6.2.V Extras/Lesen aus Dateien – Versuch

6.2.V.1 Ziel

Es sollen Gleitkomma-Zahlenwerte aus einer Datei mit dem Namen `zahlen.txt` gelesen und in ein Array geschrieben werden. Bei Auftreten eines Fehlers soll eine Meldung ausgegeben werden.

6.2.V.2 Hilfsmittel

- Variable `FILE *dm` für die Datei
- Funktionen `fopen`, `fclose` und `fscanf` sowie `perror`
- Array-Variable `double array[MAX_ANZ]`
- Anzahl der in das Array eingelesenen Werte `int anzahl`

6.2.V.3 Schritt 1

Schreiben Sie das Programm so weit, dass Sie die Datei öffnen und schließen. Bei einem Fehler soll eine Meldung ausgegeben werden.

6.2.V.4 Schritt 2

```
lies Zahl aus datei nach wert
solange gelesen
    gib wert aus
    lies Zahl aus datei nach wert
```

Ergänzen Sie das Programm so, dass Sie nacheinander jeden Wert aus der Datei in eine Variable `double wert` lesen. Dieser Wert soll sofort ausgegeben werden.

6.2.V.5 Schritt 3

```
anzahl <- 0
lies Zahl aus datei nach array[anzahl]
solange gelesen und anzahl<MAX_ANZ
    gib array[anzahl] aus
    erhoehe anzahl um 1
    lies Zahl aus datei nach array[anzahl]
```

Modifizieren Sie das Programm so, dass der gelesene Wert in ein Element des Arrays gebracht wird. Damit nicht jeder Wert in dasselbe Array-Element gebracht wird, braucht man einen Zähler, der sich die aktuelle Array-Position merkt. Dieser Zähler muss nach dem Schreiben in das Array um eins hochgezählt werden. Damit kann dieser Zähler auch sagen, wie viele Elemente schon gelesen und in das Array geschrieben wurden. Achten Sie darauf, dass nicht mehr Werte geschrieben werden als das Array Elemente besitzt!

6.2.V.6 Schritt 4

Ergänzen Sie das Programm unterhalb der Lese-Schleife um eine Fehlerprüfung. Folgende Fehler sind möglich:

- Lesefehler (`ferror(dm)!=0`)
- Fehler im Dateiformat (`rc==0`)
- zu große Datei (`rc==1 && anzahl==MAX_ANZ`)

6.2.V.7 Lösungen

Schritt 1: Nur öffnen und schließen

```
#include <stdio.h>
int main(void)
{
    FILE *dm;
    dm=fopen("datei.txt", "r");
    if(dm==NULL)
    {
        perror("datei.txt");
        return 1;
    }
    /* ... hier ergaenzen: Schritte 2 und 3 */
    fclose(dm);
    return 0;
}
```

Schritt 2: Nur einlesen bis zum Ende, aber ohne Array

```
int rc;
double wert;
rc=fscanf(dm, "%lf", &wert);
while(rc==1)
{
    printf("Gelesen: %lf\n", wert);
    rc=fscanf(dm, "%lf", &wert);
}
```

Schritt 3: Schreiben des Wertes in ein Array

```
int rc, anzahl=0;
double array[MAX_ANZ];
rc=fscanf(dm, "%lf", &array[anzahl]);
while(rc==1 && anzahl<MAX_ANZ)
{
    printf("Gelesen: Pos.: %i, Wert:%lf\n",
                    anzahl, array[anzahl]);
    ++anzahl;
    rc=fscanf(dm, "%lf", &array[anzahl]);
}
/* ... hier ergaenzen: Schritt 4 */
```

Schritt 4: Herausfinden, ob Dateiende oder Fehler

```
if(ferror(dm))
    printf("Fehler beim Lesen.\n");
else if(rc==0)
    printf("Fehler im Dateiformat.\n");
else if(rc==1 && anzahl==MAX_ANZ)
  printf("Datei zu gross.\n");
else
  printf("Datei vollstaendig eingelesen.\n");
```