

## Fachhochschule Bingen

### Programmieren

#### Dateiein- und -ausgabe

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](mailto:mengel@fh-bingen.de)

#### Verzeichnisse und Dateien

- Daten und Programme werden im Rechner in Dateien gespeichert.
- In einem Verzeichnis (directory) werden Informationen über die enthaltenen Dateien geführt: Name, Länge, Erzeugungsdatum, Datum der letzten Änderung (sowie weitere wie z.B. Besitzer, Zugriffsrechte...).
- Dateiverzeichnisse sind im Allgemeinen hierarchisch angeordnet.
- Es gibt ein sogenanntes Wurzelverzeichnis.
- In einem Verzeichnis können Dateien und weitere Verzeichnisse angelegt sein.

28.11.2003

2

#### Zugriff auf Dateien

- Über Ein- und Ausgabemöglichkeiten können Programme lesend oder schreibend auf Dateien bzw. auf die enthaltenen Daten zugreifen.
- Dadurch können:
  - Berechnungsergebnisse dokumentiert,
  - Ausgaben zum Zwischenspeichern (d.h. zum späteren wieder Einlesen) erzeugt,
  - Protokolle (Zwischenschritte) abgelegt,
  - Ergebnisse an andere Programme (Austausch von Daten) weitergegeben werden oder
  - Programme in gewissem Rahmen konfigurierbar gehalten werden.

28.11.2003

3

#### Dateihandling I

- Umgang mit Dateien :
  - Dateien können zum Lesen oder Schreiben geöffnet werden.
  - Es ist auch möglich, gleichzeitig eine Datei zum Lesen und Schreiben zu öffnen; dies sollte möglichst vermieden werden
  - Dateivariablen sind vom Typ FILE (Bsp: FILE \*eingabe).
  - Es muß eine Verbindung zwischen dem Programm und der Datei hergestellt werden (Befehl: fopen). Der Schreib- bzw. Lesekopf wird positioniert.

28.11.2003

4

## Dateihandling II

### ■ Umgang mit Dateien:

- Mit jeder Lese- bzw. Schreibanweisung (fscanf, fprintf) wird angegeben, auf welche Datei zugegriffen werden soll und welche Daten (Art und Reihenfolge) gelesen oder geschrieben werden sollen.
- Durch einen Format-String werden folgende Aspekte festgelegt:
  - Der Typ
  - Die Zuordnung der Variablen zu den Ein- bzw. Ausgaben

28.11.2003

5

## Dateihandling III

### ■ Umgang mit Dateien:

- Der Zugriff auf die Datei erfolgt immer an der aktuellen Position des Lese-/Schreibkopfes.
- Der Lese- bzw. Schreibkopf wird unmittelbar danach weiter geschoben.
- Am Ende der Eingabe/Ausgabe sollte die Verbindung zwischen Datei und Programm wieder beendet und die Datei geschlossen werden (fclose).

28.11.2003

6

## Zugriff auf Datei

- Für die Kommunikation mit Dateien gibt es in der C-Bibliothek `stdio.h` den Typ **FILE**.
- Kommunikation mit einer Datei:

```
FILE *fp; /*Vereinbarung einer Variable, die auf eine Datei zeigt*/
fp = fopen (name, mode);
/* Geöffnet wird die Datei mit dem Namen 'name'; die Zuweisung
   erfolgt an die Variable fp; die Datei kann:
       wenn mode = "r" ist: gelesen werden
       wenn mode = "w" ist: geschrieben werden
       wenn mode = "a" ist: erweitert werden
*/
fclose (fp); /* Datei schließen */
```

28.11.2003

7

## Funktionen zur Formatierten Dateiein- und -ausgabe

- |             |                             |
|-------------|-----------------------------|
| ■ fopen()   | Öffnen einer Datei          |
| ■ fclose()  | Schließen einer Datei       |
| ■ fprintf() | Datei-Variante von printf() |
| ■ fscanf()  | Datei-Variante von scanf()  |
| ■ feof()    | Ist das Dateende erreicht?  |

28.11.2003

8

## fopen()

- Zum Öffnen einer Datei wird fopen() genutzt:  
`fp = fopen(name,modus)`
- Die Funktion besitzt zwei Parameter:
  - `char *name`
    - Der Name (eventuell mit Pfad) der zu öffnenden Datei
  - `char *modus`
    - Der Modus, in dem die Datei geöffnet werden soll ( "r" für lesen; "w" für schreiben; "a" für erweitern)
- Die Funktion besitzt einen Rückgabewert:
  - `FILE *fp`
    - Der „Verweis“ auf die geöffnete Datei; trat beim Öffnen ein Fehler auf, wird die Konstante NULL zurückgegeben

28.11.2003

9

## fclose() / feof()

- fclose() schließt eine zuvor geöffnete Datei:
  - `fclose(fp)`
  - Parameter: „Verweis“ auf geöffnete Datei:
    - `FILE *fp`
  - Rückgabewert vom Typ Integer:
    - `int resultat` (wird praktisch nie genutzt)
- feof() überprüft ob das Dateiende erreicht ist
  - `resultat = feof(fp)`
  - Parameter: „Verweis“ auf geöffnete Datei
  - Rückgabewert vom Typ Integer
    - ungleich Null wenn Dateiende erreicht

28.11.2003

10

## fprintf()

- fprintf() schreibt in eine zuvor geöffnete Datei:  
`resultat = fprintf(fp,"%d",x)`
- Folgende Parameter werden übergeben:
  - `FILE *fp`
  - `char *format`
    - Formatstring (analog zu printf())
  - Ausdrücke entsprechend dem Formatstring
- Rückgabewert :
  - `int resultat`
    - Bei Erfolg: die Anzahl der ausgegebenen Werte; sonst EOF

28.11.2003

11

## fscanf()

- fscanf() liest aus einer zuvor geöffneten Datei:  
`resultat = fscanf(fp,"%d",&x)`
- Folgende Parameter werden übergeben:
  - `FILE *fp`
  - `char *format`
    - Formatstring (analog zu scanf())
  - Adressen von Variablen
- Rückgabewert :
  - `int resultat`
    - Die Anzahl der eingelesenen Werte; sonst EOF

28.11.2003

12

## Standard Ein- und Ausgabe

---

- Folgende Dateiströme sind immer verfügbar sobald `stdio.h` eingefügt ist:
  - `stdin`: Standard-Eingabe (die Tastatur)
  - `stdout`: Standard-Ausgabe (der Bildschirm)
  - `stderr`: Standard-Fehler-Ausgabe (im Allgemeinen auch der Bildschirm)
- Die Funktionen `printf()` und `scanf()` sind nur Abkürzungen für:
  - `fprintf(stdout, ... .. );`
  - `fscanf(stdin, ... .. );`

## Übung: Datei-IO

---

- Schreiben Sie ein Programm daß, eine Datei öffnet und
  - Die Anzahl der Worte zählt (jede durch Trennzeichen getrennte Zeichensequenz soll als Wort zählen)
  - Eine Statistik über die Wortlängen erstellt
  - Sowohl die Anzahl der Worte als auch die Statistik in einer Datei `Statistik.txt` ablegt.