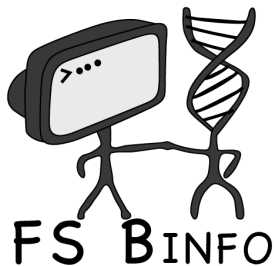


Die Shell - Teil 2 - Arbeiten mit Dateien



Bernd Streppel

Fachschaft Bioinformatik

3. November 2009

Zeitplan

- 27.10 Einführung in Linux und die Shell
 - 03.11 **Arbeiten mit Dateien auf der Shell**
 - 10.11 Einführung in Java
 - 17.11 Umgang mit Arrays, Schleifen und Dateien
 - 24.11 Einführung in objekt-orientierte Programmierung
 - 01.12 Rekursion und Iteration
 - 08.12 GUI Entwicklung
 - 15.12 Debuggen mit Eclipse
-
- 12.01 Taschenrechner in Java I
 - 19.01 Taschenrechner in Java II
 - 26.01 Wiederholung

Webseite:

<http://www.bioinformatik-muenchen.com/bioinfocom/informatik->

Übungen für Daheim

Gibt es Fragen zu den Übungen?

1. Finde heraus was die Befehle `kill` und `cat` tun
2. Finde heraus was genau der Befehl `ps aux` ausgibt
3. Finde einen Befehl, der das aktuelle Datum und die Uhrzeit ausgibt
4. Finde heraus was die Datei `/etc/passwd` beinhaltet
5. Lese das Kapitel "Benutzer- und Berechtigungskonzepte unter Linux" unter http://www.selflinux.org/selflinux/html/nutzer_unter_linux

Selber ausprobieren

- Hier im CIP-Pool
- Von zu hause aus per SSH
 - ssh unter Linux
 - putty unter Windows
- Muss zuerst freigeschaltet werden unter:
- <https://tools.rz.ifi.lmu.de/cipconf/>



less

Was vom letzten mal noch übrig geblieben ist:

- Programm zum lesen von Dateien (sog. Pager)
- Wird auch verwendet um Man pages darzustellen
- **Übung:**
 1. Finde heraus wie man mit `less` in Dateien sucht

Umgang mit komprimierten Dateien

- GZIP und BZIP2 sind Komprimierungsverfahren wie ZIP
- (Ent)packen nur eine einzige Datei
- Mehrere Dateien oder Verzeichnisse müssen zuvor mit tar zu einer Datei zusammengefasst werden
- GZIP:
 - Entpacken:
 - `gunzip ARCHIV.gz`
 - `tar xzvf ARCHIV.tar.gz`
 - Packen:
 - `gzip ARCHIV.gz DATEI`
 - `tar czvf ARCHIV.tar.gz DATEI...`



Umgang mit komprimierten Dateien

- BZIP2:
 - Entpacken:
 - `bunzip2 ARCHIV.bz2`
 - `tar xjvf ARCHIV.tar.bz2`
 - Packen:
 - `bzip2 ARCHIV.bz2 DATEI`
 - `tar cjvf ARCHIV.tar.bz2 DATEI...`
- less-Varianten: `zless` für `.gz` und `bzless` für `.bz2`
- **Übung:**
 1. Öffne die Datei `"bash/README.commands.gz"`
 2. Suche nach `"Computer"` und finde heraus was der Befehl zum herunterfahren ist
 3. Packe das Verzeichnis `"bash"` als `.tar.gz` und `.tar.bz2`
 4. Entpacke `.tar.gz` Datei nach `"bash-gz"` und `.tar.bz2` nach `"bash-bz2"`

find

- Befehl um nach Dateien zu suchen
- `find VERZEICHNIS -name MUSTER`
- **Übung:**
 1. Finde alle README Dateien in `"/usr/share/doc"`
 2. Finde alle versteckten Verzeichnisse in deinem HOME-Verzeichnis
 3. Finde alle leeren Verzeichnisse in deinem HOME-Verzeichnis

Texteditoren

- nano Sehr einfacher Texteditor
- vim Sehr umfangreicher und gewöhnungsbedürftiger Texteditor
- emacs Sehr umfangreicher und gewöhnungsbedürftiger Texteditor
- **Übung:**
 1. Erstelle mit dem Editor deiner Wahl eine Datei hallo mit Inhalt "Hallo Welt"!
 2. Öffne die Datei erneut und ersetze den Inhalt durch etwas anderes!

Grundlegende Idee

- Jedes Programm hat genau **eine** Funktion
- Zum Beispiel sortieren, filtern oder anzeigen

Komplexe Problem werden
durch die Verkettung von
Befehlen gelöst

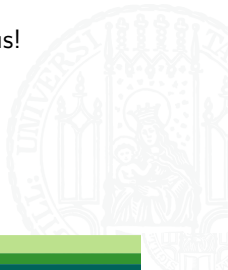
Verbinden von Programmen - die Pipe

- Jedes Programm hat 1 Eingabekanal und 2 Ausgabekanäle
- STDIN, STDOUT, STDERR
- | verbindet STDOUT eines Programms mit STDIN des Nächsten.
- Beispiel: `ls | less`
- > Verbindet STDOUT mit einer Datei
- < Verbindet STDIN mit einer Datei
- Beispiel: `find / -name '*.mp3' > ergebnis.txt`
- Fehler werden trotzdem angezeigt, weil STDERR nicht umgeleitet wird!



Einige Programme

- `sort [-n]` Sortiert Daten alphabetisch/numerisch
- `grep MUSTER` Filtert die Daten nach MUSTER
- `head` Zeigt den Anfang einer Datei an
- `tail` Zeigt das Ende einer Datei an
- `wc` Zählt Buchstaben, Wörter und Zeilen
- **Übung:**
 1. Finde in der Datei “`etc/passwd`” deine eigene Userid!
 2. Wieviele Benutzer haben “`bash`” als Standardshell?
 3. Gib die ersten 20 Dateien in “`/usr/lib`” alphabetisch aus!



Berechtigungen

- Es gibt 3x3 Stufen der Berechtigungen
- Wer: u(ser), g(ruppen) und a(lle)
- Was: Lesen 4, Schreiben 2, Ausführen 1
- `chmod BERECHTIGUNG DATEI` setzt die Berechtigungen
 - `chmod g+w DATEI` Für die Gruppe zum Schreiben freigeben
 - `chmod +r DATEI` Für alle (user/gruppe/alle) zum Lesen freigeben
 - `chmod 644 DATEI` Für alle zum lesen und für den User zum Schreiben freigeben



Berechtigungen setzen

- `chgrp GRUPPE DATEI` ändert die Gruppenzugehörigkeit einer Datei
- `chown BENUTZER DATEI` ändert den Besitzer einer Datei
- `ls -la DATEI` Zeigt die Berechtigungen einer Datei an
- **Übung**
 1. Erstelle eine Datei `test` und schränke die Rechte so ein dass nur du Zugriff hast
 2. Erstelle einen Ordner und lasse nur dich und die Gruppe `bioinfo` darauf Zugreifen

Übungen für Zuhause

- Versuche dich in vim oder emacs einzuarbeiten
 - <http://www.informatikserver.at/selflinux/html/vim.html>
 - <http://www.linuxhaven.de/dlhp/HOWTO/DE-Emacs-Einsteiger-HOWT>
- Erstelle einen Ordner “gruppenarbeit” und gib ihn nur für deine Gruppe (stud) frei.
- Kopiere den Ordner “/usr/share/doc/zsh” in den neu erstellten Ordner.
- Komprimiere den Ordner zu zsh.tar.gz und entpacke ihn wieder nach zsh-neu.
- Durchsuche den entpackten Ordner nach allen .gz Dateien und lasse die Anzahl ausgeben.