

Introduction to KornShell

Soha Péter

2018. november 26.

Kivonat

Csak saját felelősségre! KSH szakértők **NE** olvassák!

1. Részletes dolgok

\$0: az éppen végrehajtott script relatív útvonalát tartalmazza az aktuális könyvtárhoz képest.

\${sh.file}: KornShell specifikus cucc, majdnem ugyanazt tudja, mint a **\$0**, csak ez abszolút path-t ad vissza

variable=\$(script): végrehajtja az argumentumában lévő parancsokat és a printoutját behelyettesíti a *variable* változóba.

pwd: **P**rint **W**orking **D**irectory. Az aktuális könyvtár *abszolút* elérési útját adja vissza (alapértelmezetten printeli).

source vagy **.** *fájlnév*: "Bemásolja" futásidőben a mögötte lévő fájl tartalmát az aktuális script "elejére". Gyakorlatilag, mint a *java import* vagy *c++ include*.

Ez az sor: "`[[feltetel]] && igaz_erteku_utasitas || masik_utasitas`"
ezt jelenti:

```
if == ;then
igaz_erteku_utasitas
else
masik_utasitas
fi
```

read: beletesz egy *printout*-ot egy változóba. Szintaxisa: `dirname $0 | read valtozonev`. Ez ekvivalens azzal, hogy: `valtozo=$(dirname $)`

2. Okosságok

- korn shebang: `#!/bin/ksh` (utána kötelező üres sor!)
- A változók nem tartalmaznak perjelet!
- printout berakása változóba: `w=$(cd $(dirname $0));pwd`
- függvény szintaxis 1: **function** name { }
- függvény szintaxis 2: **name()**{ } (nem használjuk!!!)
- if szintaxis 1: **if** [*feltétel*]; then *command* **fi**
- if szintaxis 2: `[["$valami"]]` (impliciten -n flaget ért alatta a ksh)
- -z/-n a feltételben: nulla/nemnulla a hossz
- a `[[$VALTOZO]]`: \$Valtozo hossza nem nulla (ugyanaz, mint: `[[-n $Valtozo]]`)
- \$IFS/\$OFS: input és output field separator. Default whitespace, átállítható, de nem ajánlott
- a **read**-del több változóba is be lehet tölteni cuccot!
- &&: logikai és művelet
- ||: logikai vagy művelet
- **file descriptor** 1: standard out
- **file descriptor** 2: standard error
- a **return** utasítás csak numerikus értéket tud visszaadni 0-256-ig
- **std error átirányítása az std outra**: `2 > &1`