

Manual

Kompilieren & Ausführen

sh compile.sh (Datei im Projektordner)

Ausführen

./file/shell

Shellparameter

- s Signalmodus, es werden empfangenen Signale ausgegeben
- d Debugmodus, es werden Signale und sonstige Informationen ausgegeben

! Benutze TAB für Autovervollständigung

Variablen

- Anlegen: setenv NAME WERT
- Löschen: unsetenv NAME

Verzeichniswechsel

- cd ORDNERNAME Wechselt Verzeichnis zu ORDNER
- cd .. Wechselt in übergeordneten Ordner

Programme

Alle Programme aus den Ordnern `"/bin/"`, `"/usr/bin/"`, `"/sbin/"` können ausgeführt werden. Ist ein Programm nicht in einem dieser Verzeichnisse, wird zusätzlich im aktuellen Arbeitsverzeichnis des Nutzer nach dem Programm gesucht.

- Prog & Programm wird im Hintergrund ausgeführt.
- < FILE Programme holen INPUT aus FILE.
- > FILE Programme schreiben OUTPUT in FILE.

Pipes

Programme können mit `„|“` verknüpft werden um ihre Ein/Ausgaben weiter zu leiten. Dies ist auch als Backgroundprozess (`&`) möglich.

Befehlslisten

Mehrere Shellkommandos könnten durch `„;“` getrennt werden. Diese werden chronologisch bearbeitet.

Kommentare

Kommentare bis zum Ende der Eingabe.
/x Einzelne geschützte Zeichen.
'text' Kommentare mit „'“ einleiten und mit „'“ wieder beenden.

Beenden

exit Beendet die Shell
Hinweis: Die Shell kann nur mit einem „exit“ beendet werden,
 Signale werden unterdrückt.

Benutzeroberfläche

Userinterface hängt von Fenstergröße ab.
>> ODER \$pid \$usr @ \$cwd >>

Eingabesyntax der Shell (EBNF)

```
SHELL = { EINGABE, " ; "*, EINGABE, [ "&" ] ;  
EINGABE = ( CD | ENV | EXIT | PROG | PIPE ) ;  
CD = " cd " , PFAD ;  
ENV = ( ( " setenv " , NAME , zahl ) | ( " unsetenv " , NAME ) ) ;  
PROG = name , ARGUMENTE , NEXT ;  
ARGUMENTE = null | text ;  
NEXT = PROG | null ;  
PIPE = 1 { PROG , " | " * , PROG ;
```

Terminale: cd, pfad, EXIT, text, null, zahl, ;, &, setenv, unsetenv, name;