

Praktikum 5 : Arbeiten mit der Shell und Shell-Variablen

1 Vorbemerkung (Anlehnung an wikibooks.org)

1.1 Allgemeines

Die Shell ist der Befehlsinterpreter, über den der Benutzer das System interaktiv oder über ein Skript steuern kann. Sie umgibt den zu kontrollierenden Systemkern wie eine (Muschel-)Schale. Es gibt eine Reihe von Befehlen, die im Befehlsinterpreter bereits eingebaut sind. Diese grundlegenden Befehle einer Shell werden als „Built-in Commands“ bezeichnet. Neben diesen integrierten Befehlen können externe Befehle in Form von eigenständigen Programmen aufgerufen werden.

1.2 Variablen

In einer Shell gibt es viele Möglichkeiten, Variablen einzusetzen. Man kann einer Variablen Text zuweisen, im nächsten Schritt eine Zahl, mit der man rechnet oder einen Befehl, der ausgeführt wird. Eine Variable ist dabei eine Zeichenkette, die mit einem Buchstaben oder Unterstrich beginnt, auf den eine beliebig lange Folge von Buchstaben, Ziffern und Unterstrichen folgt. Dabei gibt es vorgegebene Systemvariablen, die vom Betriebssystem genutzt werden.

Beispiele für Systemvariablen:

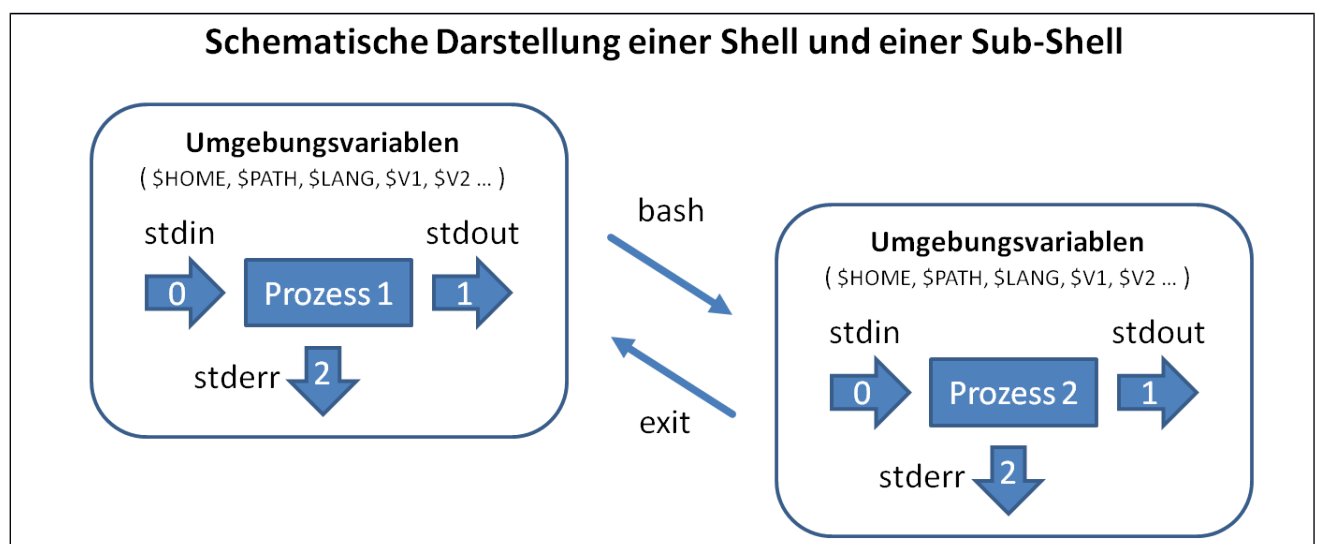
- **PATH** => Pfade, in denen ein eingegebener Befehl zur Ausführung gesucht wird
- **HOME** => das Heimatverzeichnis des aktuellen Benutzers

1.3 Regular Expressions und Meta-Zeichen

Ein regulärer Ausdruck (regular expression) ist in der Informatik eine Zeichenkette, die der Beschreibung von Mengen von Zeichenketten mit Hilfe bestimmter syntaktischer Regeln dient. Bei der Angabe von Dateinamen können eine Reihe von Meta-Zeichen verwendet werden, um mehrere Dateien gleichzeitig anzusprechen oder um nicht den vollen Dateinamen ausschreiben zu müssen. Meta-Zeichen werden auch Wildcards, Joker-Zeichen oder Platzhalter genannt.

1.4 Sonderzeichen der Shell (z.B. für Umlenkung der Ein- und Ausgabe)

Je nach verwendeter Shell stehen verschiedene Sonderzeichen zur Verfügung. Eine häufig genutzte Funktion ist dabei die Umlenkung der Ein- und Ausgabekanäle, um z.B. die normalen bzw. Fehlerausgaben eines Programms in einer Datei für die spätere Weiterverarbeitung zu speichern.



2 Übungsaufgaben

Info: Die Ausgabe der Variablenwerte erfolgt nur in Verbindung mit den angegebenen Skripts. Nutzen Sie daher für alle Aufgaben die im Tabellenkopf angegebenen Skripte.

Kommando	Aufgabe des Kommandos
pr05-watch.sh	Anzeige von Variablenwerten bzw. Inhalt von Log-Dateien

2.1 Grundsätzliche Verwendung von Variablen

Erarbeiten Sie sich an den folgenden Befehlen die grundsätzliche Verwendung von Shell-Variablen.

pr05-2.1.sh	Aufgabe des Kommandos
V1="/home"	Wertzuweisung von Variablen
echo \$V1	Ausgabe vom Variablenwert
ls \$V1	Nutzung einer Variable in Verbindung mit einem Befehl
V2=\$V1	Wertzuweisung von Variablen mit Werten von andern Variablen
V3=\$(ls \$V1)	Wertzuweisung von Variablen mit dem Ergebnis von einem Befehl
echo "\$V1 - \$V2 - \$V3"	Ausgabe mehrerer Variablenwerte als Zeichenkette

2.2 Verwendung von Variablen am Beispiel von HOME und PATH

Erklären Sie die Ergebnisse in Abhängigkeit vom aktuellen Benutzer („root“ bzw. „user“) sowie dem aktuellen Wert der jeweiligen Systemvariable.

Achten Sie darauf, in welchem Verzeichnis Sie sich nach jedem einzelnen Kommando befinden.

pr05-2.2-1.sh	Aufgabe/Ergebnis des Kommandos
echo \$HOME	
cd /	
cd	
HOME=/tmp	
cd	

Ermitteln Sie, welche Auswirkung der Wert von \$PATH auf die Ausführung der Befehle hat.

pr05-2.2-2.sh	Aufgabe/Ergebnis des Kommandos
echo \$PATH	
which ls	
ls /home	
SAVE=\$PATH	
PATH=\$HOME	
ls /home	
/bin/ls /home	
PATH=\$PATH:\$SAVE	
ls /home	

2.3 Arbeiten mit Regular Expressions und Meta-Zeichen

Erklären Sie die jeweiligen Ergebnisse in Verbindung mit den unterschiedlichen Kommandos unter besonderer Beachtung der verwendeten Zeichen „*“, „?“ und „[]“.

pr05-2.3.sh	Aufgabe/Ergebnis des Kommandos
ls /dev	Info: Verzeichnis „/dev“ enthält spezielle Geräte-Dateien (später mehr)
ls /dev/t*	
ls /dev/t*1	
ls /dev/tty?	
ls /dev/tty[13]	
ls /dev/tty[1-4]	
ls /dev/tty[!2]*	

2.4 Arbeiten mit Sonderzeichen zur Umlenkung von Ausgaben

Erklären Sie den Inhalt der Log-Dateien in Verbindung mit den unterschiedlichen Kommandos unter besonderer Beachtung der verwendeten Zeichen „>“, „>>“ und „2>“.

pr05-2.4.sh	Aufgabe/Ergebnis des Kommandos
echo "data1" 1> a.log	
echo "data2" > a.log	
echo "data3" >> a.log	
cat a.log b.log	
V3="cat a.log b.log"	
\$V3	Nutzung einer Variable zur Ausführung als Befehl
\$V3 1> c.log 2> d.log	
grep d [cd].log	

2.5 Gültigkeit von Variablen

Erklären Sie die unterschiedlichen Variablenwerte in Abhängigkeit von der aktuellen Shell.

pr05-2.5.sh	Aufgabe/Ergebnis des Kommandos
V1=wert1	
export V2=wert2	
echo "V1=\$V1 : V2=\$V2"	
bash	Neue Shell als Sub-Shell aufrufen
echo "V1=\$V1 : V2=\$V2"	
export V1=wert3	
echo "V1=\$V1 : V2=\$V2"	
exit	Sub-Shell verlassen und zur alten Shell zurückkehren
echo "V1=\$V1 : V2=\$V2"	

Name:

Studiengang:

Gruppe:

3 Praxisaufgabe

Füllen Sie diesen Aufgabenzettel in **elektronischer (!)** Form aus und geben Sie ihn ausgedruckt beim nächsten Praktikum ab. Handschriftliche Ausarbeitungen werden nicht akzeptiert.

3.1 Arbeiten mit Variablen

Ermitteln Sie die notwendigen Kommandos, um die vorgegebenen Aufgaben sowohl als Benutzer „root“ und „user“ korrekt auszuführen. Die Befehle sollen dabei unabhängig vom aktuellen Benutzer bzw. aktuellen Verzeichnis sein.

Achtung: Pro Aufgabe ist immer nur ein Kommando anzugeben.

Kommando	Aufgabe des Kommandos
	Pfad vom Heimatverzeichnis des aktuellen Nutzers als Variable V1 speichern.
	Ermittel, in welchem Pfad sich das Programm „ grep “ befindet und als Variable V2 speichern.
	Programm „ grep “ unter Verwendung der Variablen V1 und V2 in das aktuelle Heimatverzeichnis kopieren
	Kopiertes Programm „ grep “ unter Verwendung der erstellten Variablen in „ mygrep “ umbenennen

Legen Sie alle (!) benötigten Variablen fest, damit sowohl das originale Programm „**grep**“ als auch das kopierte Programm „**mygrep**“ durch den in der Tabelle vorgegebenen Aufruf ausgeführt wird.

Es sollen im Verzeichnis „**/etc**“ alle Dateien, die mit den Buchstaben „**p**“ oder „**g**“ anfangen, nach der Zeichenkette „**user**“ durchsucht werden. Die Ergebnisse sollen in der Datei „**/tmp/result.log**“ und ggf. auftretende Fehler in der Datei „**/tmp/error.log**“ gespeichert werden. Am Ende soll von allen Dateien, die im Verzeichnis „**/tmp**“ mit „**log**“ enden, der Inhalt ausgegeben werden.

Kommando	Aufgabe des Kommandos
<code>grep \$V3 \$V4 > \$V5 2> \$V6</code>	Durchsuchung mit Befehl „ grep “ ausführen.
<code>mygrep \$V3 \$V4 >> \$V5 2>> \$V6</code>	Durchsuchung mit Befehl „ mygrep “ ausführen.
	Inhalt aller Dateien ausgeben, die mit „ log “ enden

Testen Sie die Ergebnisse nach jedem Befehl, Fehlermeldungen sind zu korrigieren !!!