

Fachhochschule Bingen

Programmieren

Dateiein- und -ausgabe 2:

Zeichen orientierte Dateiein- und Ausgabe, Binäre Dateien

Prof. Dr. Maximilian Mengel,
Professur Programmiermethodik,
Grundlagen der Informatik und Multimedia
Gebäude 1, Raum 212
Tel.: 06721-409 152
E-Mail: mengel@fh-bingen.de

Funktionen zur Zeichenorientierten und zur binären Dateiein- und -ausgabe

- `fopen()` Öffnen einer Datei
- `fclose()` Schließen einer Datei
- `fgetc()` Einlesen eines Zeichens
- `fputc()` Schreiben eines Zeichens
- `fread()` Binäres Lesen
- `fwrite()` Binäres Schreiben
- `fseek()` Positionieren des Lese-/Schreibkopfs

28.11.2003

2

`fgetc()`

- Die Funktion `fgetc()` liest aus einer zuvor geöffneten Datei ein Zeichen:

```
zeichen = fgetc(fp)
```
- Folgender Parameter wird übergeben:
 - `FILE *fp`
- Die Funktion liefert als Rückgabewert:
 - `char zeichen`
 - Das eingelesene Zeichen wird zurückgegeben. Wenn ein Fehler auftrat, wird EOF zurückgegeben

28.11.2003

3

`fputc()`

- Die Funktion `fputc()` schreibt in eine zuvor geöffnete Datei ein Zeichen:

```
res_Zeichen = fputc(fp, zeichen)
```
- Folgende Parameter werden übergeben:
 - `FILE *fp`
 - `char zeichen`
- Die Funktion liefert als Rückgabewert:
 - `char res_Zeichen`
 - Das erfolgreich ausgegebene Zeichen wird auch zurückgegeben. Wenn ein Fehler auftrat, wird EOF zurückgegeben

28.11.2003

4

getchar und putchar

- Wie printf() und fprintf() einmal für die Bildschirm orientierte bzw. Datei orientierte Ausgabe existieren, so existieren auch entsprechende zeichenorientierte Funktionen für die Standardein- und -ausgabe:

```
1 zeichen = getchar();  
1 putchar(zeichen);
```

fread()

- Die Funktion fread() liest aus einer zuvor geöffneten Datei binär gespeicherte Daten:

```
resultat = fread(adresse,size,anzahl,fp)
```

- Folgende Parameter werden übergeben:

1 void *adresse	Adresse bei der die eingelesenen Daten abgelegt werden
1 size_t size	Größe eines einzulesenden Datums
1 size_t anzahl	Anzahl der einzulesenden Daten
1 FILE *fp	Verweis auf die Datei

- Die Funktion liefert als Rückgabewert:

1 size_t resultat	Anzahl der gelesenen Daten
-------------------	----------------------------

fwrite()

- Die Funktion fwrite() schreibt in eine zuvor geöffnete Datei binäre Daten:

```
resultat = fwrite(adresse,size,anzahl,fp)
```

- Folgende Parameter werden übergeben:

1 void *adresse	Adresse bei der die auszugebenden Daten abgelegt sind
1 size_t size	Größe eines auszugebenden Datums
1 size_t anzahl	Anzahl der auszugebenden Daten
1 FILE *fp	Verweis auf die Datei

- Die Funktion liefert als Rückgabewert:

1 size_t resultat	Anzahl der geschriebenen Daten
-------------------	--------------------------------

fseek()

- Die Funktion fseek() positioniert den Lese-Schreibkopf an eine bestimmte Stelle:

```
resultat = fseek(fp,offset,position)
```

- Folgende Parameter werden übergeben:

1 FILE *fp	Verweis auf die Datei
1 long offset	Anzahl der zu versetzenden Byte
1 int position	Dateianfang: SEEK_SET Dateiende : SEEK_END Aktuelle Position: SEEK_CUR

- Die Funktion liefert als Rückgabewert:

1 int resultat	0 = OK; sonst Fehler
----------------	----------------------

Beispiel

```
#include <stdio.h>
main()
{
    FILE *fp;
    int zahlen[]={1,2,3,4,5}, a,i;
    if ((fp = fopen("test.bin","w")) != NULL)
    {
        fwrite((void*)zahlen, sizeof(int),5,fp);
        fclose(fp);
    }
    else fprintf(stderr,"Ein Fehler beim Öffnen zum schreiben!\n");
    for (i=0; i<5;++i) zahlen[i]=0;
    if ((fp = fopen("test.bin","r")) != NULL)
    {
        fread((void*)zahlen, sizeof(int),4,fp);
        fread((void*)&a, sizeof(int),1,fp);
        fclose(fp);
    }
    else fprintf(stderr,"Ein Fehler beim Öffnen zum lesen!\n");
    fprintf(stdout,"zahlen[:");
    for (i=0; i<5;++i) fprintf(stdout,"%d ",zahlen[i]);
    fprintf(stdout,"\n a= %d\n",a);
}
```
