

# Parancssor

## Kötelező

1. Írd ki, hogy melyik a kurrens könyvtár (pwd).
2. Listázd ki, hogy ebben a könyvtárban milyen fájlok és mappák vannak (ls).
3. Nézd meg a fájlok és könyvtárak részletes adatait. Mikor és ki hozta ezeket létre (ls -l)? Nézd meg, milyen további kapcsolói vannak az ls parancsnak (man ls).
4. Hozz létre egy könyvtárat (mkdir).
5. Lépj be a könyvtárba (cd). Ellenőrizd le, hogy a kurrens könyvtár tényleg megváltozott-e. Lépj vissza a szülő könyvtárba (cd ..), majd újra a létrehozott könyvtárba.
6. Hozz létre egy szövegfájlt, és írd ki a tartalmát (cat). A cat parancs a concatenate szó rövidítése. Hogy lehet több fájl tartalmát egymás után konkatenálva kiírni?
7. Másold le a szövegfájlt (cp) az aktuális mappába, majd az eggyel feljebb lévő mappába. Ellenőrizd, hogy megtörtént-e a másolás, majd lépj vissza az eredeti fájl tartalmazó könyvtárba.
8. Nevezd át az aktuális mappában lévő másolatot (mv).
9. (a Hello world után) Távolítsd el (rm) a fordítással létrejövő futtatható állományt (a.out)! Lépj át a szülő könyvtárba, helyezd át ide a forrást, majd töröld ki a korábban létrehozott mappát (rm -r)!

## Optionális

1. Írd ki, hogy a fájlodban hány sor, szó és karakter található (wc). Írd ki, csak a fájl sorainak számát. Ha nem tudod, hogy ezt milyen kapcsolóval lehet megcsinálni, nézd meg a manualban.
2. Írj programot, mely kiírja a képernyőre, hogy Hello world. A program kimenetét irányítsd bele a wc nevű programba, ami kiírja a kimeneted sorainak, szavainak és karaktereinek számát.

## Haladó

1. Készítsünk pár szöveges fájlt, az egyikben rejtjük el az alma szót. A grep program segítségével keressük meg mely fájlban van elrejtve az alma.
2. Készítsünk több könyvtár mély könyvtárszerkezetet. Rejtjük el egy alma.txt nevű fájlt valahol. A könyvtárszerkezet legtetetéről találjuk meg az elrejtett fájlt a find parancs segítségével.

## Standard I/O

## Kötelező

1. Írj programot, amely kiírja a képernyőre, hogy Hello World! (Ezt variáljuk: mutassuk meg, mi a warning, mi az error, töröljünk ki annyi szóközt, amennyit lehet, nézzük meg, mi a különbség a szóközzel és a tabulátorral való behúzás között, stb.)
2. Írj programot, amely kiírja a nevedet.
3. Írj programot, mely kiírja két szám összegét.

## Optionális

1. Írj programot, melyben elosztasz két számot. Írd ki az eredményt. Mi történik, ha az osztó 0? Mi történik, ha int vagy float típusú változókat használunk?
  - Használd a %f formázó karaktert. Figyelj oda, hogy kiíráskor két tizedesig írja csak ki az eredményt.
  - Használd a %d formázó karaktert. Nézd meg, hogy mi történik.
  - A program fordításánál használd a -W, -Wall, -Wextra és -pedantic kapcsolókat. Vajon fordítási hibát okoz, ha hibás formázó karaktert használasz?
2. Írj programot, melyben kiszámolod egy négyszög, illetve egy kör kerületét és területét. A négyszög két oldalát, illetve a kör sugarát kérd be a standard inputról.
  - Használd a scanf("%d", &a) függvényt. Használd a & operátort, amely a változó címét adja vissza. A  $\pi$  értékét vedd 3.1415-nek.
3. Írj programot, ami a neveden köszönt. A nevedet tárold változóban.
  - Használd a char[] típust, és a %s formázó karaktert. Figyelj oda, tömböknél fontos a szintaxis: char name[10].

## Haladó

1. Írj programot, mely a napok számát átszámolja évekre, hetekre és napokra. Pl. 375 nap = 1 év 1 hét 3 nap. Mindig csak a nagyobb egység maradékával dolgozz.
2. Írj Celsius-Fahrenheit konvertáló programot.