

# Adatbázisrendszerek

## 1.Gyakorlat

2025.02.19.

### **Készítette:**

Orosz Kristóf Bsc

Szak: Programtervező Informatikus

Neptunkód: EYZWG9

**Sárospatak, 2025**

- 1. feladat** - C nyelven készítsen egy programot, amely szabvány billentyűzetről olvasson be sorokat, egészen a végjelig (#). A beolvasott szöveg: a hallgató teljes neve és a neptunkód. A beolvasott sorokat írja ki egy szövegfile-ba (neptunkod.txt). A szövegfile nevét a bevitel első sorában adja meg (File name). Az így létrehozott, lezárt állományt utána nyissa meg és írja vissza a lementett szöveget nagybetűs formában.

A kódrészlet négy alapvető C könyvtárat tartalmaz. A **<stdio.h>** biztosítja a standard bemenet-kimeneti műveletekhez szükséges függvényeket, míg az **<stdlib.h>** segédfüggvényeket kínál, a memóriafoglaláshoz és a programvezérléshez. A **<string.h>** és a **<ctype.h>** könyvtárak pedig segítséget nyújtanak a karakterek nagybetűkké való átváltásához.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
```

A program elején két karaktertömböt deklaráltam. Ezek a fájlnev és a sor tárolására szolgálnak. A **fgets** függvény segítségével a standard bemenetről beolvasom a fájlnevet. Esetleg, ha ez a művelet sikertelen lenne, akkor egy hibaüzenet jelenik meg, majd a program hibakóddal kilép. Végül a beolvasott fájlnévből eltávolítja a sortöréseket.

```
int main() {
    char filename[100];
    char line[256];

    if (fgets(filename, sizeof(filename), stdin) == NULL) {
        printf("Hiba a fájlnev beolvasásakor.\n");
        return 1;
    }
    filename[strcspn(filename, "\n")] = '\0';
```

A kódrészlet megnyitja a fájlt. Ha a megnyitás sikertelen, akkor a **printf** segítségével hibát jelez, majd a program kilép. A **while** ciklusban a program soronként olvassa a standard bemenetet, eltávolítja a sortöréseket, és amint egy sor megegyezik a "#" karakterrel, a ciklus megszakad.

```
FILE *fp = fopen(filename, "w");
if (fp == NULL) {
    printf("A fájl megnyitása sikertelen.");
    return 1;
}

while (fgets(line, sizeof(line), stdin) != NULL) {
    line[strcspn(line, "\n")] = '\0';
    if (strcmp(line, "#") == 0) {
        break;
    }
    printf(fp, "%s\n", line);
}
fclose(fp);
```

A kódrészlet megnyitja a fájlt olvasási módban, és ellenőrzi, hogy sikerült-e a megnyitás. Hiba esetén leállítja a programot. A program a fájl tartalmát soronként olvassa be, majd egy belső ciklussal minden karaktert nagybetűssé alakít a **toupper** függvény segítségével, végül a módosított sorokat kiírja a képernyőre.

```
fp = fopen(filename, "r");
if (fp == NULL) {
    fprintf(stderr, "A fájl megnyitása olvasásra sikertelen!\n");
    return 1;
}

while (fgets(line, sizeof(line), fp) != NULL) {
    for (int i = 0; line[i] != '\0'; i++) {
        line[i] = toupper((unsigned char)line[i]);
    }
    printf("%s", line);
}
fclose(fp);

system("pause");
return 0;
}
```

**2.feladat** - Az 1. feladathoz hasonlóan olvasson be sorokat a billentyűzetről a végjelig. (#). A beolvasott szöveg: a hallgató teljes neve és a neptunkód. Végezze el egy fájl tartalmának másolását egy másik fájlba (fp1, fp2;). A fileneveket másoló függvényt hívó részben kell beolvasni.

A kódrészlet négy alapvető C könyvtárat tartalmaz. A **<stdio.h>** biztosítja a standard bemenet-kimeneti műveletekhez szükséges függvényeket, míg az **<stdlib.h>** segédfüggvényeket kínál, a memóriafoglaláshoz és a programvezérléshez. A **<string.h>** és a **<ctype.h>** könyvtárak pedig segítséget nyújtanak a karakterek nagybetűkké való átváltásához.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
```

A kódrészlet egy fájl másoló függvényt tartalmaz. Először megnyitja a program a forrásfájlt olvasásra, majd a célfájlt írásra, ha bármelyik fájl megnyitása sikertelen, hibaüzenetet ír ki és kilép a programból. Ha sikerül megnyitni mindkét fájlt, akkor soronként átmásolja a forrásfájl tartalmát a célfájlba, majd bezárja a fájlokat.

```
void masolas(const char *forrasfajl, const char *celfajl) {
    FILE *fp1 = fopen(forrasfajl, "r");
    if(fp1 == NULL) {
        printf("Hiba: A forrasfajl megnyitasa sikertelen.\n");
        return;
    }
    FILE *fp2 = fopen(celfajl, "w");
    if(fp2 == NULL) {
        printf("Hiba: A celfajl megnyitasa sikertelen.\n");
        fclose(fp1);
        return;
    }
    char buffer[256];
    while(fgets(buffer, sizeof(buffer), fp1) != NULL) {
        fputs(buffer, fp2);
    }
    fclose(fp1);
    fclose(fp2);
}
```

A program először beolvassa a fájl nevét, eltávolítva az esetleges újsor karaktert, majd megnyitja a megadott nevű fájlt írásra. Hiba esetén hibajelzést ad és kilép. Ezután soronként olvassa be a felhasználó által megadott szövegeket, és minden beolvasott sort a fájlba ír, míg a beolvasott sor "#" karaktert nem tartalmaz. Amikor a "#" karakter szerepel a bemenetben, a ciklus megszakad, és a program bezárja a fájlt.

```
int main(void) {
    char fajlnev[100], sor[256];

    if(fgets(fajlnev, sizeof(fajlnev), stdin) == NULL) {
        printf("Hiba a fajlnev beolvasasakor.\n");
        return 1;
    }
    fajlnev[strcspn(fajlnev, "\n")] = '\0';

    FILE *fp = fopen(fajlnev, "w");
    if(fp == NULL) {
        printf("Hiba: A fájl megnyitása írásra sikertelen.\n");
        return 1;
    }

    while(fgets(sor, sizeof(sor), stdin) != NULL) {
        sor[strcspn(sor, "\n")] = '\0';
        if(strcmp(sor, "#") == 0) {
            break;
        }
        printf(fp, "%s\n", sor);
    }
    fclose(fp);
}
```

A program először beolvassa a felhasználótól a forrás- és célfájl nevét, eltávolítva az esetleges újsor karaktereket. Ezután meghívja a korábban definiált masolas() függvényt, amely a forrásfájl tartalmát másolja a célfájlba.

```
char forrasFajl[100], celFajl[100];

printf("\nAdja meg a forrasfajl nevet: ");
if(fgets(forrasFajl, sizeof(forrasFajl), stdin) == NULL) {
    printf("Hiba a forras fajl nevenek beolvasasakor.\n");
    return 1;
}
forrasFajl[strcspn(forrasFajl, "\n")] = '\0';

printf("Adja meg a cel fajl nevet: ");
if(fgets(celFajl, sizeof(celFajl), stdin) == NULL) {
    printf("Hiba a cel fajl nevenek beolvasasakor.\n");
    return 1;
}
celFajl[strcspn(celFajl, "\n")] = '\0';

masolas(forrasFajl, celFajl);
printf("A fajl masolasa kesz.\n");

system("pause");
return 0;
}
```