

1. Übung zur Vorlesung „Programmierung“

Serie 1

Abgabe am Freitag, 29. Oktober – 12:00

Informationen zur Vorlesung, zu den Übungen und zur Programmiersprache Scheme können Sie im Internet unter den folgenden Adressen finden:

- <http://www.informatik.uni-kiel.de/~mh/lehre/Inf-Prog-WS10/>
- <http://www.informatik.uni-kiel.de/prog/mitarbeiter/fabian-reck/lehre/programmierung-10/>

Die Aufgaben sollen in Zweiergruppen bearbeitet und abgegeben werden. Bitte suchen Sie sich möglichst frühzeitig einen Übungsgruppenpartner in der gleichen Übungsgruppe. Sobald Sie von uns Ihre Zugangsdaten für die elektronische Übungsverwaltung *iLearn* bekommen haben, können Sie sich zu einer Kleingruppe zusammenschließen.

Präsenzaufgaben werden in den Übungsgruppen besprochen und sollen vorbereitet aber nicht abgegeben werden.

Beachten Sie, dass die Übungen ab dem 25.10.2008 beginnen.

1 Die wichtigsten UNIX-Befehle

1.1 Allgemein

- **man** *command* - Anzeigen einer Manual-Page zu *command*
- **lpr** *filename* - Ausdrucken der Datei *filename*
- **lpq** - Anzeigen der aktuellen Drucker-Warteschlange
- **lprm** *jobid* - Entfernen eines Druckauftrages aus der Drucker-Warteschlange
- **echo** *arg₁ ... arg_n* - Ausgaben der Strings *arg₁ ... arg_n*
- **cat** *filename* - Ausgeben der Datei *filename*
- **grep** *pattern files* - Suchen nach dem Muster *pattern* in den *files*
- **diff** *file file* - Vergleich zweier Textdateien
- **du** - Ausgabe des Speicherbedarfs von Dateien

1.2 Informationen über Prozesse und Benutzer im System

- **users** - eingeloggte Benutzer auf dem Rechner
- **rusers** - eingeloggte Benutzer auf allen Rechnern
- **id** - persönliche Benutzerinformationen
- **who** - Anzeigen aller aktiven Shells
- **w** - wie **who**, jedoch zusätzlich Anzeige der laufenden Programme
- **w user** - wie **w**, jedoch eingeschränkt auf den Benutzer *user*
- **ps** - Anzeigen aller eigenen Shells zugeordneten Prozesse
- **ps -x** - Anzeigen aller eigenen Prozesse
- **ps -aux** - Anzeigen aller Prozesse mit jeweiligem Besitzer

1.3 Arbeiten mit Prozessen

- *command* - Starten eines Benutzerprogramms *command*
- *command arg₁ ... arg_n* - Starten eines Benutzerprogrammms mit Parametern *arg₁ ... arg_n*
- **^C** - Beenden eines Programms
- **^Z** - Stoppen eines Programms
- *command &* - Starten eines Benutzerprogramms im Hintergrund
- **fg**, **bg** - gestoppte Programme weiterlaufen, bzw. im Hintergrund weiterlaufen lassen
- **jobs** - Anzeige aller im Hintergrund laufenden Programme einer Shell
- **fg job** - Hintergrundprozesse im Vordergrund laufen lassen
- **kill pid ...** - Beenden eines Prozesses *pid*
- **kill -STOP pid ...** - Stoppen eines Prozesses *pid*

1.4 Das Unix-Filesystem

- **cd directory** - Wechseln in das Verzeichnis *directory*
- **cp srcfile destfile** - Kopieren des *srcfiles* nach *destfile*
- **mv oldname newname** - Umbenennen von *oldname* nach *newname*
- **mkdir directory** - Anlegen des Verzeichnisses *directory*
- **rmdir directory** - Löschen des Verzeichnisses *directory*
- **pwd** - Anzeigen des aktuellen Verzeichnisses
- **ls** - Anzeigen des Inhaltes des aktuellen Verzeichnisses
- **ls directory** - Anzeigen eines Verzeichnisses *directory*
- **ls -l** - Anzeigen eines Verzeichnisses im Langformat
- **chmod mods file ...** - Verändern der Rechte einer Datei des Verzeichnisses *file*; *mods* ist entweder eine Oktalkodierung (s. **ls -s**) oder von der Form **ugo±rwx**
- **ln -s sourcefile target** - Einrichten eines Links (auch für Verzeichnisse)

Präsenzaufgabe 1 - Scheme-Ausdrücke

Fügen Sie in den folgenden Ausdrücken Klammern so ein, dass gültige Scheme-Ausdrücke entstehen.

1. 4 2

2. + 17 4

3. * + 0 8 15

4. * + 5 * 2 3 2

5. * * + 4 3 2 * + 1 2 * 6 + 1 3

Präsenzaufgabe 2 - Algorithmus

Formulieren Sie einen Algorithmus, der in einer Folge von arithmetischen Operatoren und Zahlen so Klammern einfügt, dass das Ergebnis ein gültiger Scheme-Ausdruck ist. Diskutieren Sie daran die Begriffe partielle und totale Korrektheit.