Fájlkezelés

A fájlkezelés célja, hogy a program adatokat tudjon elmenteni és újra beolvasni. C-ben a fájlműveletekhez a <stdio.h> könyvtárat használjuk.

Fájl megnyitása (fopen)** **

```
FILE *fp;
fp = fopen("adat.txt", "r");
```

Módok:

Mód	Jelentés
`“r”`	Olvasásra nyitás (a fájlnak léteznie kell)
`“w”`	Írásra nyitás (ha létezik, törli a tartalmat)
`“a”`	Hozzáfűzés (append)
`“r+“`	Olvasás és írás, meglévő fájl
`”w+“`	Új fájl olvasásra és írásra
`”a+“`	Hozzáfűzés + olvasás

Fájl lezárása (fclose)

```
fclose(fp);
```

Szöveges fájl olvasása/írása

Művelet	Függvény	Példa
Írás	`fprintf()`	`fprintf(fp, ”%d %s”, kor, nev);`
Olvasás	`fscanf()`	`fscanf(fp, “%d %s”, &kor, nev);`
Karakter írás	`fputc()`	`fputc('A', fp);`
Karakter olvasás	`fgetc()`	`ch = fgetc(fp);`

Bináris fájlok

Bináris fájlok esetén:

- fwrite() és fread() használatos,
- az adat nem szövegként, hanem nyers bájtokként kerül mentésre.

```
fwrite(&valtozo, sizeof(valtozo), 1, fp);
fread(&valtozo, sizeof(valtozo), 1, fp);
```

Tipikus hibák

- Elfelejtett fclose(fp);
- Rossz megnyitási mód (pl. “r” amikor a fájl nem létezik)
- Rossz formátum a fscanf()-ben

Gyakorlás

1. Írj programot, ami létrehoz egy fájlt, és beleír pár sort.

```
#include <stdio.h>

int main() {
    FILE *fp = fopen("szoveg.txt", "w");
    if (fp == NULL) {
        printf("Hiba: nem sikerult megnyitni a fajlt!\n");
        return 1;
    }

    fprintf(fp, "Ez az elso sor.\n");
    fprintf(fp, "Ez a masodik sor.\n");
    fclose(fp);
    printf("Fajl sikeresen letrehozva.\n");
    return 0;
}
```

2.Fájl olvasása és képernyőre írása

```
#include <stdio.h>

int main() {
    FILE *fp = fopen("szoveg.txt", "r");
    char sor[100];

    if (fp == NULL) {
        printf("Nem sikerult megnyitni a fajlt!\n");
        return 1;
    }

    while (fgets(sor, 100, fp) != NULL)
        printf("%s", sor);

    fclose(fp);
    return 0;
}
```

3. Karakterenkénti másolás

```
#include <stdio.h>

int main() {
    FILE *in = fopen("bemenet.txt", "r");
    FILE *out = fopen("kimenet.txt", "w");
    char ch;

    if (in == NULL || out == NULL) {
        printf("Hiba fajl megnyitasakor!\n");
        return 1;
    }

    while ((ch = fgetc(in)) != EOF)
        fputc(ch, out);

    fclose(in);
    fclose(out);
    printf("Masolas kesz.\n");
    return 0;
}
```

4. Számok beírása fájlba

```
#include <stdio.h>

int main() {
    FILE *fp = fopen("szamok.txt", "w");
    int i;
    for (i = 1; i <= 10; i++)
        fprintf(fp, "%d\n", i);
    fclose(fp);
    return 0;
}
```

5. Számok beolvasása és összegzés

```
#include <stdio.h>

int main() {
    FILE *fp = fopen("szamok.txt", "r");
    int szam, osszeg = 0;

    while (fscanf(fp, "%d", &szam) == 1)
        osszeg += szam;

    fclose(fp);
    printf("Osszeg: %d\n", osszeg);
    return 0;
```

```
}
```

6. Bináris írás és olvasás

```
#include <stdio.h>

int main() {
    int tomb[5] = {10, 20, 30, 40, 50};
    int beolvasott[5];
    int i;

    FILE *fp = fopen("adat.bin", "wb");
    fwrite(tomb, sizeof(int), 5, fp);
    fclose(fp);

    fp = fopen("adat.bin", "rb");
    fread(beolvasott, sizeof(int), 5, fp);
    fclose(fp);

    for (i = 0; i < 5; i++)
        printf("%d ", beolvasott[i]);
    return 0;
}
```

7. Struktúra fájlba mentése (szöveges)

```
#include <stdio.h>

typedef struct {
    char nev[30];
    int kor;
} Szemely;

int main() {
    Szemely ember = {"Kovacs Bela", 32};

    FILE *fp = fopen("ember.txt", "w");
    fprintf(fp, "%s %d\n", ember.nev, ember.kor);
    fclose(fp);

    printf("Ember adatai mentve.\n");
    return 0;
}
```

8. Struktúra olvasása fájlból

```

```

```
#include <stdio.h>

typedef struct {
    char nev[30];
    int kor;
} Szemely;

int main() {
    Szemely ember;
    FILE *fp = fopen("ember.txt", "r");
    fscanf(fp, "%s %d", ember.nev, &ember.kor);
    fclose(fp);
    printf("%s (%d ev)\n", ember.nev, ember.kor);
    return 0;
}
```

9. Struktúra és mátrix fájlban

```
#include <stdio.h>

#define N 2
#define M 2

typedef struct {
    char nev[30];
    int pontok[N][M];
} Jatekos;

int main() {
    Jatekos j = {"Anna", {{1,2},{3,4}}};
    FILE *fp = fopen("jatekos.txt", "w");

    fprintf(fp, "%s\n", j.nev);
    for (int i = 0; i < N; i++) {
        for (int k = 0; k < M; k++)
            fprintf(fp, "%d ", j.pontok[i][k]);
        fprintf(fp, "\n");
    }
    fclose(fp);
    printf("Jatekos adatai mentve.\n");
    return 0;
}
```

10. Egyszerű adatbázis fájlban (szöveges)

```
#include <stdio.h>

#define N 3
```

```
typedef struct {
    char nev[30];
    int kor;
} Diak;

int main() {
    Diak diakok[N] = {
        {"Anna", 18},
        {"Bela", 19},
        {"Csilla", 17}
    };

    FILE *fp = fopen("adatbazis.txt", "w");
    int i;
    for (i = 0; i < N; i++)
        fprintf(fp, "%s %d\n", diakok[i].nev, diakok[i].kor);
    fclose(fp);

    printf("Adatbazis mentve.\n");

    // visszaolvasás
    fp = fopen("adatbazis.txt", "r");
    printf("\n--- Beolvasott adatok ---\n");
    for (i = 0; i < N; i++) {
        fscanf(fp, "%s %d", diakok[i].nev, &diakok[i].kor);
        printf("%s (%d ev)\n", diakok[i].nev, diakok[i].kor);
    }
    fclose(fp);

    return 0;
}
```

11. 10 diák kap egy 0–100 közötti pontszámot. A pont alapján kategóriát kap:

1. A: 90–100
2. B: 75–89
3. C: 60–74
4. D: 40–59
5. F: 0–39

Ezeket az adatokat fájlba kell írni.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define N 10

typedef struct {
```

```
    char nev[30];
    int pont;
    char kat;
} Diak;

int main() {
    Diak diakok[N] = {
        {"Anna"}, {"Bela"}, {"Csilla"}, {"David"}, {"Erika"},
        {"Ferenc"}, {"Gabor"}, {"Hanna"}, {"Istvan"}, {"Judit"}
    };

    srand(time(NULL));

    FILE *fp = fopen("diakok.txt", "w");
    int i;
    for (i = 0; i < N; i++) {
        diakok[i].pont = rand() % 101;

        if (diakok[i].pont >= 90) diakok[i].kat = 'A';
        else if (diakok[i].pont >= 75) diakok[i].kat = 'B';
        else if (diakok[i].pont >= 60) diakok[i].kat = 'C';
        else if (diakok[i].pont >= 40) diakok[i].kat = 'D';
        else diakok[i].kat = 'F';

        fprintf(fp, "%s %d %c\n", diakok[i].nev, diakok[i].pont,
diakok[i].kat);
    }

    fclose(fp);
    printf("Adatok kiirva a diakok.txt fajlba.\n");
    return 0;
}
```

12. Tárolj 7 napi hőmérsékletet egy struktúrában (napok neve + random -10 és 35 °C között). Írd fájlba a táblázatot.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define N 7

typedef struct {
    char nap[10];
    int homerseklet;
} HetNap;

int main() {
    HetNap het[N] = {
```

```
        {"Hetfo"}, {"Kedd"}, {"Szerda"}, {"Csutortok"},
        {"Pentek"}, {"Szombat"}, {"Vasarnap"}
    };

    srand(time(NULL));
    FILE *fp = fopen("homerseklet.txt", "w");

    int i;
    for (i = 0; i < N; i++) {
        het[i].homerseklet = (rand() % 46) - 10;
        fprintf(fp, "%s %d\n", het[i].nap, het[i].homerseklet);
    }

    fclose(fp);
    printf("Homersekletek kiirva fajlba.\n");
    return 0;
}
```

13. Tárolj 3 hallgatót, mindegyik 3 vizsgát írt. Generálj véletlen pontokat (0–100) egy 3×3 mátrixba, számítsd ki az átlagukat, és írd fájlba.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define N 3
#define M 3

typedef struct {
    char nev[30];
    int pont[M];
    float atlag;
} Hallgato;

int main() {
    Hallgato hallgatok[N] = {
        {"Anna"}, {"Bela"}, {"Csilla"}
    };

    srand(time(NULL));
    FILE *fp = fopen("vizsga.txt", "w");

    int i, j;
    for (i = 0; i < N; i++) {
        int osszeg = 0;
        for (j = 0; j < M; j++) {
            hallgatok[i].pont[j] = rand() % 101;
            osszeg += hallgatok[i].pont[j];
        }
```

```
        hallgatok[i].atlag = (float)osszeg / M;
        fprintf(fp, "%s ", hallgatok[i].nev);
        for (j = 0; j < M; j++)
            fprintf(fp, "%d ", hallgatok[i].pont[j]);
        fprintf(fp, "-> %.2f\n", hallgatok[i].atlag);
    }

    fclose(fp);
    printf("Vizsgaadatok fajlba mentve.\n");
    return 0;
}
```

14. 5 oktató → 4 kurzus mindegyiknél, random pontszám 0–100. Átlag alapján kategória:

- A: 85–100
- B: 70–84
- C: 55–69
- D: 40–54
- F: < 40

Minden oktató nevét, kurzuspontjait, átlagát és kategóriáját írd a fájlba.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define N 5
#define M 4

typedef struct {
    char nev[30];
    int pontok[M];
    float atlag;
    char kat;
} Oktato;

int main() {
    Oktato oktatok[N] = {
        {"Kovacs"}, {"Toth"}, {"Nagy"}, {"Varga"}, {"Horvath"}
    };

    srand(time(NULL));
    FILE *fp = fopen("kurzusok.txt", "w");

    int i, j;
    for (i = 0; i < N; i++) {
        int osszeg = 0;
        for (j = 0; j < M; j++) {
            oktatok[i].pontok[j] = rand() % 101;
```

```
            osszeg += oktatok[i].pontok[j];
        }
        oktatok[i].atlag = (float)osszeg / M;

        if (oktatok[i].atlag >= 85) oktatok[i].kat = 'A';
        else if (oktatok[i].atlag >= 70) oktatok[i].kat = 'B';
        else if (oktatok[i].atlag >= 55) oktatok[i].kat = 'C';
        else if (oktatok[i].atlag >= 40) oktatok[i].kat = 'D';
        else oktatok[i].kat = 'F';

        fprintf(fp, "%s ", oktatok[i].nev);
        for (j = 0; j < M; j++)
            fprintf(fp, "%d ", oktatok[i].pontok[j]);
        fprintf(fp, "Atlag: %.1f Kat: %c\n", oktatok[i].atlag,
oktatok[i].kat);
    }

    fclose(fp);
    printf("Oktatoi pontok es kategoriak mentve fajlba.\n");
    return 0;
}
```

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szamitastechnika:fajlkezeles

Last update: **2025/10/29 08:42**