



0. Übung zu „Modellierung/Programmierung“ Wintersemester 2012/2013

1. Kurze Einführung zu Linux-Shell

Auf den Rechnern des Fachbereichs ist das Betriebssystem Linux installiert. Da die meisten Hörer dieser Vorlesung keine Vorkenntnisse im Umgang mit diesem System haben, gibt es im Folgenden eine kurze Einführung hierfür.

Die wichtigste Regel lautet: **Das System beißt nicht !** Als normaler Benutzer kann man an diesen Rechnern nichts kaputt machen!

Das System hat zwar eine ähnliche graphische Oberfläche wie Windows, aber die meisten Dinge werden über die Kommandozeile, die sogenannte Shell, erledigt.

Achtung: Linux ist **case sensitive**, d. h. im Gegensatz zu Windows wird zwischen Groß- und Kleinschreibung unterschieden!

Hierzu öffnen Sie zuerst das Terminal. Hier ist eine Auflistung der wichtigsten Befehle, die Sie benötigen:

cp kopiert Dateien (copy)

cp Datei_1 Datei_2 erstellt eine Kopie von *Datei_1* und nennt diese *Datei_2*.

cd wechselt das Verzeichnis (change directory)

cd Neues_Verzeichnis wechselt vom aktuellen Verzeichnis nach *Neues_Verzeichnis*. Dabei kann *Neues_Verzeichnis* auch '.' oder '..' sein.

cd . bleibt im aktuellen Verzeichnis, d.h. es passiert nichts, *cd ..* wechselt in das übergeordnete Verzeichnis. Der Befehl *cd* ohne weitere Angaben wechselt in das Benutzerverzeichnis des jeweiligen Benutzers \$HOME.

mv verschiebt Dateien (move)

mv Datei_1 Datei_2 verschiebt die Datei *Datei_1* in die Datei *Datei_2*.

ls listet den Inhalt eines Verzeichnisses auf (list segments)

ls v gibt den Inhalt des Verzeichnisses *v* an. *ls* ohne weitere Angaben listet den Inhalt des aktuellen Verzeichnisses auf.

mkdir erstellt ein neues Verzeichnis (make directory) *mkdir v* erstellt ein neues Verzeichnis *v* im aktuellen Verzeichnis.

pwd gibt das aktuelle Verzeichnis an (print working directory)

rm löscht Dateien (remove)

rm v löscht die Datei *v*. **Dieses Löschen ist endgültig!**

Verzeichnisse können mit *rm -r Verzeichnis* gelöscht werden. Hierbei steht *-r* für recursive, d.h. es werden zunächst alle Dateien innerhalb des Verzeichnisses, dann das Verzeichnis selbst gelöscht.

rmdir löscht leere Verzeichnisse (remove directory)

man ruft die Hilfe auf (manual)

man befehl ruft die Hilfeseiten zu *befehl* auf. Hier ist eine kleine Warnung angebracht: Die Hilfe zu *ls* ist mit ca. 100 Optionen ziemlich umfangreich. Hiervon werden i.A. nur etwa 10 benötigt.

Bei der Angabe von Verzeichnissen ist der Slash '/' der Trenner, **nicht** der Backslash '\' wie unter Windows bzw. DOS.

1.1. Editoren

Sie benötigen einen Editor, um Programme zu schreiben. Auf den Rechnern des Fachbereiches sind *gedit* oder *emacs* zwei der installierten Editoren.

1.2. Internet

Der Befehl *firefox* öffnet den entsprechenden Browser. Mit dem Befehl *thunderbird* können Sie das gleichnamige E-Mail-Programm starten. Um Thunderbird einzurichten, benötigen Sie folgende Daten:

Name Ihr Name

E-Mail-Adresse Ihre Uni-E-Mail-Adresse (s9abcdef@stud.uni-sb.de)

Typ des Servers IMAP

Posteingangsserver sbustd.stud.uni-saarland.de

Postausgangsserver smtp.uni-saarland.de

Wenn Sie Ihre private E-Mail-Adresse einrichten möchten, wählen Sie als Server-Typ POP3. Den Namen des Postausgangsservers erfahren Sie i.A. auf den Webseiten der Anbieter. Der Posteingangsserver bleibt smtp.uni-saarland.de.

2. Aufgaben

1. Loggen Sie sich mit Ihrem Login und Passwort ein. Die Kennung erhalten Sie im Geschäftszimmer.
2. Legen Sie sich ein Verzeichnis namens *Uebungen* an. In dieses Verzeichnis können Sie die bearbeiteten Übungsaufgaben abspeichern.
3. Öffnen Sie den Editor, indem Sie ein Terminal öffnen und den Befehl `gedit &` eingeben.
4. Schreiben Sie ein Programm, das 'Hallo Welt' als Ausgabe hat. Speichern Sie die Datei unter '*hallo_welt.c*' im angelegten Verzeichnis *Uebungen* ab und kompilieren Sie dieses anschließend mittels

```
gcc hallo_welt.c -o hallo_welt
```

Führen Sie das kompilierte Programm anschließend mit dem Befehl '`./hallo_welt`' aus.