



Betriebssysteme

Projekt 2

Dokumentation

Universität Rostock
Fakultät für Informatik und Elektrotechnik
Institut für Informatik

Dozentin
Prof. Dr. Peter Luksch

Gruppe:
Christian Baganz (Matr. 1200810)
Anne Bendix (Matr. 210207811)
Kay-Heinrich Jadyschke (Matr. 212024168)

Rostock, 11. Juli 2012

Inhaltsverzeichnis

1	Aufgabenstellung	1
2	Theorie	1
2.1	Kommandozeile	1
2.2	Shell	1
2.3	Pipes	2
2.4	Prozessverwaltung	2
3	Nutzerhandbuch	3

1 Aufgabenstellung

Für das Praktikum Betriebssysteme soll im zweiten Projekt eine Konsole für die Kommunikation verschiedener Prozesse unter Linux implementiert werden. Zudem soll optional eine Prozessverwaltung oder Interprozesskommunikation umgesetzt werden. Wir haben uns für die zweite Option entschieden und werden Pipes verwenden.

In diesem Rahmen sollen verschiedene Aufgaben umgesetzt werden. Es soll eine Eingabeaufforderung geben, die bearbeitet wird und eine Befehlshistory speichert. Bei der Bearbeitung sollen Befehle mit beliebigen Argumenten im Vordergrund ausgeführt werden, wobei sequenzielle Befehle durch ein Semikolon getrennt werden. Mit `ℳ` wird die Befehlseingabe beendet, mit `Strg+C` terminieren die Vordergrundprozesse. Signale sollen abgefangen und entsprechende Nachrichten daraufhin auf der Konsole ausgegeben werden. Zudem soll auch auf entsprechende Fehlermeldungen geachtet werden. Neben einigen weiteren Formatierungsanforderungen und Variablen ist weiterhin vorgegeben, dass die Konsole mit `exit` beendet wird und mit `cd Pfad` das Arbeitsverzeichnis geändert werden kann.

2 Theorie

In diesem Absatz werden einige theoretische Grundlagen zu diesem Projekt erklärt.

2.1 Kommandozeile

Eine Kommandozeile ist eine Benutzerschnittstelle, über welches eine Software gesteuert werden kann. Sie wird oft auch als Konsole oder Terminal bezeichnet. Der Nutzer gibt über die Tastatur Befehle ein, welche dann von einer Shell oder einem Kommandozeileninterpreter gesteuert und verarbeitet werden. Welche der beiden Möglichkeiten verwendet wird ist abhängig von Betriebssystem. Ein Befehl besteht meist aus einer Zeichenkette und ist mit Parametern versehen, welche für die Ausführung der Befehle wichtig sind. Meist wird zudem ein eingegebener Befehl erst vollständig ausgeführt, bevor eine erneute Befehleingabe möglich ist. Werden automatisch gleich mehrere Befehle bei einer Eingabe ausgeführt, so nennt man dies Stapelverarbeitung.

2.2 Shell

Eine Shell ist eine Benutzerschnittstelle auf einem Computer. Sie ermöglicht somit eine Kommunikation zwischen dem Nutzer und dem Computer und bietet beispielsweise Kerneldienste an. Es gibt verschiedene Arten von Shells. So existiert neben der Kommandozeile auch noch die grafische Benutzeroberfläche.

2.3 Pipes

Als Pipe bezeichnet man einen gepufferten uni- oder bidirektionalen Datenstrom zwischen zwei Prozessen. Die Pipe arbeitet nach dem FIFO¹ Prinzip, wodurch Ausgaben eines Prozesses zu Eingaben eines anderen werden.

Die Implementierung von Pipes kann über einen Filedescriptor realisiert werden. Hierfür gibt es bereits standardmäßig die Eingabe *stdin* und die Ausgabe *stdout*. Unter Unix besteht ein solcher Filedescriptor aus einem Array mit zwei Feldern, jeweils einem für Ein- und Ausgabe. Indem die zusammengehörigen Ein- und Ausgabefelder aufeinander gelegt werden wird die Pipe zusammengesetzt.

2.4 Prozessverwaltung

Prozesse sind ein einzelnes ausführbares Programm. Mehrere Prozesse zusammen bilden einen Befehl. Wird ein Befehl eingegeben, so können die Prozesse dazu im Vordergrund oder Hintergrund ausgeführt werden. Vordergrundprozesse sind mit der Konsole gekoppelt. Wartet der Nutzer auf die Ausgabe zu einem eingegebenen Befehl, so handelt es sich meist um einen Vordergrundprozess. Hintergrundprozesse dagegen werden in gespiegelten Kindprozessen vom Betriebssystem separat ausgeführt und behindern somit nicht die Arbeit des Hauptprogrammes.

¹First-In-First-Out

3 Nutzerhandbuch

Im Folgenden sollen die Funktionen der Shell beschrieben werden:

Nachdem die Shell gestartet wurde (./kac_shell) signalisiert der Eingabeprompt die Bereitschaft der Shell.

Folgende Befehle gehören zum Funktionsumfang der Shell:

- help** listet die integrierten Befehle und deren Funktion auf
- ls** listet den Inhalt des aktuellen Verzeichnisses auf
- echo** gibt die Argumente in der Konsole aus
- cd** wechselt in das angegebene Verzeichnis
- exit** beendet die Shell

Um die Hilfe des jeweiligen Befehls anzuzeigen, wird der Befehl gefolgt von “-help“ in die Kommandozeile eingegeben.

Die Ausgabe eines Befehls lässt sich mit “>“ in die angegebene Datei umleiten.