Számítógépes alapismeretek

7. előadás

zoltan.illes@elte.hu

Illés Zoltán ELTE IK

Ami eddig volt…

- Számítógépek architektúrája
- Alapvető alkotóelemek
- Hardver elemek
- Szoftver
  - Gépi kódtól az operációs rendszerig
  - Unix alapok

zoltan.illes@elte.hu

Ami ma következik…

- Shell script programozás I.
  - Változók
  - Paraméterek
  - Eredmény
  - Input-Output
  - Elágazások
  - Ciklusok

zoltan.illes@elte.hu

Script változók

- Hasonló a környezeti változókhoz
- Változók szövegesek
  - Aritmetikai műveletek közvetlenül nem használhatók.
  - x=alma; y=$x+fa; echo $y # alma+fa
  - a=5; b=$a+1; echo $b #5+1
- Közvetlenül se szöveges, se aritmetikai műveleteket nem támogat!

zoltan.illes@elte.hu

Aritmetikai utasítások

- let utasítás, a bash része, régi sh-ban nincs
  - a=2; let b=a+1; echo $b # eredmény 3
- expr utasítás
  - expr $a op $b PL: expr 3 * 4 !!!
  - op: <,<=,>,>=,!=,=,+,-,*,/,% (mod)
- bc parancs (szűrő)
  - C nyelv jellegű bemeneti szöveget vár
    - Létezik ciklus, szögfüggvények, fv definíció,stb
  - Példa: echo 2*16 | bc #32

zoltan.illes@elte.hu

BC példa

- Egy összetettebb példa: cat bcexam1|bc -l
  - X jogot kapott, így is indítható: bcexam1

```
#!/usr/bin/bc -l
#a –l kapcsoló a math könyvtárat használja (s-hez)
define fakt (x) {
if (x <= 1) return (1);
return (fakt(x-1) * x);
}
print "Az 5 faktorialis erteke: ";
print fakt(5);
print " !\n";
print "A 25 négyzetgyöke: ";
print sqrt(25);
print "\n";
print "Színusz PI/2 értéke:";
print s(3.1415/2);
print "\n";
quit
```

zoltan.illes@elte.hu

Parancs paraméterek I.

- Példa: legjobb csapat a Fradi! #ez parancs ☺
  - Mi a parancsnév?
- $1, $2, $3, … # paraméter változók
  - echo $1 # csapat
- $0 # parancs neve (legjobb)
- $# # paraméterek száma
- $* # összes paraméter, egyben
- „$@” #külön a paraméterek, példa: param

zoltan.illes@elte.hu

A param példa

- Hívása: param fű fa ‘szál fa’

```
echo paraméter használat
echo "adjunk meg összetett paramétert is ('alma fa')„
#minden külön
echo '$* használat'
for i in $*
do
echo $i
done
#minden egyben
echo '"$*" használat for ciklusban'
for i in "$*"
do
echo $i
done
#" között egyben
echo '"$@" használat'
# az alábbi sor a for i in "$@" alak rövid változata
for i
do
echo $i
done
```

zoltan.illes@elte.hu

Parancs paraméterek II.

- ${10}
  - Tizedik paraméter, zárójelek nélkül $10, a $1-hez fűzné a 0 karaktert!
- Shift utasítás
  - A paramétereket egyel balra lépteti.
    - Ez $1-et kidobja, majd balra lépnek egyet a paraméterek: $10->$9 ($10 –be $11 kerül, ha van)
  - Jellemző feldolgozás ciklus segítségével.

zoltan.illes@elte.hu

Program Eredmény

- Minden utasításnak, parancsnak van eredménye.
  - Eredmény értéke: $? környezeti változóban
- exit eredmény # eredmény:0-255,
- Ha az eredmény 0 az logikai igaz-ként is érthető. Többi érték hamis. (C nyelv fordítottja!)

zoltan.illes@elte.hu

Input-Output

- Legtöbb parancs szabvány kimenetre ír
- echo parancs
- Bemenetről olvas:
  - read parancs
    - Példa: read alma #alma változóba olvas enterig
    - read korte; echo $korte # ok
    - echo szia|read a; echo $a # üres, a read nem szűrő

zoltan.illes@elte.hu

Egy shell példa...

- Milyen formátumot várunk alma változóba?

```
#
read alma
login=`echo $alma|cut -f1 -d\&|cut -f2 -d=`
pw=`echo $alma|cut -f2 -d\&|cut -f2 -d=`
#
cat <<ali
Content-Type: text/html

<html>
<body bgcolor="#a1c1a1">
ali
 echo Azonosítója: $login , jelszava: $pw !
cat <<pali
</body> </html>
pali
```

zoltan.illes@elte.hu

Elágazások, test utasítás

- test, vagy […] logikai vizsgálat
  - 0 – igaz, 1- nem igaz, echo $?
  - -lt,-gt.-le,-ge,-eq,-ne numerikus vizsgálat
    - [$x –lt 5]
  - =, != sztring vizsgálat
    - -z string igaz, ha string hossza 0
    - -n string igaz, ha string hossza nem 0
  - -f file, -d dir file vagy könyvtár létezés
  - -o, vagy, -a az és operátor, ! a tagadás
  - [ ] esetén fontos a helyköz [után és] előtt

zoltan.illes@elte.hu

Test példák

- Teljes kapcsoló listához: man test

```
$ x=4
$ [ $x –lt 6 ]          # test $x –lt 6
$ echo $?
0 (igaz)
$test –z „alma”
$echo $?
1
$ y=fradi
$ [ $x -eq 4 –a $y != „vasas” ]
$ echo $?
0
$ test –f fradi         # igaz, ha fradi fájl létezik
$ [ ! $x –eq 4 ]
$ echo $?
1 (hamis)
```

zoltan.illes@elte.hu

Shell script elágazás

- if
  utasítások
  then
  utasítások
  else
  utasítások
  fi

```
if
        [ $x –lt 10 ]
then
        echo Kisebb mint 10
else
        echo Nagyobb
fi
```

zoltan.illes@elte.hu

Feltételes parancsvégrehajtás

- if sikeres then másikparancs fi
  - sikeres && másikparancs
- if nemsikeres then másikparancs fi
  - nemsikeres || másikparancs
  - Példa:

```
$ echo szia && echo kata
szia
kata
$ hamis || echo almafa # hamis: exit 1
almafa
```

zoltan.illes@elte.hu

Shell többirányú elágazás

- case $alma in
  idared) echo az alma idared
           ;;
  golden) echo az alma golden
           ;;
  *)   echo ismeretlen alma
           ;;
  esac

zoltan.illes@elte.hu

Elágazás példa I.

- Feladat: Olvasson be egy számot, és írja ki, hogy pozitív, vagy negatív!

```
#!/bin/sh
# beolvasás, ha az első sor megjegyzés, akkor az a shell megadása
# lehet csak!        #!/bin/sh
read x
if
    [ $x –lt 0 ]
then
    echo Az X változó negatív, értéke: $x
else
        if
            [ $x –eq 0 ]
        then
            echo A változó értéke nulla!
        else
            echo Az X változó pozitív, értéke: $x
        fi
fi
```

zoltan.illes@elte.hu

Elágazás példák II.

- Tetszőleges parancs is lehet elágazás tesztelő

```
if
 who |grep jani >/dev/null
then
 echo a jani be van jelentkezve
else
 echo a jani nincs bejelentkezve
fi
#
read x
case $x in
  [dD]*)   date       ;;
  [wW]*)  who         ;;
  l*|L*)    ls –l       ;;
  *)          echo rossz választás  ;;
esac
```

zoltan.illes@elte.hu

Shell ciklus - for

- **for** változó **in** adatlista
  - **– do**
    - utasítások
  - **– done**
- for $i in `who`
  do
  echo $i
  done
- for i in alma körte barack
  do
  echo $i
  done

zoltan.illes@elte.hu

Shell ciklus - while

- while utasítások # while példa
  do
    utasítások
  done
- while
  echo –n Írd be a neved:
  read nev
  do
  echo A Te neved: $nev
  done

zoltan.illes@elte.hu

Ciklus példa I.

- Ha a paraméter fájlnév, akkor kilistázzuk annak tartalmát

```
#!/bin/sh
#
while
 [ $# != 0 ]
do
 if test -f $1
 then
  echo $1 tartalma:
  cat $1
 else
  echo $1 nem fájl!
 fi
# paraméterek léptetése balra
 shift
 echo Még $# paraméter maradt!
done
```

zoltan.illes@elte.hu

Shell ciklus - until

- until

  utasítások

  do

  utasítás(ok)

  done

- Ha az utolsó hamis, akkor végrehajtja a ciklusmagot.

zoltan.illes@elte.hu

Ciklus példa II.

- Until példa:

```
#!/bin/sh
# kezdőérték megadása
v=igen
until
 [ $v = "nem" ]
 do
  echo -n Add meg a neved:
  read nev
  echo $nev >>nevek.txt
  echo Folytassuk? \(igen/nem\)
# Fontos a \(
  read v
 done
```

zoltan.illes@elte.hu

Ugró utasítások

- break
  - Hatására az adott ciklus félbeszakad, és a done után folytatódik a végrehajtás
- continue
  - Hatására a ciklusmag hátralévő részén átugrik, majd folytatódik a ciklus végrehajtás.
- exit [n]
  - Kilép a programból és n lesz a visszatérési érték

zoltan.illes@elte.hu

Shell script függvény definiálás

- C stílusú, paramétereit a shell scripthez hasonlóan dolgozza fel.
- Példa:

```
illes@panda:~$ cat fvpelda
#!/bin/sh
#

szia()
{
 echo Szia $1!
}
#
szia Zoli
illes@panda:~$ fvpelda
Szia Zoli!
```

zoltan.illes@elte.hu

Parancs futtatás opciók

- Sh –v fvpelda
  - -v kiírja a parancsot végrehajts előtt. A teljes fv-t is!
- Sh –x szamolvas
  - A végrehatási szál utasításait írja ki. A –v a teljes elágazást kiírja, míg a –x a tesztelt logikai kifejezéseket írja csak.
- Sh –n szamolvas
  - Szintaktikus ellenőrzés

zoltan.illes@elte.hu

Script példa I.

- Mit csinál a következő példa?

```
for i
do
 case $i
 in
 [A-Z]*) F="$F $i";;
 [a-z]*) f="$f $i";;
 [0-9]*) break;;
 esac
done
echo $F
echo $f
```

zoltan.illes@elte.hu

Script példa II: Csomagoljunk

- Példa:

```
#!/bin/sh
#
echo '# csomagoló program ‘
echo ‘# A szétszedés parancsa: sh ./fájlnév
# Egyenként vesszük a paramétereket
for i
do
  echo "echo $i 1>&2"
  echo "cat >$i <<'$i vege'"
  cat $i
  echo "$i vege"
done
```

```
$ csomagol forpelda valogat
$ ….
$ csomagol forpelda valogat >csomag
$ sh ./csomag  # kicsomagolás
```

zoltan.illes@elte.hu

Köszönöm a figyelmet!

zoltan.illes@elte.hu