05

BASH script programozás II.
Vezérlési szerkezetek

Parancsmegadás

- szokásos módon parancs megadása (mint ahogy a parancssorban megszokták)
- beágyazott parancs megadása:

  ekkor a parancs lefut és az eredménye fog behelyettesítődni (parancsként megjelenni) az adott sorban

  - `PARANCS`
  - **$(**PARANCS**)**

Matematikai műveletek

- expr KIF
  - + - * / % (csak számokon, ha nem az akkor hiba)
  - < <= > >= == != (karaktereket ASCII kód alapján)
  - !!SZÓKÖZ KELL!!, a fenti karaktereket a BASH is használja → HIBA → INKÁBB NE HASZNÁLD!
- **$((**KIF**))** ← ezt érdemes inkább!
  - csak egész számokon értelmezett, eredmény is egész (használhatóak változók)
  - +, -, *, /, %, ++, - –, **(hatványozás), !, ~, &, ^, &&, ||, ?:, =, +=, -=, *=, /=
  - < <= > >= == != (igaz: 1, hamis: 0)

Logikai műveletek

if [FELTÉTEL **]; then**

PARANCS(OK)

elif [FELTÉTEL **]; then**

PARANCS(OK)

else

PARANCS(OK)

fi

!! [ ] zárójel előtt, után KELL a SZÓKÖZ !!

!! FELTÉTEL megadása csak speciális kapcsolókkal lehetséges !! Hagyományos módon átirányítás fog történni !!

Logikai műveletek

if ((FELTÉTEL **)); then**

PARANCS(OK)

elif ((FELTÉTEL **)); then**

PARANCS(OK)

else

PARANCS(OK)

fi

Logikai műveletek

```
case $VÁLTOZÓ in
    minta1)
        PARANCS(OK);;
    minta2|minta3)
        PARANCS(OK);;
    *)
        PARANCS(OK)
esac
```

Feltételes kifejezések

- Numerikus összehasonlítás:
  - KIF1 -eq KIF2 (egyenlő)
  - KIF1 -ne KIF2 (nem egyenlő)
  - KIF1 -lt KIF2 (kisebb mint)
  - KIF1 -le KIF2 (kisebb egyenlő)
  - KIF1 -gt KIF2 (nagyobb mint)
  - KIF1 -ge KIF2 (nagyobb egyenlő)
- Logikai kifejezések:
  - KIF1 -a KIF2 (and) vagy KIF1 && KIF2
  - KIF1 -o KIF2 (or) vagy KIF1 || KIF2
  - !KIF1 (tagadás)
- Csoportosítás zárójelekkel megtehető: (KIF)

Feltételes kifejezések

- Szöveges összehasonlítás (egyetlen szó)
  - KIF1 == KIF2
  - KIF1 != KIF2
  - -z KIF (üres szó)
  - -n KIF (nem üres szó)
- Állományok jellemzőinek a vizsgálata:
  - -e KIF (létezik az állomány), -d KIF (létezik a könyvtár), -f KIF (létezik a közönséges állomány), -h (létezik a szimbolikus link), -p (létezik a csővezeték)
  - -r, -w, -x KIF olvasási, írási és végrehajtási jog az aktuális felhasználó szemszögéből
  - -O, -G KIF tulajdonos, csoport megegyezik-e az aktuális felhasználóval
  - -s KIF (nem üres)
  - KIF1 -nt KIF2 (újabb mint)
  - KIF1 -ot KIF2 (régebbi mint)
  - KIF1 -ef KIF2 (azonos a két állomány)

for

- lista bejárása:

  for VÁLTOZÓ **in** LISTA**; do**

  PARANCS(OK)

  done

- LISTA megadása:
  - szöveg felsorolva szóközzel elválasztva
  - mintaillesztéssel fájlok listája
  - $() ← beágyazott parancs kimenete (ami szóközes vagy több soros)
  - változóhivatkozás (olyan változó, ami szóközes vagy többsoros tartalommal rendelkezik)

for

- index alapú for:

```
for (( i=0; i<=$N; i++ )); do
    PARANCS(OK)
done
```

while, egyebek

while [FELTÉTEL **]; do**

PARANCS(OK)

done

- vagy másképpen

while ((FELTÉTEL **)); do**

  - PARANCS(OK)

done

- ehhez hasonlóan létezik **until** vezérlési szerkezet
- alkalmazható a **break**, **continue** és az **exit**

függvények, függvényhívás

```
function FUGGVENYNEV() {
    PARANCS(OK)
}
```

FUGGVENYNEV PARAM1 PARAM2 … PARAMN

Feladatok

- Írjunk olyan scriptet, ami a paraméterül kapott számokat összeadja és végül egy osszeg.txt fájlba írja a kapott eredményt
- Írjunk olyan scriptet, ami a paraméterül kapott számig kiírja 0-tól indulva a páros számokat.
- Nézzük meg a fájlokra vonatkozó feltételes kifejezéseket. Írjunk egy scriptet, amiben a paraméterül kapott mappában lévő fájlokat tudjuk vizsgálni. Például kiírjuk a legújabb fájlt.