Löscht Datei (⚠️ kein Papierkorb)	remove
<code>rmdir</code>	Löscht Verzeichnis (⚠️ kein Papierkorb)	rm directory
<code>chmod</code>	Berechtigungen ändern (mehr später)	change file mode
<code>find</code>	Datei in Pfad suchen	–

Shell-Befehle – Arbeit mit Text

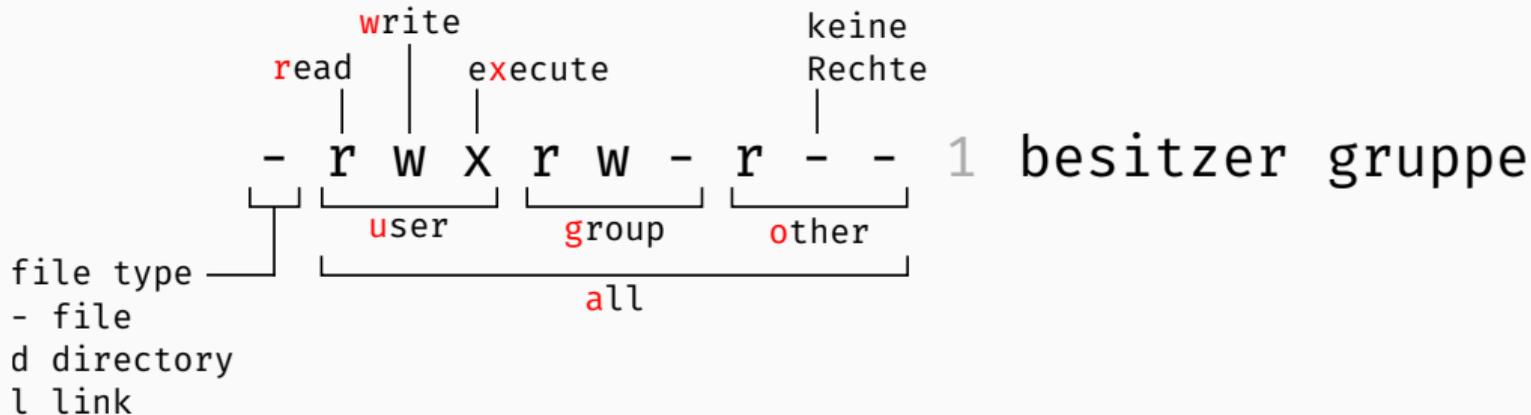
<code>less</code>	Betrachten von Input (als Text)	Alternative zu “ more ”
<code>nano</code>	Einfacher Texteditor	–
<code>vi/vim</code>	Komplexerer Texteditor	–
<code>grep</code>	Suche in Input nach regulärem Ausdruck (Regex)	g lobally search r egex and p rint
<code>wc</code>	Anzahl Wörter in Input	w ord c ount
<code>cat</code>	Konkateniert Input (wird meist für Ausgabe verw.)	c on c atenate
<code>sort</code>	Sortiert Input	–
<code>uniq</code>	Entfernt/Zeigt Duplikate in Input	u nique
<code>cut</code>	“Schneidet” Teile von Input ab	–
<code>head</code>	Ausgabe des Beginns von Input	–
<code>tail</code>	Ausgabe des Endes von Input	–
<code>diff</code>	Unterschiede im Input Zeile für Zeile anzeigen	d ifference
<code>echo</code>	String ausgeben	–

Shell-Befehle – Diverses

<code>man</code>	Anleitung zu Befehl/Anwendung	manual
<code>apropos</code>	Suche nach Stichwort in manpages	–
<code>tar</code>	Dateien zu einem Archiv zusammenfassen	tape archiver
<code>gzip</code>	Dateien komprimieren	GNU zip
<code>ps</code>	Laufende Prozesse ausgeben	process status
<code>top/htop</code>	Wie ps, aber mit laufenden Updates, interaktiv	table of processes
<code>kill</code>	Prozess beenden	–
<code>wget</code>	Datei von http/ftp-Server herunterladen	WWW get
<code>ssh</code>	Remote-Zugriff auf Shell eines externen Rechners	secure shell
<code>scp</code>	Kopieren von/zum remote Rechner über ssh-Protokoll	secure copy
<code>history</code>	Listet zuletzt eingegebene Befehle auf	–

Dateirechte

```
mustermann@zolling.cip.ifi.lmu.de:/home/proj/tutorium_bioinformatik/proteinSeqs $ ls -la
drwxr-xr-x  2 berchtolde bio          4096 Mär 28  2018 .
drwxrwxr-x 11 berchtolde vlg_tutorium 4096 Jul 12 09:16 ..
-rw-r--r--  1 berchtolde bio       1749480 Mär 28  2018 Escherichia_coli.fasta
-rw-r--r--  1 berchtolde bio        691681 Mär 28  2018 Haemophilus_influenzae.fasta
-rw-r--r--  1 berchtolde bio        585764 Mär 28  2018 Helicobacter_pylori.fasta
```



chmod <Berechtigung> <Datei>

<Berechtigung> – symbolic mode

[ugoa] [-+=] [rwx]

[ugoa] Rechte von user, group, other, all auswählen

[-+=] Rechte entziehen, hinzufügen, setzen

[rwx] Zu verändernde Rechte für **r**ead, **w**rite, **e**xecute

```
rw-rw-r--
chmod u-x  rw-rw-r--
chmod o+xw rw-rw-rwx
chmod o=   rw-rw----
chmod og+rwx rw-rwxrwx
chmod -x   rw-rw-rw-
chmod o=rx rw-rw-r-x
chmod a+x  rwxrwxr-x
```

(rot = 'Ziele' des Befehls)

```
chmod <Berechtigung> <Datei>
```

<Berechtigung> – octal mode

Oktalzahl für jeweils u, g, o

0	keine Rechte
1	execute
2	write
2+1 = 3	write, execute
4	read
4+1 = 5	read, execute
4+2 = 6	read, write
4+2+1 = 7	read, write, execute

```
(764) -rwxrw-r--  
chmod 664 rw-rw-r--  
chmod 667 rw-rw-rwx  
chmod 660 rw-rw----  
chmod 677 rw-rwxrwx  
chmod 666 rw-rw-rw-  
chmod 665 rw-rw-r-x  
chmod 775 rwxrwxr-x
```

(rot = 'Ziele' des Befehls)

Anmelden auf einem Rechner mit bestimmter Adresse :

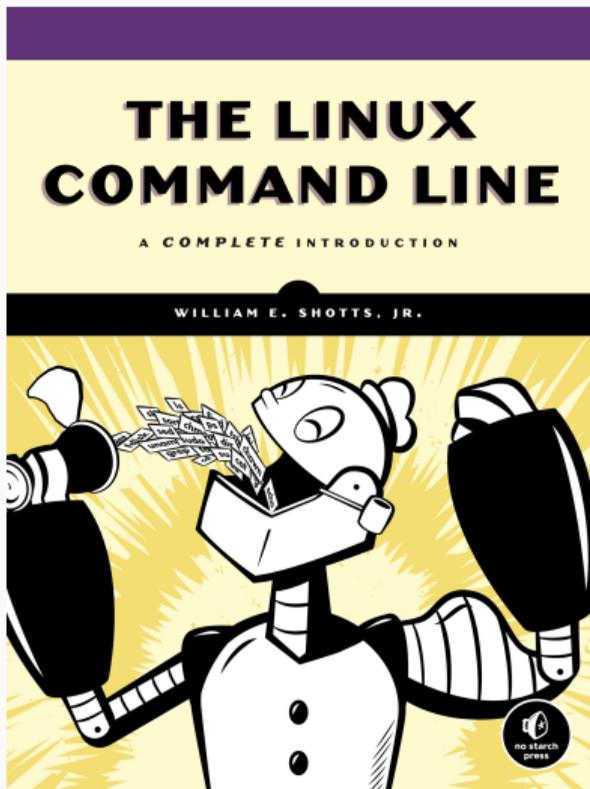
```
ssh -p <port> <username>@<hostname>
```

Kopieren von Daten von einem Rechner zu einem anderen. Die Adresse ist *nicht* nötig wenn man von/zum lokalen Rechner kopiert:

```
scp <user>@<host> : <quellpfad> <user>@<host> : <zielpfad>
```

Tipps zum Arbeiten in der Shell

<code>Tab</code>	Autovervollständigung von Befehlen
<code>Strg + Shift + t</code>	Öffnet neuen Tab im Terminal (CIP-Rechner)
<code>Strg + c</code>	Abbruch des aktuellen Befehls
<code>Strg + l</code>	Leert den Inhalt der Shell (analog zu <code>clear</code>)
<code>Strg + r</code>	Durchsuchbare Chronik der letzten Befehle
<code>↑</code> , <code>↓</code>	Manuell durch letzte Befehle navigieren
<code>q</code>	Zum Beenden von <code>man</code> oder <code>less</code>
<code>!!</code>	Alias für <code>ls -l</code>



Für diejenigen die mehr wissen wollen:

The Linux Command Line

A Complete Introduction

William E. Shotts, Jr.

Verfügbar als eBook in der TUM Bibliothek