Ludwig-Maximilians-Universität München
Institut für Informatik
Lehrstuhl für Mobile und Verteilte Systeme
Prof. Dr. Claudia Linnhoff-Popien

mobile and distributed systems group

Betriebssysteme im Wintersemester 2017/2018
Übungsblatt 1

Abgabetermin: 30.10.2017, 18:00 Uhr

Besprechung: Besprechung der T-Aufgaben in den Tutorien vom 23. – 27. Oktober 2017
Besprechung der H-Aufgabe in den Tutorien vom 30. Oktober – 03. November 2017

Ankündigungen: Herzlich willkommen zum Übungsbetrieb zur Vorlesung Betriebssysteme. Bitte melden Sie sich zu den Übungsgruppen unter `https://uniworx.ifi.lmu.de/` an. Beachten Sie dazu die Hinweise auf dem Merkblatt.
Der Stoff dieses Übungsblattes dient zur Einarbeitung in das Betriebssystem Linux.

Aufgabe 1: (T) Einstieg in Linux

(– Pkt.)

Diese Aufgabe soll Ihnen den Einstieg in den Umgang mit der Linux-Shell erleichtern, falls Sie bisher nur wenig Erfahrung damit gesammelt haben. Da die Aufgabe zum Selbststudium konzipiert wurde, wird sie in der Übungsgruppe nicht im Detail besprochen. Experten können diese Aufgabe ignorieren. ;-)

Nach dem Anmelden am CIP-Rechner:

- Je nachdem, welchen Desktop-Manager Sie eingestellt haben, sehen Sie z.B. die grafische Oberfläche Gnome oder KDE. Informieren Sie sich zunächst, wie Sie unter der entsprechenden Oberfläche ein Programm starten und starten Sie nun eine Konsole (Shell).
- Die Konsole zeigt Ihnen an, in welchem Verzeichnis Sie sich gerade befinden (im Normalfall ist das zu Beginn Ihr Home-Verzeichnis) und stellt einen Cursor zur Verfügung, so dass Sie jetzt Befehle eingeben können.

Erste Schritte mit der Linux-Shell:

a. Geben Sie den Befehl `ls` ein (und bestätigen Sie mit Enter). Sie sehen eine Auflistung der Dateien und Ordner in Ihren Home-Verzeichnis. Selbst wenn das Verzeichnis leer ist, sehen Sie dort immer die Einträge `.` und `..`, bei denen es sich um Links auf das aktuelle Verzeichnis selbst und auf das Oberverzeichnis handelt.

b. Geben Sie `mkdir neu` ein, um ein Unterverzeichnis mit dem Namen `neu` zu erstellen. Wechseln Sie mit `cd neu` in dieses Verzeichnis und mit `cd ..` von dort aus wieder zurück in Ihr Home-Verzeichnis.

c. Mit dem Befehl `echo` lassen sich beliebige Zeichenketten auf der Konsole ausgeben. Testen Sie das mit `echo Betriebssysteme`.

d. Geben Sie nun `echo Betriebssysteme > test.txt` ein. Die Ausgabe der Zeichenkette erfolgt jetzt nicht mehr direkt in der Konsole, sondern wird in eine Datei umgeleitet. Es wird also im aktuellen Verzeichnis eine neue Datei mit dem Namen `test.txt` angelegt, die den Text `Betriebssysteme` enthält.

e. Verwenden Sie den Befehl `ls` erneut, um sich zu überzeugen, dass sich die Datei `test.txt` nun tatsächlich in Ihrem Home-Verzeichnis befindet.

f. Zu nahezu jedem Befehl lassen sich spezielle zusätzliche Parameter angeben, die Einfluss auf die Wirkung des Befehls haben. Solche Parameter werden Flags genannt und beginnen mit einem Minus-Zeichen. Geben Sie folgendes ein: `ls -l`. Der Befehl `ls` wird nun also mit dem Parameter (Flag) `-l` aufgerufen. Wie Sie sehen, wird die Ausgabe wesentlich ausführlicher als zuvor.

g. Aber welche weiteren Flags gibt es zum `ls`-Befehl? Zu jedem Shell-Befehl lässt sich eine Hilfe-Seite (Manual) anzeigen. Rufen Sie die Hilfe-Seite zum Befehl `ls` auf, indem Sie `man ls` eingeben. Mit den Pfeil-Tasten und der Bild-auf- und Bild-ab-Taste können Sie im Text scrollen. Suchen Sie die Erklärung zum `-l`-Flag. Drücken Sie die Taste `q`, um die Hilfe-Seite wieder auszublenden.

h. Rufen Sie die Hilfe-Seite zum Befehl `cat` auf und finden Sie heraus, was dieser Befehl macht. Testen Sie mit `cat test.txt`.

i. Geben Sie folgendes ein: `ls -l >> test.txt`. Anschließend rufen Sie erneut `cat test.txt` auf. Was ist passiert? Worin besteht der Unterschied zwischen `>` und `>>`? (Betrachten Sie dazu vor allem den Anfang der Datei `test.txt`.)

j. Wie Sie noch im Detail lernen werden, werden in Ausführung befindliche Programme und Hilfsprogramme durch Prozesse abgebildet, die vom Betriebssystem verwaltet werden. In der Shell können Sie sich die aktuell laufenden Prozesse zum Beispiel so anzeigen lassen: `ps -e -f -u`. Finden Sie heraus, wofür die Flags `-e`, `-f` und `-u` stehen (`man ps`).

k. Verwenden Sie den Befehl `wc` und einen geeigneten Parameter, um die Zeichen in der Datei `test.txt` zu zählen.

l. Ein sehr wichtiges Konzept (nicht nur von Linux) sind die Umgebungsvariablen. Dabei handelt es sich um Behälter für Daten, die das Betriebssystem gelegentlich benötigt. Sie können sich jede Umgebungsvariable als eine Art Spickzettel des Betriebssystems vorstellen. Geben Sie `env` ein, um sich alle Umgebungsvariablen und deren aktuelle Werte anzeigen zu lassen. Die Darstellung hat das Format `VARIABLENNAME=Wert`.

m. Lassen Sie sich nur den Wert der Umgebungsvariablen `SHELL` anzeigen, indem Sie `echo $SHELL` eingeben. Sie sehen jetzt zum Beispiel `bin/zsh` (CIP-Pool) oder `bin/bash`, was einfach dem Namen der Shell entspricht, die Sie gerade nutzen. (Ähnlich wie es verschiedene Internet-Browser wie Netscape, Mozilla oder Firefox gibt, gibt es auch verschiedene Shells.)

Remote-Login von zuhause aus: Sie können auch von zuhause aus auf die Rechner im CIP-Pool zugreifen, ohne Linux dafür bei sich installieren zu müssen.

– Es gibt (unter Windows) die Möglichkeiten für einen Remote-Login an einem CIP-Rechner, indem Sie einen SSH-Client (z.B. PuTTY) herunterladen und installieren.

– Gehen Sie auf `http://www.rz.ifi.lmu.de/FAQ/Aussenzugriff.faq.html`. Hier finden Sie alle Informationen und Schritt-für-Schritt-Anleitungen für den Remote-Login am CIP-Pool.

Aufgabe 2: (T) Systemaufrufe

(– Pkt.)

In dieser Aufgabe sollen Sie die Bedeutung von Systemaufrufen kennen lernen.

a. Erklären Sie kurz was man unter einem Systemaufruf versteht, wozu diese dienen und wie sie heutzutage implementiert werden.

b. Wozu dient das Unix Kommando `strace`?

c. Verwenden Sie das Kommando `strace` mit der Option `-c` um die folgenden Kommandos zu analysieren

- `ls`
- `ls -l`
- `ls -la`

Vergleichen Sie die Ausgabe von `strace` bei Anwendung auf die drei Kommandos. Was fällt Ihnen in Bezug auf die Anzahl und Art der autretenden Systemaufrufe auf? Erklären Sie diesen Sachverhalt kurz.

Aufgabe 3: (T) Grundlagen Shell-Programmierung

(– Pkt.)

Alle Linux-Shells stellen Konstrukte zur Verfügung, um Shell-Skripte zu erstellen. Einfache Shell-Skripte sind nichts anderes als Sequenzen von Befehlen. Zusätzlich stehen Konstrukte für typische imperative Konzepte wie Wertzuweisungen, Schleifen oder Verzweigungen (`if`) zur Verfügung.
Eine gute Einführung in das Konzept der Linux-Shell finden Sie z.B. in der Leseprobe zum Buch *Linux für Studenten* unter dem Link `http://www.pearson-studium.de/media_remote/katalog/bsp/9783827372055bsp.pdf` [1].

Wenn Sie ein beliebiges Skript geschrieben haben, muss es zunächst als ausführbar markiert werden. Das funktioniert mit dem Befehl `chmod +x filename` (`filename` ist der Dateiname des Skripts).

Durch Eingabe von `./filename` wird das Skript gestartet.
Die folgende Tabelle beinhaltet einen kleinen Überblick über typische Linux Konsolen Befehle:

Befehl	Beschreibung
`echo "mein text"`	gibt "mein text" auf der Konsole aus
`ls`	listet alle Dateien im aktuellen Verzeichnis auf
`cp quelle ziel`	kopieren
`mv alt neu`	umbennenen bzw. verschieben
`rm datei`	löschen
`grep 'text' datei`	sucht in einer Datei nach der Zeichenkette "text"
`cat datei`	gibt den Inhalt einer Datei auf dem Bildschirm aus
`head datei`	gibt einige Zeilen vom Dateianfang auf dem Bildschirm aus
`tail datei`	gibt einige Zeilen vom Dateiende auf dem Bildschirm aus
`file datei`	gibt aus, von welchem Dateityp eine Datei ist
`sort datei`	sortiert die Zeilen einer Datei alphabetisch
`wc -l datei`	zählt die Zeilen in einer Datei
`wc -w datei`	zählt die Wörter in einer Datei
`wc -m datei`	zählt die Anzahl der Buchstaben in einer Datei
`read var`	fordert den Benutzer zu einer Eingabe auf

a. Schreiben Sie ein Skript, das den Text "Hello World" zunächst in einer Variablen ablegt und dann auf der Konsole ausgibt.

b. Gegeben ist folgendes Shell-Skript:

```
1 #!/bin/sh
2 x=Raum
3 echo "$xschiff"
```

[1] Linux für Studenten, Michael Kofler/Jürgen Plate, ISBN: 978-3-8273-7205-5

Welches Problem besteht? Wie sieht die Lösung aus?

c. Schreiben Sie (mit Hilfe von Pipes) ein Shell-Skript, das ausgibt, in wie vielen Zeilen der Datei `file.txt` das Wort "Text" vorkommt.

d. Schreiben Sie ein Shell-Skript `smart`, das erkennt, ob eine beliebige Datei eine ZIP-komprimierte Datei (Dateityp `.zip`) oder eine Text-Datei (Dateityp `.txt`) ist und diese Datei automatisch entweder korrekt entpackt oder (im Falle einer Text-Datei) im XEmacs-Editor öffnet.

e. Schreiben Sie ein Shell-Skript `select`, das den Benutzer nach dem Aufruf fragt, ob seine Datei im XEmacs- oder im KWrite-Editor geöffnet werden soll.

Aufgabe 4: (H) Realisierung von Unterprogrammen

(8 Pkt.)

a. Nennen Sie einige Nachteile, die sich ergeben, wenn ein Programmierer ausschließlich offene Unterprogramme verwendet.

b. Offensichtlich ist es sehr ineffizient, auf geschlossene Unterprogramme (Prozeduren) ganz zu verzichten. Bei besonders kleinen Unterprogrammen ist es aber wiederum ungünstig, diese als geschlossene Unterprogramme zu implementieren. Warum?

c. Welche grundsätzlichen Arten der Parameterübergabe gibt es?

d. Wie werden Sprünge innerhalb eines Programms technisch realisiert?

e. Was ist der wesentliche Unterschied zwischen Sprüngen, die mit den Befehlen `JMP` und `CALL` eingeleitet werden?

f. Welche zwei wesentlichen Möglichkeiten gibt es, den `RET`-Befehl zu implementieren und woraus ergeben sie sich?

Aufgabe 5: (H) Einfachauswahlaufgabe: Das Betriebssystem

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n"). Nennen Sie dazu in Ihrer Abgabe explizit die jeweils ausgewählte Antwortnummer ((i), (ii), (iii) oder (iv)). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

a) Welche logische Hierarchieebene liegt direkt unter der Ebene des Betriebssystems?			
(i) Physikalische Geräte	(ii) Mikroprogrammierung	(iii) Anwendungsprogramme	(iv) Maschinensprache
b) Was ist keine Aufgabe des Betriebssystems?			
(i) Textverarbeitung	(ii) Komplexitätsreduktion	(iii) Ressourcenverwaltung	(iv) Zugriffskontrolle
c) Welches der folgenden Programme gehört zum Betriebssystem in dem Sinne, dass es im Kernel-Mode ausgeführt wird?			
(i) Web-Browser	(ii) Devicetreiber	(iii) Compiler	(iv) Office-Anwendung
d) Was ist kein Betriebssystemtyp?			
(i) Mainframe-Betriebssysteme	(ii) PC-Betriebssysteme	(iii) Echtzeit-Betriebssysteme	(iv) Analog-Betriebssysteme
e) Wie heißt ein Unterprogramm, das sich selbst aufruft?			
(i) reflexiv	(ii) regressiv	(iii) rekursiv	(iv) relativ