Andrew Torda
Björn Hansen
Iryna Bondarenko

Zentrum für Bioinformatik

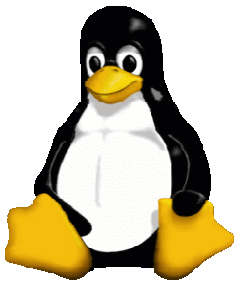
Übung zur Vorlesung
Angewandte Bioinformatik

Sommersemester 2014

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

4./7. April 2014

Übung 1: Linux



1. Einführung

In der folgenden Übung werden Sie mit der Arbeit an den Computern im Studenten-Rechnerpool des ZBHs vertraut gemacht: Sie werden die Arbeitsgrundlagen unter einem Linux-System sowie den Umgang mit wichtigen Linux-Programmen erlernen. Dieses Wissen brauchen Sie für die nächsten Übungen.
Wenn Sie Fragen haben oder nicht weiter wissen, dürfen sie sich jederzeit vertrauensvoll an die Übungsgruppenleiter wenden. Diese werden Ihnen gerne weiterhelfen.

Auf allen Rechnern im ZBH läuft Linux. Dieses ist ein frei verfügbares Open-Source-Betriebssystem. Prinzipiell darf somit jeder Linux herunterladen und den eigenen Wünschen entsprechend verändern. In der Wissenschaft ist Linux beliebt, da viele Programme, die für die wissenschaftliche Arbeit wichtig sind, für Linux frei verfügbar sind.

Als Benutzer hat man unter Linux prinzipiell zwei Möglichkeiten, um mit dem Rechner zu kommunizieren:

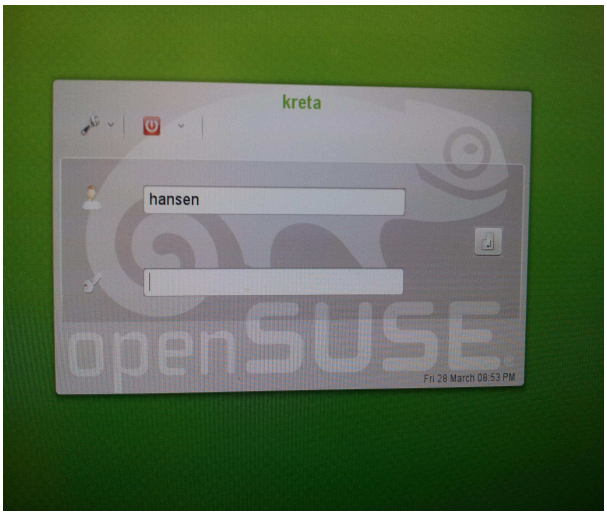
1.) über die grafische Desktop-Oberfläche, welche sehr mit der Oberfläche eines MS-Windows-Systems vergleichbar ist

2.) rein textbasiert über die Kommandozeile einer Shell (Kommandozeileninterpreter)

Am Ende dieser Übung werden Sie verstanden haben, wie man über diese beiden Schnittstellen Dateien anlegen und bearbeiten kann. Auch wenn die Kommunikation mit dem Dateisystem über eine textbasierte Eingabe-Konsole zunächst ein wenig gewöhnungsbedürftig erscheinen mag, so erleichtert und beschleunigt sie die Arbeit in vielen Fällen erheblich. In dieser Übung werden wir uns deshalb zunächst mit der Kommandozeile beschäftigen.

Login:

Das erste, was Sie sehen, wenn sie Ihren Monitor einschalten ist der Anmeldebildschirm. Dieser fordert Sie auf, Ihren Benutzernamen und ihr Passwort einzugeben.



Prinzipiell haben Sie vor dem Einloggen auch die Möglichkeit, zwischen verschiedenen Linux-Arbeitsumgebungen zu wählen. Eine solche Arbeitsumgebung legt fest, wie die grafische Benutzeroberfläche dargestellt werden soll und wie Sie mit ihr interagieren können. Standardmäßig wird die Arbeitsumgebung KDE geladen und wir empfehlen Ihnen, es zumindest anfänglich auch dabei zu belassen.

Falls noch nicht geschehen, dann loggen Sie sich bitte jetzt unter Ihrem Benutzernamen ein. Zu beachten ist hierbei, dass Linux zwischen Groß- und Kleinschreibung unterscheidet. “Hamburg“ ist also beispielsweise nicht dasselbe wie „hamburg“ oder „HamBurG“.

Linux ist ein Mehrbenutzer und Multitasking-System. Das bedeutet, dass mehrere Benutzer gleichzeitig mit mehreren unterschiedlichen Programmen arbeiten können. Um Konflikte und Missbrauch zu vermeiden, hat jeder Benutzer sein eigenes Verzeichnis, das sogenannte Home-Verzeichnis. Linux gibt Ihnen die Möglichkeit, Lese- und Schreibrechte einzeln für jede Ihrer Dateien festzulegen. Standardmäßig hat jeder Benutzer nur in seinem eigenen Home-Verzeichnis Schreibrechte.

Über die grafische Benutzeroberfläche können Sie Programme direkt starten. Viele der für Sie nützlichen Programme sind direkt über das Menü der Taskleiste (Button links unten) verfügbar. Einige Programme werden hier allerdings nicht aufgeführt. Sie werden im Verlauf dieser Übung lernen, die Kommandozeilenumgebung zu nutzen, um auch diese Programme auszuführen.

Diese Übung umfasst zwei Abschnitte: Im ersten werden Sie vor allem mit der Kommandozeile arbeiten. Im zweiten Teil werden Sie die graphische Benutzeroberfläche benutzen, um einige Aufgaben zu lösen.

Teil I (Kommandozeilenschnittstelle):

(1) Rufen Sie das Konsolenprogramm “Terminal” über das Menü der Taskleiste auf. Das Terminal ist ein mächtiges Werkzeug von Linux, welches es Ihnen ermöglicht, über eine Kommandozeile mit Ihrem Rechner zu kommunizieren. Innerhalb des Terminals sehen Sie die sogenannte Eingabeaufforderung, auch Prompt genannt, welche für Sie wie folgt aussieht:

stud2014/ihr_benutzername>

Wann immer Sie diese Eingabeaufforderung sehen, wartet das Terminal darauf, dass Sie einen Befehl eingeben. Ein Befehl bzw. ein Kommando ist ein Programm, welches das Linux-System dazu auffordert etwas zu tun. Die von Ihnen eingegebenen Befehle werden dann von einem Kommandozeileninterpreter, auch Shell genannt, in eine Sprache übersetzt, die Ihr Rechner versteht, und anschließend ausgeführt.

Ein Befehl setzt sich dabei wie folgt zusammen:

command -options arguments

An erster Stelle steht das Kommando selbst, welches angibt, WAS gemacht werden soll. Mit den Optionen kann dann näher spezifiziert werden, WIE das Kommando ausgeführt werden soll. Die Argumente bestimmen, WORAUF das Kommando angewendet werden soll. Häufig steht hier der Name einer Datei. Optionen werden mit einem Bindestrich eingeleitet und, falls mehrere Optionen verwendet werden, direkt hintereinander geschrieben:

command -option1option2

Durch Drücken der Enter-Taste, wird ein Befehl ausgeführt.

Falls Sie sich bei einem bestimmten Befehl nicht sicher sind, welche Eingabeargumente erforderlich sind und welche Optionen Ihnen zur Verfügung stehen, können Sie sich die Hilfeinformationen zu diesem Befehl aufrufen.

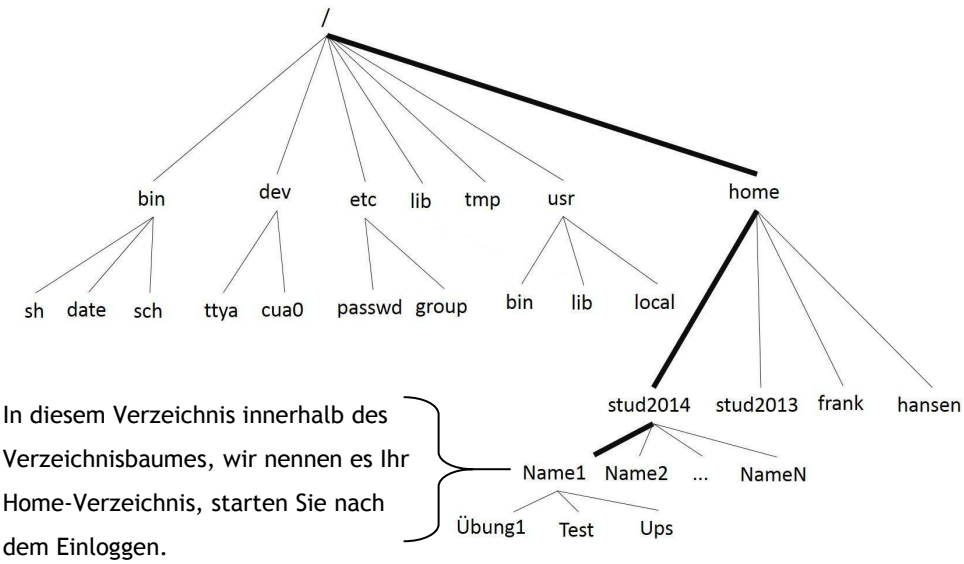
Geben Sie hierzu

man command

in die Eingabeaufforderung ein und ersetzen Sie ***command*** durch den Befehl, über den Sie mehr erfahren möchten.

Zum Verständnis der nächsten Kommandozeilenbefehle sind noch ein paar Informationen zum Dateisystem erforderlich. Ein Dateisystem bestimmt, wie

Dateien auf einem Datenträger abgelegt werden. Das Dateisystem von Linux ist wie ein Baum organisiert:



Ausgehend von einem Wurzelverzeichnis "**/**" kann jedes Verzeichnis Dateien und weitere Verzeichnisse enthalten. Der Name einer Datei zusammen mit dem Pfad, der ausgehend vom Wurzelverzeichnis zu dieser Datei führt, ergibt also für jede Datei eine eindeutige Adresse, die bestimmt, wo sich diese Datei befindet.

Beispiel:

/home/hansen/my_file

Die Datei "***my_file***" befindet sich im Verzeichnis "***hansen***", welches ein Unterverzeichnis im Verzeichnis "***home***" ist, welches sich wiederum im Wurzelverzeichnis „**/**" befindet. Die Verzeichnis- bzw. Dateinamen werden hierbei durch einen Querstrick von einander getrennt.

Um herauszufinden, wo im Verzeichnisbaum Sie sich gerade befinden, gibt es den Befehl ***pwd***, was für „print working directory" steht. Geben Sie diesen Befehl in die Eingabeaufforderung ein und führen Sie Ihn aus (Enter-Taste).

(2) Wechseln Sie jetzt in Ihr Home-Verzeichnis:

cd ~

Das Kommando ***cd*** steht für "change directory" und erlaubt es Ihnen, in ein anderes Verzeichnis zu wechseln. Die Tilde (~) ist eine Abkürzung für den Dateipfad zu Ihrem Home-Verzeichnis ***/home/stud2014/ihr_benutzername***.

(3) Überprüfen Sie, ob Sie sich wirklich in Ihrem Home-Verzeichnis befinden:

pwd

(4) Lassen Sie sich nun die Dateien anzeigen, die sich in Ihrem Home-Verzeichnis befinden:

ls

(5) Einige Dateien sind versteckt. Das sind die Dateien, deren Dateinamen mit einem Punkt beginnen. Wenn Sie wollen, dass Ihnen der Befehl ***ls*** auch diese versteckten Dateien anzeigt, müssen Sie die Option ***-a*** verwenden:

ls -a

(6) Falls Sie sich nicht nur für die Dateinamen interessieren, sondern zusätzlich noch weitere Informationen sehen möchten, z.B. zu Zugriffsrechten, Dateibesitzer und Dateigröße, dann können Sie die Option ***-l*** verwenden:

ls -l

(7) Mehrere Optionen lassen sich auch kombinieren. Wenn Sie das ausführliche Ausgabeformat wünschen UND gleichzeitig auch versteckte Dateien angezeigt bekommen möchten, dann schreiben Sie

ls -l -a

oder besser einfach

ls -la oder ***ls -al*** ,

denn mehrere Optionen können einfach auch ohne Trennzeichen in beliebiger Reihenfolge hintereinander geschrieben werden.

(8) Wie bereits erläutert, können Sie mit dem Befehl ***cd*** im Verzeichnisbaum navigieren. Hierzu geben Sie als Eingabeargument den Pfad an, in den Sie wechseln möchten, z.B.

cd /usr/local ,

um in das Verzeichnis ***/usr/local*** zu wechseln.

Es wird hierbei zwischen relativen und absoluten Pfaden unterschieden. Ein absoluter Pfad enthält das Wurzelverzeichnis **/** und bezieht sich immer auf dasselbe Verzeichnis, unabhängig davon, wo im Verzeichnisbaum man sich als Benutzer gerade befindet. Bei einem relativen Pfad ist das nicht der Fall.

Hier ein Beispiel: Der absolute Pfad zu ihrem Desktop-Verzeichnis lautet ***/home/stud2014/ihr_benutzername/Desktop***. Nehmen wir an Sie befinden sich gerade in Ihrem Home-Verzeichnis, dann ist der relative Pfad zu Ihrem Desktop-Verzeichnis einfach nur ***Desktop***.

Laufwerke wie C: oder D: wie unter einem Windows-System gibt es unter Linux übrigens nicht. Alles ist dem Wurzelverzeichnis untergeordnet.

(9) Befinden Sie sich noch in Ihrem Home-Verzeichnis? Falls nicht, wechseln Sie bitte dorthin zurück.

(10) Wechseln Sie jetzt im Verzeichnisbaum eine Ebene nach oben:

cd ..

Mit Ausnahme des Wurzelverzeichnisses gibt es zu jedem Verzeichnis genau ein übergeordnetes Verzeichnis. Die beiden Punkte .. sind ein relativer Pfad zu dem Verzeichnis, das dem Verzeichnis übergeordnet ist, in dem Sie sich gerade befinden.

(11) Wechseln Sie ins Wurzelverzeichnis:

cd /

(12) Wechseln Sie jetzt wieder zurück in Ihr Home-Verzeichnis:

cd ~

(13) Legen Sie in Ihrem Home-Verzeichnis ein Verzeichnis mit dem Namen ***linuxcourse*** an:

mkdir linuxcourse

(14) Wechseln Sie in dieses Verzeichnis:

cd linuxcourse

(15) Erstellen Sie im Verzeichnis ***linuxcourse*** ein Unterverzeichnis mit dem Namen ***test***:

mkdir test

(16) Lassen Sie sich alle Dateien inklusive aller versteckten Dateien im ***linuxcourse/test*** Unterverzeichnis anzeigen:

ls -a test

Die beiden Einträge, die Sie anschließend sehen, „.“ und „..“, sind sogenannte symbolische Verknüpfungen. Auch ein leeres Verzeichnis wie das gerade von Ihnen erstellte hat mindestens diese beiden Einträge. Die zwei Punkte „..“ haben Sie bereits kennengelernt. Sie verweisen auf das übergeordnete Verzeichnis eines Verzeichnisses. Der einzelne Punkt „.“ in einem Verzeichnis verweist dagegen auf

das Verzeichnis selbst, in diesem Fall also auf ***linuxcourse/test*** in ihrem Home-Verzeichnis.

(17) Löschen Sie das ***test*** Unterverzeichnis wieder:

rmdir test

VORSICHT: Wann immer Sie unter Linux etwas löschen, wird es endgültig gelöscht. Wenn Sie mit der Kommandozeile arbeiten, gibt es keinen Papierkorb (wie unter Windows) oder eine andere Art der Datenwiederherstellung.

(18) Benutzen Sie jetzt die Befehle ***cd*** und ***ls***, um sich ein wenig innerhalb des Verzeichnisbaumes zu bewegen. Zeichnen Sie nebenher einen Verzeichnis-Baum, der so ähnlich aussieht wie der auf Seite 3 dieser Übung.
Hinweis: In einige Verzeichnisse können Sie nicht wechseln, da Ihnen dafür die notwendige Berechtigung fehlt.

(19) Innerhalb des Wurzelverzeichnisses **/** befindet sich das Verzeichnis ***/etc***. Wir wollen jetzt in Erfahrung bringen, welche Zugriffsrechte Sie für dieses Verzeichnis haben. Lassen Sie sich den Inhalt des Wurzelverzeichnisses im ausführlichen Format anzeigen:

ls -l /

Suchen Sie innerhalb der Ausgabe eine Zeile, die in etwa wie die folgende aussieht:

drwxr-xr-x 95 root root 8192 2009-03-27 12:45 etc

Welche Zugriffsrechte Sie für das Verzeichnis ***etc*** besitzen, sagt Ihnen die Angabe ***drwxr-xr-x*** gleich zu Beginn der Zeile:

Das "***d***" bedeutet lediglich, dass es sich bei ***etc*** um ein Verzeichnis handelt. Die nächsten drei Zeichen (***rwx***) geben die Rechte des Besitzers dieses Verzeichnisses an. Hierbei steht

- r für Lesezugriff (read)
- w für Schreibrechte (write) und
- x für das Recht, eine Datei auszuführen (execute)

Bei einem Verzeichnis bedeuten Ausführungsrechte, dass in dieses Verzeichnis gewechselt werden darf.

Die nächsten drei Zeichen (***r-x***) sind die Zugriffsrechte für die Gruppe, in der sich der Dateibesitzer befindet. Das Minuszeichen bedeutet, dass die entsprechende Berechtigung fehlt. Benutzer in derselben Gruppe wie der Besitzer des Verzeichnisses ***etc*** haben also keine Schreibrechte für dieses Verzeichnis.

Die letzten drei Zeichen (***r-x***) sind die Zugriffsrechte für alle anderen Benutzer.

Die Angabe ***root root*** bedeutet, dass Nutzer ***root*** in der Gruppe ***root*** Besitzer des Verzeichnisses ***/etc*** ist.

Finden Sie jetzt heraus, welche Zugriffsrechte Sie für das Verzeichnis ***/etc*** haben. Hierzu müssen Sie wissen, wie ihr Benutzername lautet und zu welcher Gruppe Sie gehören. Mit dem Befehl ***id*** können Sie Ihren Benutzernamen in Erfahrung bringen, falls Sie diesen vergessen haben. Über den Befehl ***groups*** erfahren Sie, zu welcher Gruppe Sie gehören.

Offensichtlich sind Sie weder der Benutzer ***root*** noch in der Gruppe ***root***. Die Zugriffsrechte, die für Sie gelten, sind also die Rechte der „anderen Benutzer“ (siehe oben).

(20) Welche Zugriffsrechte haben Sie für die Datei ***/etc/passwd***?

(21) Lassen Sie sich alle Verzeichnisse im Verzeichnis ***/home*** anzeigen. Können Sie sich vorstellen, was das für Verzeichnisse sind? Wer hat wohl die Zugriffsrechte für diese Verzeichnisse und was bedeutet das?

(22) Es soll nun das Verzeichnis ***/etc*** in ein Unterverzeichnis ***linuxcourse/etc*** in Ihrem Home-Verzeichnis kopiert werden. Beachten Sie hierbei, in welchem Verzeichnis innerhalb des Verzeichnisbaumes Sie sich aktuell befinden und passen Sie das folgende Kommando entsprechend Ihrer Situation an:

cp -r Quelle Ziel

Wenn Sie sich beispielsweise gerade in Ihrem Home-Verzeichnis befinden, dann ersetzen Sie ***Quelle*** mit ***/etc*** und ***Ziel*** mit ***linuxcourse***. Falls Sie sich dagegen im Verzeichnis ***linuxcourse*** befinden, dann schreiben Sie als Ziel einfach einen Punkt (siehe 16). Im Wurzelverzeichnis können Sie auch den relativen Pfad ***etc*** als ***Quelle*** einsetzen. Sollten Sie sich irgendwo anders befinden, dann verwenden Sie am besten die absoluten Pfade für ***Quelle*** und ***Ziel***.

Lassen Sie sich nicht davon beirren, wenn während des Kopiervorganges ein paar Fehlermeldungen erscheinen. Sie haben nicht für alle Dateien im Verzeichnis ***/etc*** die Leserechte und diese Dateien können Sie dann auch nicht kopieren.

(23) Wie Sie inzwischen gelernt haben, ist für jede Datei und jedes Verzeichnis eindeutig festgelegt, welche Benutzer welche Zugriffsrechte haben. Es bestehen separate Zugriffsrechte für den Besitzer der Datei selbst (***u***), für Mitglieder seiner Gruppe (***g***) und für alle anderen Nutzer (***o***). Zugriffsrechte sind allerdings nicht in

Stein gemeißelt sondern dürfen vom Besitzer einer Datei (und nur von diesem) auch verändert werden. Dies ist mit dem Befehl ***chmod*** möglich.

Wir wollen jetzt die Zugriffsrechte für das Verzeichnis ***~/linuxcourse/etc*** ändern und Sie gleichzeitig mit ein paar Tricks vertraut machen, welche Ihnen dabei helfen werden, längere Befehle schneller einzugeben.

Geben Sie sich selbst zunächst Lese- und Schreibrechte:

chmod -R u+rw ~/linuxcourse/etc

Mit dem nächsten Befehl

chmod -R g+r-w ~/linuxcourse/etc (Bitte noch nicht eingeben!)

werden Sie Benutzern in Ihrer Gruppe Lese-, aber keine Schreibrechte einräumen. Schreiben Sie von diesem Befehl zunächst nur ***chmod -R g+r-w ~/li*** und drücken Sie dann die Tabulator-Taste  . Wie Sie sehen, wird der Pfad ***~/li*** automatisch zu ***~/linuxcourse/*** erweitert. Dies ist die sogenannte Autovervollständigung. In Ihrem Home-Verzeichnis gibt es nämlich zurzeit kein weiteres mit den Buchstaben ***li*** beginnendes Verzeichnis (und auch keine Datei). Somit ist es eindeutig, was Sie meinen, und der Pfad kann automatisch vervollständigt werden. Wenn Sie die Tabulator-Taste noch ein weiteres Mal drücken, wird der Pfad zu ***~/linuxcourse/etc/*** verlängert, da sich außer dem Unterverzeichnis ***etc*** nichts weiter im Verzeichnis ***~/linuxcourse/*** befindet und somit auch hier wieder eindeutig ist, was Sie meinen.

Wir wollen jetzt noch die Zugriffsrechte aller anderen Benutzer für das Verzeichnis ***~/linuxcourse/etc*** anpassen und Ihnen an dieser Stelle auch den Kommandozeilenspeicher vorstellen. Dieser enthält die Kommandozeilenbefehle, die Sie zu einem früheren Zeitpunkt eingegeben haben. Mit den Pfeiltasten können Sie die Liste dieser Befehle durchsuchen. Drücken Sie jetzt die Pfeiltaste  einmal und ändern Sie den in der Eingabeaufforderung erscheinenden Befehl so ab, dass er dem folgenden gleicht:

chmod -R o-rwx ~/linuxcourse/etc

Die Option ***-R*** steht für "rekursiv". Das bedeutet, dass der Befehl auch für alle Dateien und Verzeichnisse angewendet wird, die dem Verzeichnis ***~/linuxcourse/etc*** untergeordnet sind.

Überprüfen Sie nun, ob sie bei der Änderung der Zugriffsrechte erfolgreich waren:

ls -l

(24) Kopieren Sie die Datei ***/etc/group*** in das ***linuxcourse*** Verzeichnis in Ihrem Home-Verzeichnis.

(25) Benennen Sie im Verzeichnis ***linuxcourse*** die Datei ***group*** in ***testfile*** um:

mv group testfile

Vergessen Sie hierbei nicht, wie in Aufgabe (22) die Pfade entsprechend anzupassen.

(26) Kopieren Sie das Verzeichnis ***linuxcourse*** mit all seinen Unterverzeichnissen in ein Verzeichnis ***work*** in Ihrem Home-Verzeichnis.

(27) Entfernen Sie das Verzeichnis ***linuxcourse*** inklusive aller Unterverzeichnisse:

rm -r linuxcourse

(28) Benennen Sie das Verzeichnis ***work*** in ***linuxcourse*** um.

(29) Ändern Sie die Zugriffsrechte für das Verzeichnis ***linuxcourse***. Sie selbst sollten alle Zugriffsrechte haben (Lese-, Schreib- und Ausführungsrechte), Benutzer in Ihrer Gruppe nur die Ausführungsrechte und alle anderen Benutzer gar keine Zugriffsrechte.

(30) Ändern Sie jetzt auch die Zugriffsrechte für alle Dateien und Unterverzeichnisse innerhalb des ***linuxcourse*** Verzeichnisses. Geben Sie sich selbst Lese- und Schreibrechte. Leute in Ihrer Gruppe sollten nur Leserechte und wie zuvor alle anderen Benutzer gar keine Zugriffsrechte haben. Da Sie sicher wenig Interesse daran haben, alle Dateinamen einzeln von Hand einzugeben, können Sie sich diese Aufgabe mit Hilfe eines sogenannten Wildcard-Zeichens erheblich erleichtern. Eine Wildcard ist eine Art Joker unter den Zeichen, welche Platzhalter für eine Gruppe von anderen Zeichen oder Zeichenketten ist. Das Sternchen ***** beispielsweise steht für eine beliebig lange Zeichenkette.

(31) Geben Sie mit Hilfe des Befehls ***echo*** Ihren Namen auf dem Bildschirm aus.
Ein Zeilenumbruch sollte vor und nach ihrem Namen erscheinen:

echo "\nVorname Nachname\n"

(32) Lassen Sie sich mit Hilfe des Befehls ***cat*** den Inhalt der Dateien ***/etc/group*** und ***/etc/passwd*** anzeigen:

cat /etc/group

(33) Gehen Sie jetzt in Ihr Home-Verzeichnis und verwenden Sie den folgenden Befehl:

ls -lRF /usr > usrfiles.txt

Stellen Sie sicher, dass Sie diesen Befehl genau so eingeben, wie er hier steht, und achten Sie dabei auch auf die Leerzeichen. Sie werden einige Meldungen mit der Nachricht “permission denied” erhalten. Diese Meldungen dürfen Sie ignorieren. Der Befehl, den Sie gerade verwendet haben, erzeugt eine Datei mit dem Namen ***usrfiles.txt*** in Ihrem Home-Verzeichnis. Diese Datei enthält nun eine sehr lange Liste von Dateien, die sich auf ihrem Rechner befinden. Die Ausführung dieses Befehls kann eine Weile dauern. Das Größer-als-Zeichen **>** innerhalb dieser Anweisung sorgt dafür, dass die Ausgabe des ***ls*** Befehls, die eigentlich auf Ihrem Bildschirm erscheinen würde, stattdessen in eine neu erstellte Datei ***usrfiles.txt*** umgeleitet wird. Falls sich bereits eine Datei mit demselben Namen in Ihrem Home-Verzeichnis befindet, wird diese überschrieben. Falls Sie das nicht wünschen, dann verwenden Sie anstelle eines einzelnen Größer-als-Zeichens zwei:

ls -lRF /usr >> usrfiles.txt

Jetzt würde eine bereits bestehende Datei ***usrfiles.txt*** nicht einfach überschrieben werden, sondern die abzuspeichernde Ausgabe einfach am Ende dieser Datei ergänzt werden.

Nutzen Sie anschließend den Befehl ***less***, um sich den Inhalt der Datei seitenweise anzeigen zu lassen.

less usrfiles.txt

Mit der Leertaste können Sie eine Seite nach vorne blättern, mit den Pfeiltasten ist zeilenweise scrollen nach oben oder unten möglich. Wenn Sie ein bestimmtes Wort suchen, zum Beispiel in unserem Fall den Teil eines Dateinamens, der in unserer Liste vorkommen soll, dann schreiben Sie einfach ***/suchwort*** und drücken anschließend Enter. Damit springen Sie zu der Position, an der der gesuchte Teilstring ***suchwort*** zum ersten Mal auftritt. Mit „n“ springen Sie zur nächsten Position, mit „N“ zur vorherigen. Durch drücken der Taste „q“ kommen Sie zurück zur Kommandozeile.

(34) Lassen Sie sich die ersten 10 Zeilen der Datei ***usrfiles.txt*** anzeigen:

head usrfiles.txt

Eine von 10 abweichende Zeilenanzahl der Ausgabe muss über die Optionen festgelegt werden.

Lassen Sie sich die letzten 5 Zeilen der Datei ***usrfiles.txt*** anzeigen:

tail -5 usrfiles.txt

(35) Verschieben Sie ***usrfiles.txt*** in das Verzeichnis ***linuxcourse***

mv usrfiles.txt ~/linuxcourse/

(36) Anstelle der Weiterleitung an eine Datei kann eine Programmausgabe auch als Eingabe für ein anderes Programm verwendet werden. Hierfür nutzen wir das sogenannte Pipe-Symbol **|**. Wenn wir beispielsweise wissen wollen, welche 10 Dateien in unserem Home-Verzeichnis als letztes bearbeitet wurden, dann können wir den folgenden Befehl verwenden:

ls -Rlat ~/* | head

Die Option ***-t*** des ***ls*** Befehls bewirkt, dass die Dateien nach ihrem Zeitstempel sortiert werden. Die zuletzt modifizierten Dateien werden daher zuerst aufgeführt.

Durch den verwendeten Pipe-Operator wird die Ausgabe des Befehls ***ls*** zur Eingabe des Befehls ***head***.

(37) Wenn wir nicht alle Zeilen der Datei ***usrfiles.txt*** angezeigt bekommen möchten, sondern nur solche, die eine bestimmte Zeichenkette (z.B. „matrix“) enthalten, dann können wir den Befehl ***grep*** verwenden:

grep matrix usrfiles.txt | less

Durch den verwendeten Pipe-Operator wird die Ausgabe von ***grep*** direkt an das Programm ***less*** weiter geleitetet.

(38) Um zum Ende des ersten Abschnitts dieser Übung noch ein wenig Bioinformatik zu betreiben, soll nun eine DNA-Sequenz, z.B. ATGTTGAGCCTCGCCTAG, in eine sequenzgleiche RNA übersetzt werden.

Wechseln Sie zunächst in Ihr ***linuxcourse*** Verzeichnis und erstellen Sie mit Hilfe des Befehls ***cat*** eine Eingabedatei, welche ihre DNA-Sequenz enthält:

cat > dna.txt

Nach Ausführen dieses Befehls wartet cat auf eine Eingabe. Geben Sie jetzt Ihre DNA-Sequenz in Großbuchstaben über die Tastatur ein und drücken Sie anschließend Enter. Drücken Sie Ctrl+D, um ***cat*** zu beenden. Überprüfen Sie anschließend mit den Befehlen ***ls*** und ***less***, ob eine Datei namens ***dna.txt*** erstellt wurde und ihre DNA-Sequenz enthält.

Um die gewählte DNA-Sequenz in eine RNA-Sequenz zu übersetzen, müssen wir eine Ersetzungsregel festlegen, nach der Thymin gegen Uracil ausgetauscht wird. Hierfür wenden wir den Befehl ***tr*** auf die Datei ***dna.txt*** an und speichern die Ausgabe in einer neuen Datei namens ***rna.txt***:

tr T U < dna.txt > rna.txt

Mit dem Kleiner-als-Zeichen wird der Inhalt der Datei ***dna.txt*** als Eingabe für den Befehl ***tr*** verwendet. Überprüfen Sie mit ***ls***, ob die neue Datei existiert, und lassen Sie sich mit ***less*** ihren Inhalt anzeigen. Sehen Sie eine RNA-Sequenz?

Löschen Sie anschließend die Dateien ***dna.txt*** und ***rna.txt***.

(39) Häufig kommt es vor, dass Sie eine bestimmte Abfolge von Kommandozeilenbefehlen mehrfach hintereinander verwenden möchten. Anstatt diese Befehle dann mühsam immer wieder von Neuem einzutippen, können Sie auch ein sogenanntes Shell-Script erstellen. Ein Shell-Script ist eine Textdatei, in welcher die Kommandozeilenbefehle stehen, die Sie verwenden möchten. Wir wollen die letzte Aufgabe nun noch einmal wiederholen und dafür ein solches Shell-Script erstellen. Legen Sie zunächst eine Datei mit dem Namen ***my_script*** an:

touch my_script

Öffnen Sie diese Datei mit einem Texteditor:

kwrite my_script &

Tragen Sie jetzt die folgenden Befehlszeilen ein und speichern Sie anschließend ab:

```
#!/bin/sh
echo Eingegebene DNA-Sequenz: $1
echo Die sequenzgleiche RNA lautet:
echo $1 | tr T U
```

Die erste Zeile des Shell-Scripts (***#!/bin/sh***) brauchen Sie jetzt nicht weiter zu beachten. Sie gibt lediglich an, dass es sich bei dieser Datei um ein Shell-Script handelt, welches mit der Standard-Unix-Shell ausgeführt werden soll.

$1 ist ein Platzhalter für Ihre DNA-Sequenz, die Sie als Eingabeargument an das Shell-Script weitergeben können. Wie bei anderen Kommandozeilenbefehlen, können Sie ein Eingabeargument einfach direkt hinter den Befehl schreiben:

./my_script ACGTUCGTUCC…

Wenn Sie diesen Befehl jetzt in die Eingabeaufforderung eingeben und ausführen, werden Sie allerdings die Fehlermeldung „***Permission denied***“ erhalten, da Ihnen bisher die Rechte fehlen, um diese Datei auszuführen. Ändern Sie daher zunächst die Zugriffsrechte für diese Script-Datei

chmod +x my_script

und führen Sie Ihr Shell-Script erst danach aus.

Teil II (Graphische Benutzerschnittstelle):

(40) Rufen Sie das Programm *Konqueror* über das Menü der Taskleiste auf. *Konqueror* ist ein Dateimanager und Webbrowser. (Falls Sie irgendwelche Hilfe bei der Nutzung eines bestimmten Programms benötigen, dann schauen Sie im Hilfe-Menü nach oder drücken Sie F1.)

(41) Wählen Sie auf der Startseite von *Konqueror* "Home Folder"  aus. Damit gelangen Sie direkt in Ihr Home-Verzeichnis.

(42) Die sogenannte "Location Bar" befindet sich unterhalb des Programm-Menüs von *Konqueror* und zeigt Ihnen das Verzeichnis an, in dem Sie sich gerade befinden. Überprüfen Sie, ob Sie sich in Ihrem Home-Verzeichnis befinden.

(43) Im *Konqueror*-Hauptfenster werden die Dateien des aktuellen Verzeichnisses angezeigt. Über das Programm-Menü von *Konqueror* gelangen Sie zu einer detaillierten Verzeichnisansicht:

View→View Mode→Details

(44) In dieser Ansicht werden nicht leere Verzeichnisse durch ein Größer-als-Zeichen vor dem Verzeichnisnamen kenntlich gemacht. Durch Klicken auf dieses Symbol können Sie sich auch die untergeordneten Dateien und Verzeichnisse anzeigen lassen und auf diese Art durch den Verzeichnisbaum navigieren. Ebenfalls möglich ist es, die in der "Location Bar" angegebene Adresse anzupassen.

(45) Zum Schluss wollen wir die Übersetzung einer DNA- in eine RNA-Sequenz aus Aufgabe (38) ein weiteres Mal durchführen. Anstelle der Kommandozeile werden wir diesmal allerdings einen Texteditor verwenden.

Gehen Sie zunächst in Ihr ***linuxcourse*** Verzeichnis und erstellen Sie eine Eingabe-Datei, welche die DNA-Sequenz enthält. Wählen Sie hierfür ***Create New→Textfile*** im Kontextmenü aus und nennen Sie die neue Datei ***dna.txt***.

(46) Öffnen Sie über das Kontextmenü die gerade erstellte Datei ***dna.txt***:

Open with→KWrite

Wie unter einem Windows-System gelangen Sie mit einem Rechtsklick auf eine Datei an das Kontextmenü dieser Datei.

Geben Sie nun die DNA-Sequenz ein und fügen Sie am Ende der Sequenz durch Drücken der Enter-Taste noch einen Zeilenumbruch ein. Speichern Sie Ihre Sequenz über das Menü (***File→Save)*** oder über die Tastatur (Ctrl+S) ab.

(47) Um Ihre Sequenz in RNA zu übersetzen, wählen wir im *KWrite*-Menü ***Edit→Replace…*** aus und geben „T“ in das Eingabefeld „Find“ und „U“ in das Eingabefeld „Replace“ ein. Durch Klicken auf „Next“ wird das erste Vorkommen von „T“ hervorgehoben. Mit „Replace“ ersetzen Sie es durch ein „U“. Ersetzen Sie alle Thymin-Nukleinbasen ihrer DNA-Sequenz durch Uracil.

(48) Speichern Sie die geänderte Sequenz in einer neuen Datei unter dem Namen ***rna.txt*** ab.

(49) Schließen Sie den Text-Editor *KWrite* und schauen Sie sich den Inhalt von ***rna.txt*** an. Wenn Sie eine RNA-Sequenz sehen, dann haben Sie Ihre erste Bioinformatik-Übung erfolgreich absolviert.

Unix/Linux Command Reference

FOSSwire.com

File Commands

ls – directory listing
ls -al – formatted listing with hidden files
cd *dir* - change directory to *dir*
cd – change to home
pwd – show current directory
mkdir *dir* – create a directory *dir*
rm *file* – delete *file*
rm -r *dir* – delete directory *dir*
rm -f *file* – force remove *file*
rm -rf *dir* – force remove directory *dir* *
cp *file1 file2* – copy *file1* to *file2*
cp -r *dir1 dir2* – copy *dir1* to *dir2*; create *dir2* if it doesn't exist
mv *file1 file2* – rename or move *file1* to *file2*
if *file2* is an existing directory, moves *file1* into directory *file2*
ln -s *file link* – create symbolic link *link* to *file*
touch *file* – create or update *file*
cat > *file* – places standard input into *file*
more *file* – output the contents of *file*
head *file* – output the first 10 lines of *file*
tail *file* – output the last 10 lines of *file*
tail -f *file* – output the contents of *file* as it grows, starting with the last 10 lines

Process Management

ps – display your currently active processes
top – display all running processes
kill *pid* – kill process id *pid*
killall *proc* – kill all processes named *proc* *
bg – lists stopped or background jobs; resume a stopped job in the background
fg – brings the most recent job to foreground
fg *n* – brings job *n* to the foreground

File Permissions

chmod *octal file* – change the permissions of *file* to *octal*, which can be found separately for user, group, and world by adding:

- 4 – read (r)
- 2 – write (w)
- 1 – execute (x)

Examples:
chmod 777 – read, write, execute for all
chmod 755 – rwx for owner, rx for group and world
For more options, see **man chmod**.

SSH

ssh *user@host* – connect to *host* as *user*
ssh -p *port user@host* – connect to *host* on port *port* as *user*
ssh-copy-id *user@host* – add your key to *host* for *user* to enable a keyed or passwordless login

Searching

grep *pattern files* – search for *pattern* in *files*
grep -r *pattern dir* – search recursively for *pattern* in *dir*
command* | grep *pattern – search for *pattern* in the output of *command*
locate *file* – find all instances of *file*

System Info

date – show the current date and time
cal – show this month's calendar
uptime – show current uptime
w – display who is online
whoami – who you are logged in as
finger *user* – display information about *user*
uname -a – show kernel information
cat /proc/cpuinfo – cpu information
cat /proc/meminfo – memory information
man *command* – show the manual for *command*
df – show disk usage
du – show directory space usage
free – show memory and swap usage
whereis *app* – show possible locations of *app*
which *app* – show which *app* will be run by default

Compression

tar cf *file.tar files* – create a tar named *file.tar* containing *files*
tar xf *file.tar* – extract the files from *file.tar*
tar czf *file.tar.gz files* – create a tar with Gzip compression
tar xzf *file.tar.gz* – extract a tar using Gzip
tar cjf *file.tar.bz2* – create a tar with Bzip2 compression
tar xjf *file.tar.bz2* – extract a tar using Bzip2
gzip *file* – compresses *file* and renames it to *file.gz*
gzip -d *file.gz* – decompresses *file.gz* back to *file*

Network

ping *host* – ping *host* and output results
whois *domain* – get whois information for *domain*
dig *domain* – get DNS information for *domain*
dig -x *host* – reverse lookup *host*
wget *file* – download *file*
wget -c *file* – continue a stopped download

Installation

Install from source:
./configure
make
make install
dpkg -i *pkg.deb* – install a package (Debian)
rpm -Uvh *pkg.rpm* – install a package (RPM)

Shortcuts

Ctrl+C – halts the current command
Ctrl+Z – stops the current command, resume with **fg** in the foreground or **bg** in the background
Ctrl+D – log out of current session, similar to **exit**
Ctrl+W – erases one word in the current line
Ctrl+U – erases the whole line
Ctrl+R – type to bring up a recent command
!! - repeats the last command
exit – log out of current session

* use with extreme caution.