

Svájcibicskák haladóknak

{ Bash/KSH, Perl, Python }

Mezei Tamás
tamas@componentsoft.eu

Miért írunk scripteket?

- segítik, gyorsítják a munkánkat
- magasszintű, egyszerűsített programozás...
 - ... de elég alacsonyszintű a hackeléshez
 - nem kell memóriaallokációval foglalkozni
 - gyenge típusosság
- beépített megoldások alapfeladatokra
 - szövegcsere, -formázás, leválogatás, hálózatkezelés stb.
- prototipizálás
- „csak egy sor Perlben...”

Miért nem írunk (ronda) scripteket?

- néha túl sok „szintaktikai édesítőszer”
 - nehéz lehet egymás kódján kiigazodni
 - „mindent megoldó” operátorok és funkciók
- laza változókezelés
 - típusnélküliség
 - láthatóság? (privát vagy publikus, globális változók)
- kevésbé refaktorálható
- lassúság, memória- és CPU-igény
- körülményes többszálúság
- skálázhatósági kérdések

Bash intró

- 1987 óta létező shell (Brian Fox), „Bourne again”
 - az eredeti sh parancsok superszetje
 - ötletek a Korn és C shellből
 - command-line editing, command history
 - egész számítások, default értékek
 - ciklusok, beépített változók (\$RANDOM)
 - sokan nem tudják: reguláris kifejezéseket is kezel
- ```
[["foo.txt" =~ ^[^\.]*\.txt$]] && echo "text";
```
- néha gonosz inkompatibilitások verzióváltáskor
    - 3.2-nél pont a regexkezelés...

# Bash 4 újdonosságok

- **\*\*** (globbing)
- **&>>** (átirányítás, append + stderr)
- **ulimit -b** (socket buffer méret), **-T** (threads)
- asszociatív tömbök (hash): **typeset -A** arr
- **coproc** (koprocesszek)

```
$ coproc printer { awk '{ print; fflush(); }'; }
$ echo "hello" > ${printer[1]}
$ cat ${printer[0]}
```

# Korn shell intró

- 198x óta létező, Bourne-kompatibilis, AT&T
- kiterjesztett utasításkészlet
- POSIX-kompatibilis működés
- sokáig kereskedelmi licencű, most már FOSS
- kevés, de néha annál váratlanabb eltérés Bash-hez képest (typeset különbségek, print, cmdline stb.)
  - persze mindenkinek az az otthonos, amit megszokott
- koprocesszek szinte a kezdetek (ksh88) óta | &

# KSH93 újdonságok

- asszociatív tömbök (-15 év!)
- lebegőpontos számítások
- getterek és setterek

```
function varname.set
 { let .sh.value = "{.sh.value} +10" }
function varname.get { .sh.value = `date` }
```

- összetett változók

```
$ rectangle=(a=10 b=20)
$ print "${rectangle.a}" "${rectangle[1]}"
10 20
```

# Perl 5

- interpretált scriptnyelv, valahol a shell- és az alacsonyszintű rendszerprogramozás között
- eredetileg szövegfeldolgozásra készült
- igazi svájcbicska, kevés olyan feladat van, amire ne lenne már kész Perl modul és/vagy egyszerűen csak egy operátor
- zseniális reguláris kifejezések
- nagy közösség, széles felhasználói tábor, sok OS
- „bármit írok le, fut Perlben”
- „there is more than one way to do it”

# Perl 5.10 újdonosságok

- `defined $a ? $a : $b → $a // $b`
- `given ($a) {  
    when (/^apples/) { ... }  
    when (/^oranges/) { ... }  
    default { ... }  
}`
- `/(<(?:(>[<>]+)|(?1))*>)/`
- `use feature 'say'; say("hello world");`
- `my $_; $* $#`
- ~~`use strict 'refs'; if (defined ($$x)) { ... }`~~

# Perl 6

- teljes újraírás: új VM (Parrot), opcionális típusosság, metódusparaméterek, kivételkezelés, lusta listák
- junctionök
  - `$evens = 2 & 4 & 6 & 8; $color = "red" | "blue";`
  - `sub buy(Item & Active active_item) { ... }`
- egyszerűsödő szintaxis
  - `@array = 1, 3, 5, 7; for @array { ... };`
  - `if 20 < $temp_in_celsius < 25 { say 'warm'; }`
- ... de mikor készül el?

# Python intró

- 1990-es évek elejétől, Guido van Rossum (Google)
- az olvashatóság a fő szempont
  - megköveteli a szép, formázott kódot: a behúzások jelzik a kódblokkokat, nem zárójelek
  - Perlen edzetteknek szokni kell
- többféle programozási mód: objektumorientált, funkcionális, procedurális
- az OO természete miatt áttekinthetőbb kód, a funkcionális programozási elemek miatt bonyolult dolgok egyszerűsíthetők

# Python intró 2

- többszörös öröklődés

```
class Alma(Gyumolcs, Vasarolható)
```

- listák, enneselek (tuple), szótárok (dict, map)

```
alist = range(1, 11); blist = [[1, 2], [11.65, 12]];
atup = 1, 'alma'; btup = atup, ('korte', 3);
adict = {'alma': 'piros', 'korte': 'sarga'};
bdict = dict(('alma', 'piros'), ('korte', 'sarga'));
```

- hibakezelés

```
try/except/finally/raise
```

**DEMÓ**

# Köszönöm a figyelmet!

## Svájcibicskák haladóknak

{ Bash/KSH, Perl, Python }

Mezei Tamás

tamas@componentsoft.eu