

Grundlagen der Informatik - 1. Praktikum

Organisatorisches

- Nutzerkennzeichen beim Operator besorgen
- zwischen Windows und Linux wählen (bevorzugt Linux):
 - Programmerstellung einfacher
 - Compilerverhalten durchsichtiger
 - verbesserte Prozess-Kontrollmöglichkeiten
- Merkblätter für Linux und Windows: <http://testserver.freebits.de/cpp/praktikum1/>

Einführung in Linux

Ressourcen

- Informationen im Netz:
URZ-Einführung für neue Nutzer:
<http://www.tu-chemnitz.de/urz/kurse/unterlagen/einfuehrung>
Linux-Benutzeroberfläche: <http://www.tu-chemnitz.de/urz/kurse/unterlagen/KDE3/>
C++ URZ Kurs-Unterlagen: <http://www.tu-chemnitz.de/urz/kurse/unterlagen/C++/>
- weitere Informationen im Skript Anhang D (auf S. 124 ist „cc“ mit „gcc“ zu ersetzen)

Login-Prozess

- falls Windows NT läuft, bitte „Strg+Alt+Entf“ drücken und „Neustarten“
- mit Nutzerkennzeichen und Passwort einloggen, „Sitzungsart“ auf „KDE“ stellen

Umgang mit der Linux-Kommandozeile

- viele verschiedene Terminaltypen verfügbar, wir nutzen hier „konsole“
- starten über: „Strg+T“ oder „Alt+F2“ und Eingabe „konsole“ oder „Start -> System -> Terminals -> Konsole“
- Beispielsitzung:

```
# Inhalt des Verzeichnisses anzeigen
ls -al
# ein Verzeichnis namens „cpp“ für zukünftige Programme erstellen
mkdir cpp
# ein Verzeichnis löschen
rm -r <verzeichnisname>
# Hilfe zu Programmen, Funktionen u.a. erhalten
man rm oder man g++
# Anzeige des aktuellen Verzeichnisses
pwd
# Anzeige laufender Prozesse
ps -aux
```

- als grafische Oberfläche zur Dateiverwaltung dient z.B. „Konqueror“
-> ausführen über: „Start -> System -> Dateiverwaltung -> Persönliches Verzeichnis“

Editoren unter Linux

- bekannte Vertreter sind „vim“ (siehe Skript Anhang D.4.), „emacs“, „kate“, „nedit“
- empfohlen für intuitive Nutzung mit Syntax-Highlighting wird „kate“:
-> öffnen über „Start -> Dienstprogramme -> Editor -> Kate“

Programmierbeispiel mit Compileranwendung

- beliebigen Texteditor öffnen
- den folgenden Quelltext eingeben:

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hallo Welt!" << endl;
    return 0;
}
```

- speichern im angelegten Verzeichnis „cpp“ als Datei „hello.cpp“
- ein Konsolen-Fenster öffnen und folgendes abarbeiten:

```
# ins Heimatverzeichnis wechseln, welches jeder Nutzer hat
cd
# ins Verzeichnis wechseln, welches die Datei „hello.cpp“ enthält
cd cpp
# Quelltext übersetzen und ausführbaren Programmcode erzeugen
g++ hello.cpp -o hello
# Programm ausführen
./hello
```

- bei Fehlerausgaben stets mit der Korrektur des ersten Fehlers beginnen, da er unter Umständen weitere erzeugt, die nicht direkt behebbar sind

Kontakt für Fragen

- Übungsleiter: Marcel Karras
- Email: toka@freebits.de
- Übungs- und Praktikumsunterlagen: <http://testserver.freebits.de/cpp/>