

Systemverwaltung AIX 6.1/7

Rolf M Dietze

Jul/Aug 2010

Korn Shell

Graphiken, Tabellen und Übersichten, mit freundlicher Genehmigung des Springer-Verlages aus: OpenSolaris für Anwender Administratoren und Rechenzentren, Dietze, Heuser, Schilling, März 2006.

LABs, Copyrighted: Rolf Dietze

Einzelne Folien mit freundlicher Genehmigung des Springer-Verlages aus OpenSolaris für Anwender, Administratoren und Rechenzentern, 5/2008

ksh, Korn Shell walk through

ksh

- ▶ Entwickelt ca 1982 durch David Korn
- ▶ vi und emacs like Kommandozeileneditor
- ▶ Defaultshell bei AIX, (never change!!)
- ▶ Wildcards, Fileexpansion, History, Jobcontrol
- ▶ Regular Expressions
- ▶ Arrays, Assizioativarrays, Berechnungen, ...

ksh, Grundeinstellung

Exec Suchpfad `export PATH=/usr/bin:/usr/sbin`

Besser: `export PATH=/usr/bin:/usr/sbin:$PATH`

Manual Suchpfad `export MANPATH=/usr/man`

Konfigfile `~/.kshrc`

Manpage `man ksh`

ksh Command

Purpose

Invokes the Korn shell.

Syntax

```
ksh [ -i ] [ { + | - } { a e f h k m n p t u v x } ] [ -o Option ... ]
      [ -c String | -s | -r | File [ Parameter ] ]
```

Note: Preceding a flag with + (plus) rather than - (minus) turns off the flag.

...

ksh, Grundeinstellung

ksh Prompt PS1

```
# export PS1=$USER@$(uname -n)':$PWD$ '
root@horus:/$
```

alias Kommandoumbenennung

```
root@horus:/$ alias ls="ls -CF"
root@horus:/$ ls
admin/    lib@          smit.script
audit/    lost+found/  scriptfile*
...
```

ksh, Grundeinstellung

```
Lineediting emacs-style  set -o emacs
    ^P  Zeile nach oben
    ^N  Zeile nach unten
    ^A  Zeilenanfang
    ^E  Zeilenende
    ^B  zeichenweise rückwärts
    ^F  zeichenweise vorwärts
    ^O  Zeile löschen
```

ksh, Grundeinstellung

Lineediting `vi-style` `set -o vi`
 vi-like Navigation: hjkl....

`.ksh` Laden bei Login ENV-Eintrag in `/.profile`
 `ENV=~/.kshrc`
 `export ENV`

ksh, Grundeinstellung → .kshrc

```
set -o emacs
alias __A='^P'
alias __B='^N'
alias __D='^B'
alias __C='^F'
alias __H='^A'
alias __F='^E'
alias __G='^O'
set -o emacs

PS1=$USER@$(uname -n)':$PWD$ '
export PS1

alias ls="ls -CF"
```

ksh, Grundeinstellung → ~/.profile

Laden der ~/.kshrc beim Login:

```
ENV=~/.kshrc
export ENV
```

Ausloggen, einloggen:

```
*****
*
*
* Welcome to AIX Version 7 Beta!
*
*
* Please see the README file in /usr/lpp/bos for information pe
* this release of the AIX Operating System.
*
*
*****
root@horus:/$
```

ksh, erste Schritte

Filenamecompletion: ESC ESC

Ein `ls hESCESC` wird z.B. zu `ls home`

wildcards * alle Zeichen oder keines

`echo *` listet alle Files im aktuellen Verzeichnis aus

wildcards ? genau ein Zeichen

`ls h?ome` wird ersetzt zu `ls home`

(nicht `ls hme`, `ls holme` etc.)

Range/Wildcard `ls -ld [a-d]*`

Listet alles aus, was mit a, b, c, d beginnt und beliebig viele Zeichen hat

ksh, I/O Kanäle

Kanalnummer	Kanal
0	stdin
1	stdout
3	stderr

ksh, I/O Redirection

Umlenkung	Funktion
>	Umlenkung
>>	Anhängen
<>	File öffnen zum lesen/schreiben
2>&1	stderr öffnen und in stdout duplizieren

ksh, Argumentvektor

\$0	Erste Zeichenkette einer Kommandozeile, das Kommando selbst
\$1 .. \$9	Argument 1 bis 9
\$n	n-tes Element des Argumentvektors
\$#	Anzahl der Argumente
\$@ \$*	Alle Argumente
\$?	Retruncode des letzten Kommandos
\$\$	ProzessID der aufrufenden Shell
\$_	ProzessID des des letzten Backgroundprozesses

ksh, Conditionals auf numerische Werte

```
(( num1 == num2 ))
```

Zahlen sind gleich

```
(( num1 != num2 ))
```

Zahlen sind ungleich

```
(( num1 < num2 ))
```

num1 < num2

```
(( num1 > num2 ))
```

num1 > num2

```
(( num1 <= num2 ))
```

num1 <= num2

```
(( num1 >= num2 ))
```

num1 >= num2

ksh, Conditionals auf Zeichenketten

<code>[[str1 == str2]]</code>	str1 identisch zu str2
<code>[[str1 != str2]]</code>	str1 und str2 unterscheiden sich
<code>[[str1 < str2]]</code>	str1 liegt vor str2
<code>[[str1 > str2]]</code>	str1 liegt hinter str2
<code>[[str1 = pattern]]</code>	str1 = Pattern
<code>[[str1 != pattern]]</code>	str1 != Pattern
<code>[[-z str]]</code>	str is null
<code>[[-n str]]</code>	str is not null
<code>[x=y -o k=j]</code>	oder
<code>[x=y -a k=j]</code>	und

ksh, Kontrollstrukturen, if-then

```
if <Expression> then
    <Commands>
else
    <Commands>
fi
```

```
if <Expression> then
    <Commands>
elif <Expression>
    <Commands>
else
    <Commands>
fi
```

ksh, Kontrollstrukturen, case

```
case <word> in
    <Pattern1> <Commands>
    <Pattern2> <Commands>
    . . . .
    <PatternN> <Commands>
*)          break ;;
esac
```

ksh, Kontrollstrukturen, Schleifen

```
while <Expression> do
    <Commands>
done
```

```
for <Variable> in <list>
    <Commands>
done
```

```
until <Expression>
do
    <Commands>
done
```

ksh, Conditional Execution

cmd2 wird ausgeführt wenn cmd1 ausgeführt wurde

```
cmd1 && cmd2
```

cmd2 wird nur dann ausgeführt wenn cmd1 einen Fehlerstatus hat

```
cmd1 || cmd2
```

ksh, Dokumentation

Die Korn Shell ist die Defaultshell unter AIX.

Daher, Dokumentation des Herstellers:

<http://publib.boulder.ibm.com/infocenter/aix/v6r1/index.jsp>

<http://www.ibm.com/developerworks/aix/library/au-kornshell93.htm>

<http://www.ibm.com/developerworks/aix/library/au-kornshellscrip-ting/index.html>