

Unix Workshop - Aufgaben 4

Martin Wichmann Jan Kretschmer

19. März 2012

Überblick

- Komplexe Aufgabe mit verschiedenen Komponenten
- Prozesse
- Eingabe- und Ausgabeströme
- Skripte

Prozesse

- Linux verwendet viele Prozesse
- Erster Prozess: init
- PID -> Process ID
- Befehle
 - ps -> Prozesse anzeigen
 - kill -> Prozess beenden

Streams

- Bash unterstützt verschiedene Streams (Datenströme)
- ```, <, <<, >, >>, |`
- Spitze Klammern -> Leitet Ein- und Ausgabe in Datei um
 - Einfache Klammern -> Erstellt immer neue Datei
 - Doppelte Klammern -> Hängt an bestehende Datei an, sonst erstellt Datei
- `|` -> Verwendet Ausgabe als Eingabe des nächsten Befehls
- ``Befehl`` -> Führt zuerst Befehl aus und verwendet Ergebnis als weitere Eingabe
- Beispiele:
 - `cat logfile.log | grep "eth0" > filteredfile.log`
 - `echo `uname -a``

Bash Skripte

- Turing-Vollständig
- Aufbau
 - Textdatei mit Endung `.sh`
 - Erste Zeile: `#!/bin/bash`
 - Ähneln anderen Skriptsprachen
 - Etwas gewöhnungsbedürftige Syntax
- Alles was mittels Shell ausgeführt werden kann, kann auch mittels Skript erledigt werden und umgekehrt
- Aufruf mit `»sh script_file.sh«`

Cron

- Wird verwendet um Regelmäßig Aktionen auszuführen
- Konfiguration wird mittels Config-Datei erstellt
- Tool crontab legt diese an
- Format der Tabelle:
 - Minuten Stunden Tage Monate Wochentage Befehl
 - `*/5 * * * * /usr/bin/message.sh`
 - Beispiel führt alle 5 Minuten das Skript message.sh aus

- Skriptsprache zur Bearbeitung von Textdateien
- Turing-Vollständig
- Oft über Kommandozeile ausgeführt:
 - `echo Hallo Welt | awk '{printf "%s %s!\n",$1,$2}'`
 - Ausgabe: »Hallo Welt!« und Zeilenumbruch
- In diesen Aufgaben wird lediglich eine Spalte gefiltert

Umgebungsvariablen

- Enthalten spezielle, wichtige Informationen für das System
- Werden durch Shell verwaltet und gesetzt
- Werden durch \$ als Systemvariable gekennzeichnet
- Beispiel
 - echo \$PATH
 - /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
 - export PATH=\$PATH:/test/bin
- Für Aufgaben relevant:
 - \$DISPLAY -> Gibt zu verwendenden Bildschirm an
 - Muss in Skript Aufruf durch cron von Hand auf :0.0 gesetzt werden, um Program anzeigen zu können

Lernsituation

Ihre Firma verwendet den VLC-Player, um Firmen-intern Musik zu streamen. Nach einem Update des Servers, scheint der Streaming-Server nach etwa einer Stunde den Geist aufzugeben. Da Sie bisher den Fehler noch nicht finden konnten, sollen Sie nun einen Quickfix einrichten, sodass der Stream regelmäßig neu startet. Ihr Kollege arbeitet indes an einer eleganteren Lösung für diese Problem.

Testen Sie ihre Lösung lokal, sodass Sie zeigen können das es funktioniert. Verändern Sie hierzu die entsprechenden Eigenschaften.

Aufgaben I

- Vorbereitung für die Aufgabe
 - Lassen Sie sich alle Prozesse eines bestimmten Benutzers anzeigen (ps)
 - Filtern Sie diese Ausgabe nach dem Begriff "vlc" (grep)
 - Starten Sie ein beliebiges Programm und beenden Sie es von der Konsole aus (kill)
 - Schreiben Sie ein kleines Skript und lassen Sie es regelmäßig ausführen
 - Erstellen Sie ein Skript, dass auf der Konsole "Hello World!" ausgibt (sh)
 - Leiten Sie die Ausgabe in eine Datei um (>>)
 - Lassen Sie das Skript regelmäßig ausführen (cron, crontab)

Aufgaben II

- Richten Sie den Quickfix für den Streaming-Server ein
 - Schreiben Sie ein Skript, dass...
 - ...die Prozesse des Benutzers "streaming" ausgibt und nach dem Begriff "vlc" filtert (ps, grep)
 - ...die Spalte PID (1. Spalte) der vorherigen Ausgabe filtert (awk)
 - ...die gefilterten Prozesse beendet (kill) und wieder einen neuen VLC-Player startet
 - Lassen Sie das Skript alle 45 Minuten automatisch ausführen (cron, crontab)