

Universität Potsdam
Institut für Informatik
GdP-Rechnerübung

Aufgabenblatt 4
(zuletzt aktualisiert: 11. November 2021)

6 Die Shell: Der UNIX-Kommandointerpreter

Die folgenden Aufgaben sind davon abhängig, dass Sie eine Bourne-again shell (Bash) verwenden (ihre Kommandozeile beginnt mit 'bash'). Einige Aufgaben beziehen sich auf Besonderheiten in der Umgebung im Sun-Pool. Achten Sie also darauf auf `ts-linux` angemeldet zu sein (`uname -n`).

1. Der Kommandointerpreter, den Sie benutzen, ist die sogenannte Bourne-again shell, die mit jedem Öffnen eines Terminal- oder Konsolenfensters automatisch gestartet wird. Es gibt eine ganze Reihe weiterer Programme zur Kommandointerpretation (Shells), die Sie per Kommando starten können, z.B. die C-Shell (Kommando `csh`), Bourne-Shell (`sh`), Tenex-C-Shell (`tcsh`) usw.
 - (a) Starten Sie diese Shells nacheinander als Subshells (von der ursprünglichen Shell abgeleitete Shells), indem Sie die entsprechenden Kommandos eingeben. Starten Sie dann noch einmal eine Bourne-again shell (Kommando: `bash`), und sehen Sie sich nun Ihre Prozesse mit `ps -f` an. Achten Sie auf die PIDs und die PPIDs!
 - (b) Beenden Sie nun die Shells wieder mit `exit`. Zählen Sie mit: Sie benötigen also viermal `exit`, um zu Ihrer ersten Bourne-again shell zurückzugelangen.
 - (c) Was passiert, wenn Sie ein fünftes `exit` eingeben? Das Terminal schließt sich

Hinweis: Die verschiedenen Shells unterscheiden sich durch ihren Kommandovorrat und die Wirkungsweise einzelner Kommandos und Shell-Mechanismen. Wir konzentrieren uns auf die Bourne-again shell.

2. Jede Shell läuft in einer definierten Umgebung, die für die Funktionsweise der Shell und der von ihr aufgerufenen Programme wichtig ist. Die Umgebung wird mit Hilfe von Variablen abgespeichert. Man unterscheidet **globale und lokale Variablen**.
 - (a) Mit `set` können Sie sich lokale Variablen anzeigen lassen und mit `env` die globalen. Fragen Sie die lokalen Variablen mit `set | less` und dann die globalen Variablen mit `env | less` ab. Welchen Wert haben folgenden globalen Variablen?
HOME /home/neunert
USER neunert
SHELL /bin/bash
Hinweis: Sie können die Ausgabe wie gewohnt mit `zB /HOME` durchsuchen.
 - (b) Führen Sie aus: `echo $HOME`. Was wird also bei `echo $variablenname` ausgegeben?
es wird der Wert der Variablen ausgegeben
 - (c) Finden Sie eine lokale Variable, die nicht auch als globale Variable existiert (von Groß-/Kleinschreibung abgesehen).
BASH

- (d) Legen Sie eine benutzerdefinierte lokale Variable an: `v=wert1`
 Legen Sie eine benutzerdefinierte globale Variable an: `export U=wert2`
 Überzeugen Sie sich mit `set` und `env`, dass die Variablen korrekt angelegt wurden.
 Greifen Sie auch mit dem `$`-Mechanismus auf `v` und `U` zu.
- (e) Öffnen Sie jetzt eine neue BASH-Shell als Subshell. Welche Variablen werden von der Elternshell an die Subshell weitergegeben?
 lokale Variablen (ja/nein): Nein globale Variablen (ja/nein): Ja
 (Hinweis: Die Systemvariablen, die automatisch angelegt werden, können Sie zur Beantwortung dieser Frage nicht heranziehen. Achten Sie auf benutzerdefinierte Variablen.)
- (f) Löschen Sie `U` mit `unset U`. Kann man noch auf `U` zugreifen? Nein
- (g) Beenden Sie die Subshell. Existiert `U` noch in der Elternshell? Ja
- (h) Löschen Sie `v` mit `unset v`. Kontrollieren Sie!
 (i) *Info: Die Unterscheidung von globalen und lokalen Variablen ist deshalb wichtig, weil die meisten Programme, die eine Shell aufruft, in einer Subshell ausgeführt werden. Dies gilt auch für Programme, die von bereits laufenden Programmen aufgerufen werden.*
 Legen Sie wieder eine eigene lokale und eine eigene globale Variable an. Öffnen Sie jetzt den `vi`. In dem Editor kann man UNIX-Kommandos in einer Subshell ausführen. Geben Sie z.B. im Kommandomodus von `vi` `:!echo Variablenname` ein, um `echo` auszuführen. Greifen Sie so einmal auf Ihre globale Variable und einmal auf Ihre lokale Variable zu. Was stellen Sie fest? _____
Global wird ausgegeben, lokal nicht
 Wie erklären Sie sich das? _____
nur Globale Variablen werden an ChildShells weitergegeben.

3. Beschäftigen Sie sich mit der Rolle einiger Systemvariablen!

- (a) Die `PATH`-Variable ist eine Umgebungsvariable (oder globale Variable). Sie enthält eine Menge von Verzeichnissen, in denen sich ausführbare Programme befinden. Wenn Sie also einen Befehl (wie `ls`) im Terminal eingeben, werden alle Verzeichnisse der `PATH`-Variable nach einem Programm `ls` durchsucht und das erste gefundene Programm wird dann ausgeführt.
- (b) Führen Sie den folgenden Befehl aus:
`export PATH=/bin:/usr/bin:/usr/local/bin`
- (c) Um den absoluten Pfad der Systemprogramme herauszufinden, kann man `which` benutzen. Notieren Sie die Pfade der Kommandos `ls`, `firefox`, `hwclock`:
`which ls /bin/ls` `which firefox /bin/firefox` `which hwclock` keine Ausgabe
 (Sollte eine Fehlermeldung ausgegeben werden, notieren Sie das bitte beim entsprechenden Kommando.)
- (d) Was bedeutet es, dass `which hwclock` keine Ausgabe erzeugt? _____
Das Programm "hwclock" existiert in keinem der in Path festgelegten Ordner
- (e) Ergänzen Sie den Wert von `PATH` wie folgt: `export PATH= PATH:/usr/sbin`
 Überprüfen Sie, ob sich der Wert von `PATH` wie gewünscht verändert hat.
 Erzeugt `which` jetzt die erwünschte Ausgabe mit dem Argument `hwclock`? Ja
- (f) Überprüfen Sie, ob Sie die Zusammenhänge von `which` und der `PATH`-Variablen richtig verstanden haben.
- Führen Sie erneut `unalias -a` aus!
 - Ergänzen Sie den Wert von `PATH` wie folgt: `export PATH=.: PATH`
 Welche Auswirkung hat diese Änderung? In der Shell können nun alle Programme, die im aktuellen Verzeichniss liegen ausführen

iii. Bestimmen Sie den absoluten Pfad der Datei, die ausgeführt wird, wenn Sie das Kommando `uname` aufrufen.

Kopieren Sie diese Datei in Ihr Loginverzeichnis mit dem Namen `ls`.

Was geschieht, wenn Sie jetzt in Ihrem Loginverzeichnis das Kommando `ls` aufrufen? Die Datei wird ausgeführt

Welche Ausgabe erhalten Sie, wenn Sie jetzt `which ls` ausführen? (Überprüfen Sie die Ausgaben in Ihrem Login Verzeichnis, sowie einem beliebigen anderen Verzeichnis.) ./ls

Begründen Sie! erstes Element in PATH

iv. Löschen Sie in Ihrem Loginverzeichnis nun die Datei `ls` wieder!

Kontrollieren Sie jetzt noch einmal mit `which ls`. /bin/ls

v. Geben Sie in eigenen Worten wieder, was die PATH-Variable ist und wofür man sie braucht.

Auflistung von Ordnern mit Ausführbaren Dateien

(g) Erzeugen Sie ein Unterverzeichnis `MyHome` und ändern Sie den Wert von `home` wie folgt: `HOME=~ /MyHome`

Überprüfen Sie Ihre Änderung und führen Sie dann `cd` (ohne Argument) aus. In welchem Verzeichnis sind Sie gelandet (`pwd`)? Warum?

/home/neunert/MyHome

(h) Haben Sie die globale Variable `HOME` mit verändert? Ja

(i) Schließen Sie Ihr Terminalfenster. Öffnen Sie ein neues. Wie sind jetzt die Werte von `HOME` und `PATH`? Warum? Auf Standard zurückgesetzt

Auf Standard zurückgesetzt

4. Ändern Sie Ihren Umgebungsbereich dauerhaft!

(a) *Die Konfiguration Ihrer Systemumgebung erfolgt beim Login durch Abarbeitung verschiedener Dateien mit Kommandos (Skripte), die sich u.a. im Verzeichnis `/etc` (einheitliche Einstellungen für alle Benutzer) und in Ihrem Login-Verzeichnis (individuelle Einstellungen) befinden. Für Benutzer der Bash sind `.bashrc` und `.bash_profile` die relevanten Dateien.*

Finden Sie heraus, zu welchem Zeitpunkt die Skripte ausgeführt werden.

i. Öffnen Sie dazu die Datei `~/.bash_profile` mit einem Editor (z.B. `vi`) und fügen Sie die Zeile `echo Hallo aus der BASH_PROFILE` am Ende hinzu. Fügen Sie genauso die Zeile `echo Hallo aus der BASHRC` am Ende von `~/.bashrc` hinzu.

Hinweis: Sollte in Ihrem Login-Verzeichnis keine `.bash_profile` existieren, erstellen Sie eine neue Datei.

ii. Beobachten Sie das neue Verhalten beim Einloggen sowie beim Starten einer neuen Shell. Wann werden die Dateien jeweils ausgeführt?

Hinweis: Um das Einloggen zu simulieren, führen Sie `su - username` aus. (Leerzeichen um - nicht vergessen!)

Wenn Sie per SSH verbunden sind, können Sie eine neue Shell mit `bash` starten. Wenn Sie am Pool-Rechner arbeiten, öffnen Sie einfach ein neues Terminal.

`.bashrc`: Wen eine neue Konsole geöffnet wird

`.bash_profile`: Wenn sich eine Person anmeldet

- (b) *Wenn Sie Modifikationen an der Systemkonfiguration vornehmen wollen, sollten Sie (um ein “Reset” zu ermöglichen) alle Änderungen in separaten Dateien organisieren und in `.bash_profile` nur für den Aufruf dieser neuen Dateien sorgen.*
- i. Wechseln Sie in Ihr Login-Verzeichnis!
Erzeugen Sie eine neue Datei `.profile.private`, die als einzige Zeile einen `export`-Befehl enthält. Es soll die `PATH`-Variable (mit `export`) so ergänzen, dass das aktuelle Arbeitsverzeichnis immer zuletzt nach ausführbaren Dateien durchsucht wird.
Kommando: `export PATH=$PATH:.`
(Was glauben Sie, wozu das nützlich sein könnte?)
 - ii. Fügen Sie jetzt in der Datei `.bash_profile` eine neue letzte Zeile mit dem Inhalt `source .profile.private` hinzu.
 - iii. Sie können jetzt die Datei `.profile.private` verändern, wenn Sie in Zukunft Veränderungen an Ihrer Systemkonfiguration vornehmen wollen.
 - iv. Öffnen Sie ein neues Terminalfenster und überprüfen Sie den veränderten Wert der `PATH`-Variablen. Fragen Sie bei Problemen eine Lehrkraft!
- (c) *Es könnte gewünscht sein, dass Kommandos wie `cp` oder `rm` immer mit Sicherheitsabfrage funktionieren, also wie mit Option `-i`, aber ohne diese Option immer angeben zu müssen. Dazu dient der Alias-Mechanismus der Shell.*
- i. Das Kommando `alias` (ohne Argumente) zeigt Ihnen alle Alias-Vereinbarungen an, die für Sie gesetzt sind. Es sollte bei Ihnen eine leere Ausgabe erzeugen. Sonst wiederholen Sie die Aufgabe 1 aus dem 3. Aufgabenblatt.
 - ii. Erzeugen Sie zwei leere Dateien. Führen Sie aus: `alias rm= rm -i .`
Führen Sie jetzt noch einmal `alias` aus, löschen Sie danach eine Ihrer leeren Dateien mit `rm` (ohne Option). Sie erhalten die Sicherheitsabfrage.
 - iii. Heben Sie für *einen* Befehlsaufruf den Alias-Mechanismus auf: Löschen Sie die zweite leere Datei mit `\rm` (mit vorangestelltem Backslash!). Überzeugen Sie sich, dass der Alias trotzdem noch vereinbart ist.
 - iv. Heben Sie die Alias-Vereinbarung dauerhaft auf: `unalias rm`. Überprüfen Sie!
 - v. Editieren Sie die Datei `.aliases` in Ihrem Loginverzeichnis, so dass die Namen `cp`, `rm`, `mv` dauerhaft so eingestellt sind, dass Sie ohne Angabe von `-i` mit Sicherheitsabfrage funktionieren. Sorgen Sie dafür, dass diese `alias`-Kommandos bei jedem Start einer Shell vom Typ `bash` automatisch ausgeführt wird, indem Sie der Datei `.bashrc` als neue letzte Zeile hinzufügen:
`source .aliases`
Starten Sie eine neue Bash-Shell als Subshell und überzeugen Sie sich, dass Ihre Vereinbarungen aus `.aliases` gesetzt wurden. Fragen Sie bei Problemen eine Lehrkraft!