Übung 1: Ein-, Aus- und Weitergabe

1. Öffnen Sie eine SSH-Verbindung zum Rechner `ella`.
2. Starten Sie einen lange (d.h. länger als eine Stunde) laufenden Prozess.
3. Schieben Sie den Prozess in den Hintergrund.
4. Holen Sie den Prozess wieder in den Vordergrund
5. Brechen Sie den Prozess ab
6. Legen Sie in Ihrem Homeverzeichnis das Verzeichnis `My directory` an.
7. Legen Sie nun ein Verzeichnis namens `My sister's directory` an.
8. Löschen Sie beide Verzeichnisse wieder.
9. Legen Sie nun in Ihrem Homeverzeichnis ein Verzeichnis `Vorkurs` an.
10. Erzeugen Sie in dem neu erstellten Verzeichnis eine Datei `poetik.txt`, in der Sie *alle* Dateien aus dem Verzeichnis `/resources/corpora/monolingual/raw/projekt_gutenberg/werke/aristote/poetik/cleaned` aneinanderhängen (konkatenieren).
11. Wir wollen nun herausfinden, welche Wörter die häufigsten in diesem Korpus sind.

    Tipp: Lesen Sie die man-page zu den erwähnten Tools.

    a) Der offensichtliche Weg zum zählen von Vorkommen in Dateien ist `grep`. Damit können wir z.B. zählen, wie oft das Wort *Epos* vorkommt. Tun Sie das doch einfach mal mit der Datei poetik.txt; zählen Sie die Wörter *Epos*, *der*, *Der* und *Melodik*. Tipp: Der reguläre Ausdruck `\b` matcht Wortgrenzen.

    b) Per default identifiziert `grep` allerdings Zeilen und keine Wörter. Unsere Zählung aus der vorherigen Aufgabe ist also etwas ungenau – Immer wenn das Wort zwei Mal in der gleichen Zeile vorkommt, verzählen wir uns.

    c) `grep` bietet die Option `-o`. Damit werden nicht Zeilen gefunden, sondern jedes einzelne Vorkommen des Musters wird aufgelistet. In Verbindung mit `wc` können wir nun alle Vorkommen der Wörter *Epos*, *der*, *Der* und *Melodik* zählen.
12. Dummerweise ist das recht umständlich auf diese Weise. Für eine komplette Häufigkeitsverteilung bräuchten wir eine Liste aller Wörter – die wir nicht haben. Wir gehen also etwas anders an die Sache heran. Die nächsten Schritte sind die einzelnen Teile einer Pipe, Sie müssen die Ausgabe also nicht speichern. Wenn Sie wissen, wie jeder Schritt geht (und ihn in ihrer Konsolen-History haben), reicht das.

    a) Da wir Wörter besser zählen können, wenn nur eines auf einer Zeile steht, sollten wir erstmal dafür sorgen, dass jedes Wort auf einer eigenen Zeile steht (verwandeln Sie Leerzeichen in Zeilenumbrüche, indem Sie `sed` oder `tr` und ggf. reguläre Ausdrücke verwenden).

b) Würden wir die Wörter jetzt zählen, hätten wir noch Satzzeichen wie Kommas, Punkte, Klammern u.a., die die Ergebnisse verunreinigen. Wir werden also als nächstes die Ausgabe von Schritt 12a um ihre Satzzeichen „erleichtern". Denken Sie an Charakterklassen bei regulären Ausdrücken.

c) Die eigentliche Zählung folgt nun und besteht aus drei Schritten. Wir machen uns dabei die Option `-c` des Kommandos `uniq` zunutze. Damit gibt `uniq` aus, wieviele Zeilen es jeweils zusammengefasst hat. Da `uniq` aber nur untereinanderstehende Zeilen zusammenfasst, sollten wir die Zeilen erst sortieren. Wollen wir die Wörter dann noch nach ihrer Häufigkeit sortieren, sortieren wir die `uniq`-Ausgabe (Schauen Sie auch mal in die man-page von `sort`, um die richtige Sortier-Art festzulegen).

Am Ende kommt eine Pipe raus, die aus ca. fünf Schritten besteht (es gibt mehr als eine mögliche Lösung). Damit Sie vergleichen können, finden Sie die zehn häufigsten Wörter in der Datei `/home/public/vorkurs_ws17/poetik10.txt`.

Zusatzaufgaben

1. Als wir vorhin die Dateien zusammengefasst haben, haben wir das Inhaltsverzeichnis mitgenommen. Es steckt in der Datei `Druckversion_poetik_clean.html`. Wie muss der Befehl aussehen, damit das Inhaltsverzeichnis ausgeschlossen wird? (Benutzen Sie die Pfeiltasten, um vorherige Befehle zu verändern und erneut zu benutzen).