

Blatt 12 Ein- und Ausgabe mit Dateien

Aufgabe 12.1 Formatiert auf Platte schreiben bzw. von Platte lesen

- a) Schreiben Sie ein Programm, das im Dialog N Zahlen einliest und formatiert in eine Datei schreibt.

Hinweis: Wenn Sie als Dateiname bei der Funktion `fopen()` einen String wie etwa "ZAHLEN.DAT" angeben, wird die Datei im aktuellen Arbeitsverzeichnis abgelegt. Was das gerade ist, hängt davon ab, wie Sie Ihr Programm starten, welche Entwicklungsumgebung Sie benutzen, etc. Falls Sie diese Datei dann in Aufgabe 12.1.b einlesen wollen, müssen Sie wissen, wo sich die Datei befindet. Suchen Sie die Datei mit dem Windows-Explorer, damit Sie wissen, wo solche Dateien abgelegt werden.

Um diese Probleme zu umgehen, ist es üblich, dass 2 Programme über ein festes Verzeichnis miteinander kommunizieren, in dem die gemeinsam bearbeiteten Dateien abgelegt werden. Legen Sie in Ihrem Home-Verzeichnis ein Verzeichnis "Blatt12" an und verwenden Sie dieses für die folgenden Aufgaben ebenso. Geben Sie dann als String bei `fopen()` "Z:/Blatt12/ZAHLEN.DAT" an

- b) Schreiben Sie ein zweites Programm, das den Durchschnitt dieser Zahlen berechnet.

Aufgabe 12.2 Zeichenweise Ein/Ausgabe mit `fputc()` und `fgetc()`

- a) Testen Sie die Programme `fput.c` und `fget.c`. Beachten Sie für die Dateinamen bei den Aufrufen von `fopen()` den Hinweis zu Aufgabe 12.1!

```
/* Datei fput.c */
/* Anhaengen einer interaktiv eingegebenen Zeile an die Datei TEST.BIN
*/

#include <stdio.h>

int main (void)
{
    FILE * fp;
    int    c;
    if ((fp = fopen ("TEST.BIN", "a")) == NULL)
        fprintf (stderr,
                 "Kann Datei %s nicht eroeffnen\n",
                 "TEST.BIN");
    else {
        printf("\nGeben Sie Zeichen ein, Ende durch ^Z:\n");
        while ((c = getchar ()) != EOF) fputc (c, fp);
        /* Tastatureingaben an das Ende der Datei haengen */
        fclose(fp);
    }
    return 0;
}

=====
```

```

/* Datei fget.c */
#include <stdio.h>

int main (void)
{
    FILE * fp;
    char puffer [81];
    int  c;
    int  i;

    fp = fopen ("TEST.BIN", "r");
    /* Liest eine Textzeile aus der Datei in einen String.      */

    for (i = 0; (i < 80) &&
           ((c = fgetc (fp)) != EOF) &&
           (c!= '\n');i++)
        puffer [i] = c;
    puffer [i] = '\0';
    printf("\nDie Zeile lautete:\n%s", puffer);
    fclose (fp);
    return 0;
}

```

- b) Erweitern Sie das Programm fget.c so, dass es nicht nur die erste Zeile der Datei ausliest, sondern die ganze Datei. Verwenden Sie die Funktion feof()!

Aufgabe 12.3 Stringweise Ein/Ausgabe fputs() und fgets()

- a) Testen Sie die folgenden Programme fputs.c und fgets.c. Beachten Sie für die Dateinamen bei den Aufrufen von fopen() den Hinweis zu Aufgabe 12.1!

```

/* Datei: fputs.c */

#include <stdio.h>

/* Anhaengen einer Zeichenkette an die Datei FPUTS.BIN */

int main (void)
{
    FILE * fp;
    char * string =
        "Zeichenkette, die in die Datei geschrieben wird\n";

    if ((fp = fopen ("FPUTS.BIN", "a")) == NULL)
        fprintf (stderr, "Kann die Datei nicht oeffnen \n");
    else
    {
        fputs (string, fp);
        fclose (fp);
    }
    return 0;
}

```

=====

```

/* Datei: fgets.c*/

#include <stdio.h>

/* Liest die 1. Zeile aus der Datei FPUTS.BIN in einen
   Puffer ein*/

int main (void)
{
    FILE * fp;
    char buffer[81];

    if ((fp = fopen ("FPUTS.BIN", "r")) == NULL)
        fprintf (stderr, "\n Kann Datei nicht oeffnen\n");
    else
    {
        if (fgets (buffer, 81, fp) != NULL)
            printf ("\n 1. Zeile in der Datei FPUTS.BIN:\n %s",
                    buffer);
        fclose (fp);
    }
    return 0;
}

```

- b) Geben Sie alle Zeilen der Datei und die Anfangsadresse des Puffers *buffer* am Bildschirm aus!

Aufgabe 12.4 Blockweise Ein/Ausgabe `fwrite()` und `fread()`

- a) Testen Sie die Programme `fschreib.c` und `flies.c`! Beachten Sie für die Dateinamen bei den Aufrufen von `fopen()` den Hinweis zu Aufgabe 12.1!

```

/* Datei fschreib.c          */

#include <stdio.h>

int main (void)
{
    int buffer [] = {0,1,2,3,4,5,6,7};
    FILE * fp;
    size_t ergebn;

    if ((fp = fopen ("TEST.DAT", "wb")) == NULL)
        printf ("\nKann Datei nicht oeffnen");
    else {
        ergebn = fwrite (buffer, sizeof(int), 8 , fp);
        printf ("\n%i Zahlen in die Datei geschrieben\n", ergebn);
        fclose (fp);
    }

    return 0;
}

```

=====

```
/* Datei flies.c          */
#include <stdio.h>

int main (void)
{
    int buffer [10];
    FILE * fp;
    int ergebn;
    int LV = 0;

    if ((fp = fopen ("TEST.DAT", "r")) == NULL)
        printf ("\nKann Datei nicht oeffnen");
    else
    {
        ergebn = (int) fread (buffer, sizeof(int), 7 , fp);
        printf ("\n");
        for (LV = 0; LV < ergebn; LV ++ )
        {
            printf ("\nin buffer[%i] wurde aus der Datei ", LV);
            printf ("TEST.BIN die folgende Zahl eingelesen: %i",
                    buffer[LV]);
        }
        if (feof (fp)) printf ("\nDateiende erreicht");
        fclose (fp);
        printf ("\ninsgesamt %i Zahlen aus der Datei gelesen",
                ergebn);
    }
    return 0;
}
```

- b) Schreiben Sie in fschreib.c nach den *int*-Zahlen 0,1,...7 zusätzlich die *double*-Zahlen 0.5, 1.5, 2.5 und 3.5, in die Datei.
- c) Ändern Sie flies.c so ab, dass nicht nur die 8 *int*-Zahlen, sondern auch die 4 *double*-Zahlen eingelesen werden.