Juhász Tibor:

Konzolalkalmazások gyors tesztelése

Kiegészítés a „Programozási ismeretek haladóknak” című könyvhöz
(Műszaki Kiadó, 2012)

A programozás versenyeken, de egyéb esetekben is fontos lehet, hogy minél gyorsabban tudjuk teszteli a konzolalkalmazásokat. Billentyűzetről történő adatbevitel esetén (1. korcsoport) kényelmetlen minden egyes hiba javítása után újrakezdeni az egész gépelést. Ha pedig fájlból olvasunk és fájlba írunk (2–3. korcsoport), akkor többféle tesztfájl próbájánál mindig át kéne írni az állomány nevét a forráskódban.

Az alábbiakban kétféle módszert ismertetünk, melyek segítségével automatizálhatjuk a tesztelést. Az elsőben parancsfájlt készítünk, a másodikban pedig a Windows Scripting Hostot használjuk fel a program futtatására és a fájlok átnevezésére.

A standard bemenet és kimenet átirányítása

A billentyűzetről történő adatbevitel és a képernyőre történő adatkivitel esetén a standard bemenetet és kimenetet[1] átirányíthatjuk szövegfájlokba. Ez azt jelenti, hogy a programot úgy írjuk meg, mintha a billentyűzetről olvasná be az adatokat és a képernyőre írná ki az eredményeket. Valójában azonban egy fájlból fog olvasni, és fájlba fog írni. Még egyszer megismételjük, hogy a programot úgy kell elkészíteni, mintha a billentyűzetről várná az adatokat, és a képernyőn jelenítené meg az eredményeket, tehát a programban nem fájlból olvasunk, és nem fájlba írunk! Azaz a szokásos *Console.ReadLine* és *Console.Write* vagy *Console. WriteLine* metódusokat használjuk.

A be- és kimenet átirányítása azt a célt szolgálja, hogy a tesztelésnél ne legyen szükség minden egyes futtatásnál az adatok begépelésére. Így felgyorsítjuk az ellenőrzés folyamatát.

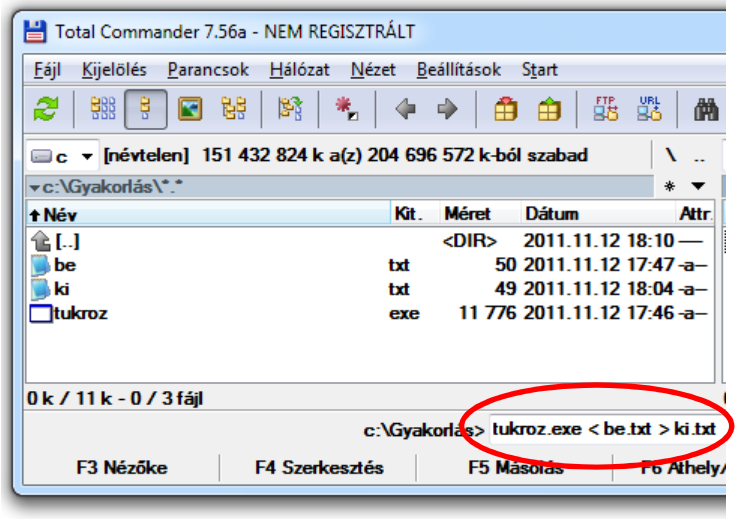
Az átirányítást a következőképpen végezzük. Először készítsük el a bemenetet tartalmazó szövegfájlt, amely a program számára a billentyűzeten begépelt adatokat tartalmazza! Ezt a fájlt mentsük el az exe-fájl mappájába! A TotalCommanderrel lépjünk be a mappába, majd a parancssorába írjuk be a következő utasítást:

```
exefájlnév < bemenet > kimenet
```

ahol az *exefájlnév* az exe-fájl neve, a *bemenet* a bemenő adatokat tartalmazó szövegfájl neve (szükség esetén a kiterjesztéssel együtt), a *kimenet* pedig az a fájl, amibe a program a képernyő helyett írni fogja az eredményeket (szintén az esetleges kiterjesztéssel együtt). Ügyeljünk a szóközökre! A későbbiekre való tekintettel a program nevében (a projekt nevében) ne alkalmazzunk ékezetes karaktereket. Például a

```
tukroz.exe < be.txt > ki.txt
```

parancs futtatja a *tukroz* programot, amelynek a billentyűzet helyett a *be.txt* fájlból adagolja a bemenetet, majd a program által a képernyőre küldött eredményeket (*Console.WriteLine*) a képernyő helyett a *ki.txt* fájlba írja.



A TotalCommander helyett természetesen a Windows parancssorát is használhatjuk (*Start/Minden program/Kellékek/Parancssor*), csak ezzel kényelmetlenebb a fájlokat tartalmazó mappa kiválasztása. ☺

[1] azaz a konzolt, alapértelmezés szerint a billentyűzetet és a képernyőt.

Parancsfájl készítése

Bár a TotalCommander parancssorának legördülő listája tartalmazza az előzőleg beírt parancsok sorát, így mégis nehézkes a program többszöri futtatása. Ezt kényelmesen egy parancsfájl segítségével végezhetjük el. A parancsfájlok szövegfájlok, de *.bat* kiterjesztéssel rendelkeznek. Indításuknál – melyet dupla kattintással is elérhetünk – a Windows végrehajtja a bennük szereplő parancsokat. **Parancsfájlok használata esetén a fájlnevekben (*.exe* fájlok, *.txt* fájlok) ne használjuk ékezetes karaktereket!**

Készítsük el a *futtat.bat* fájlt, amely egyetlen sort tartalmaz:

```
exefájlnév < bemenet > kimenet
```

Például:

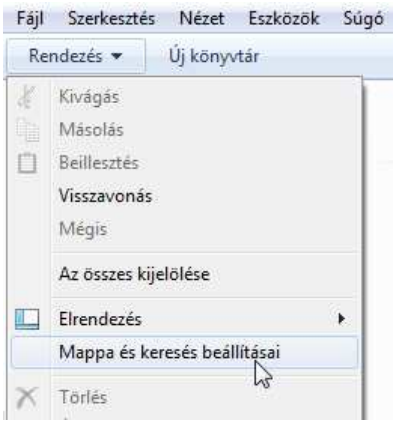
```
tukroz.exe < be.txt > ki.txt
```

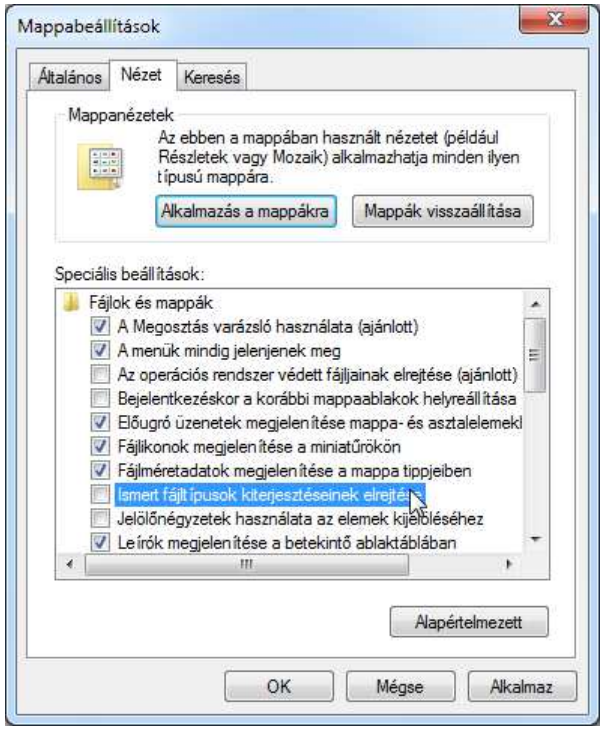
A fájlt bemásolva az exe-fájl mappájába, majd futtatva, ugyanazt az eredményt érjük el, mintha magunk gépeltük volna be az utasítást a parancssorba.

Megjegyezzük, hogy a TotalCommander az *F3* billentyűvel megtekintésre, az *F4*-gyel pedig szerkesztésre nyitja meg a szövegfájlt. **Új szövegfájlt (például parancsfájlt) a *Shift+F4* billentyűvel hozhatunk létre.** A megjelenő párbeszédablakba a kiterjesztéssel együtt írjuk be a fájl nevét, például *futtat.bat*. Így elkerülhetjük az Intézővel történő, kényelmetlenebb fájlkezelést.

A parancsfájlokba írható utasítások listáját és magyarázatát a Windows súgójában találjuk meg. Keressünk rá a súgóban a *parancssori utasítások* (Windows XP), illetve a *parancssori referencia* (Windows 7) kifejezésre!

Ha az Intézővel készítünk parancsfájlt, akkor feltétlenül jelenítsük meg a fájlok kiterjesztését (Windows 7-ben: *Rendezés/Mappa és keresés beállításai*, majd a *Nézet* panelen a *Speciális beállításoknál* kapcsoljuk ki az *Ismert fájltípusok kiterjesztésének elrejtése* jelölőnégyzetet). Ennek hiányában ugyanis hiába adjuk meg a *.bat* kiterjesztést, a Windows hozzáírja a *.txt* kiterjesztést is. Így a fájlnév például *futtat.bat.txt* lesz, ami nem indul el, ha parancsfájlként akarjuk futtatni.





Fájlok kiterjesztésének megjelenítése a Windows 7 Intézőjében

Futtatás több tesztfájllal

A fenti módszer használata esetén több tesztadatnál csak egyesével futtathatjuk programunkat. Többszöri futtatáshoz (más-más bemenő adatokkal) olyan parancsfájlt készítünk, amelyben egy ciklus segítségével adjuk meg a bemenő adatokat tartalmazó fájlokat.

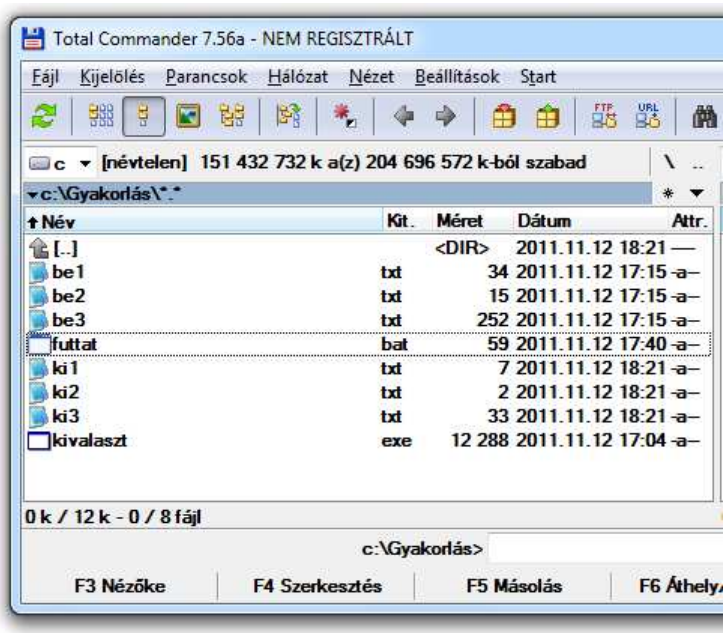
Az alábbi példában feltételezzük, hogy a bemenő fájlok neve: *be1.txt*, *be2.txt*, *be3.txt*, az eredmények pedig a *ki1.txt*, *ki2.txt*, *ki3.txt* fájlokba kerülnek. Ez az elnevezés kényelmessé teszi a parancsfájl elkészítését.

A parancsfájl egyetlen sorból áll:

```
for %%i in (1 2 3) do kivalaszt.exe < be%%i.txt > ki%%i.txt
```

ahol a *for* a ciklusutasítást jelöli, az *%%i* a ciklusváltozó, melynek értékeit az *in* után álló zárójelből veszi, a *do* után pedig az előzőleg bemutatott parancssori utasítás látható, de a fájlnevekben felhasználtuk a ciklusváltozót. A fájlt például *futtat.bat* néven menthetjük.

A futtatáshoz másoljuk az exe-fájl mappájába a bemeneteket tartalmazó szövegfájlokat és a *futtat.bat* fájlt, majd dupla kattintással indítsuk el. A parancsfájl a ciklusban szereplő értékekre futtatja a programunkat, melynek kimenetét az előzőekhez hasonlóan a megadott szövegfájlokba írja.



Ha a kimenet átirányítását `a > ki%%i.txt` helyett a `>> ki.txt` formában adjuk meg, akkor egyetlen kimeneti fájl készül, melyben egymás után sorakoznak az egyes futások eredményei (a következőt hozzáfűzi az előzőhöz). Ekkor azonban Visual Basic programunkban gondoskodjunk az utolsó adat utáni sortörésről!

A parancsfájlban szereplő ciklust a szokásos *kezdőérték*, *lépésköz*, *végérték* megadásával is felírhatjuk. Ehhez adjuk meg a */L* kapcsolót:

```
for /L %%i in (kezdőérték lépésköz végérték) do …
```

Ügyeljünk a paraméterek sorrendjére!

Az előző példában szereplő ciklussal egyenértékű megoldás:

```
for /L %%i in (1 1 3) do kivalaszt.exe < be%%i.txt > ki%%i.txt
```

Tesztelés a Windows Scripting Host segítségével

Ha a tesztelésre kerülő program fájlból olvas és fájlba ír, akkor kissé hosszadalmas a tesztadatokat tartalmazó állományok átnevezése parancsfájl segítségével. Használjuk inkább a Windows Scripting Host (WSH) által nyújtott segítséget! A tesztelő programot VBScript (Visual Basic Script) vagy JScript (Java Script) nyelven írhatjuk meg, majd szövegfájlba mentjük *.vbs* (vagy *.js*) kiterjesztéssel. Dupla kattintásra a Windows (a WSH) végrehajtja a szövegfájlban szereplő utasításokat. Itt nem térünk ki a szkriptek részletes ismertetésére, csak egy példát mutatunk a tesztelésre.

Az alábbi szkript a *Fajlnev* változóban megadott *exe* fájlt futtatja egymás után többször úgy, hogy minden egyes futtatás előtt a bemenő adatokat tartalmazó *Fajlnev.bexx* fájlt átnevezi *Fajlnev.be*-re, a futtatás után pedig az output eredményeket tartalmazó *Fajlnev.ki* fájlt átnevezi *Fajlnev.kixx*-re. Ez megfelel a programozási versenyeken megszokott elnevezéseknek.

A program az *xx* értékét a *Min* és a *Max* változókkal megadott egész számok között változtatja. A futtatások során az összes kimeneti fájlt egyesíti a *Fajlnevmind.txt* fájlban. Az alábbi forráskódban értéket adtunk ezeknek a változóknak (*osszead*, 1, 4).

A forráskódot másoljuk be egy szövegfájlba, és mentsük el *.vbs* kiterjesztéssel (például *futtat.vbs*)!

```
Min = 1
Max = 4
Fajlnev = "osszead"
' Fajlnev = InputBox("Fájlnév:") ' így is lehetne

Fajlbe = Fajlnev & ".be"
Fajlki = Fajlnev & ".ki"
Fajlexe = Fajlnev & ".exe"

Set fso = CreateObject("Scripting.FileSystemObject")
Set oShell = CreateObject("WScript.Shell")
For I=Min To Max
  If fso.FileExists(Fajlbe & I) Then
    fso.CopyFile Fajlbe & I, Fajlbe, True
    Return = oShell.Run(Fajlexe, 0, True)
    If Return=0 Then ' rendben lefutott
      fso.CopyFile Fajlki, Fajlki & I, True
    End If
  End If
Next
fso.DeleteFile Fajlbe
fso.DeleteFile Fajlki
```

Nem térünk ki a szkript részleteire, csak annyit jegyzünk meg, hogy az objektumváltozók értékadásánál kötelező kiírni a *Set* kulcsszót.

Ismételt futtatás esetén a szkript felülírja az előző kimeneti fájlokat (a *Fajlnevmind.txt*-t is). A szkript nincs felkészítve az *exe* fájl futtatásának hibáira, de hiányzó be- vagy kimeneti fájl nem okoz hibás működést. **A futtatás után a fájlkezelőben (például TotalCommander) frissíteni kell a mappa tartalomjegyzékét (*Ctrl+R* vagy *F5*)!**

Ha egyszerűsíteni szeretnénk az output állományok ellenőrzését, akkor készítsük el a helyes kimeneteket tartalmazó fájlokat *Fajlnevmo.kixx* néven! A szkriptet (ugyanabban a szövegfájlban) egészítsük ki az alábbi sorokkal:

```
Fajlmegold = Fajlnev & "mo.ki"
Set Ir = fso.OpenTextFile("Ertekel.txt", 2, True)
    ' 2: törli, ha már volt ilyen
For I=Min To Max
  If fso.FileExists(Fajlki & I) Then
    Set Olvas = fso.OpenTextFile(Fajlki & I, 1) ' 1: olvasásra nyitja meg
    Set Megold = fso.OpenTextFile(Fajlmegold & I, 1)
    Sor1 = Olvas.ReadAll
    Sor2 = Megold.ReadAll
    Olvas.Close
    Megold.Close
    If Sor1 = Sor2 Then
      Ir.WriteLine I & ". jó"
    Else
      Ir.WriteLine I & ". hibás"
    End If
  Else
    Ir.WriteLine I & ". hiányzik"
  End If
```

```
Next
Ir.Close
```

Ezek eredményeként a szkript megkeresi az *exe* fájl mappájában a helyes megoldásokat tartalmazó *Fajlnevmo.kixx* fájlokat, és összehasonlítja a program kimeneteként kapott *Fajlnev.kixx* fájlokkal. Az értékelést az *Ertekel.txt* fájlba menti.

A szkriptek részleteiről az MSDN webhelyén tájékozódhatunk:

Bevezetés:	http://msdn.microsoft.com/en-us/library/ec0wcxh3(VS.85).aspx
WSH:	http://msdn.microsoft.com/en-us/library/9bbdkx3k.aspx
Reference WSH:	http://msdn.microsoft.com/en-us/library/98591fh7(v=vs.85).aspx
Language reference:	http://msdn.microsoft.com/en-us/library/d1wf56tt(v=vs.85).aspx

A fenti módszereket felhasználhatjuk arra, hogy a programozási versenyeken vagy a felkészülésnél gyorsan tudjuk tesztelni programjainkat. A feladat szövegében szereplő bemenő adatokat írjuk szövegfájlba, majd egy parancsfájl segítségével futtassuk a programot! A feladat szövegében megadják a fájl (projekt) nevét, amely általában nem tartalmaz ékezeteket. Így ez nem okoz problémát a parancsfájl végrehajtásánál.

A parancsfájlt és a bemenő adatokat tartalmazó fájlt célszerű a *bin*/*Release* mappába tenni. A futtatás előtt ne felejtsük el lefordítani a programot (*Debug*/*Build* menüparancs)!

A fenti példákat a mellékelt *Teszteles.zip* fájlban található projektek tartalmazzák. A bemenő adatok szövegfájljait, illetve a parancsfájlokat lásd a *bin*/*Release* mappákban.