

Dateiarbeit in C

- Datei(engl. File) ist eine Menge von Daten (Bytes) auf einem geeigneten Datenträger.
 - Festplatte
 - USB-Stick
 - Früher: Magnetband, Lochband, Lochkartenstapel, Diskette
- Eine Datei enthält Daten und ist in der Regel über ein Verzeichnis, einen Namen identifizierbar.

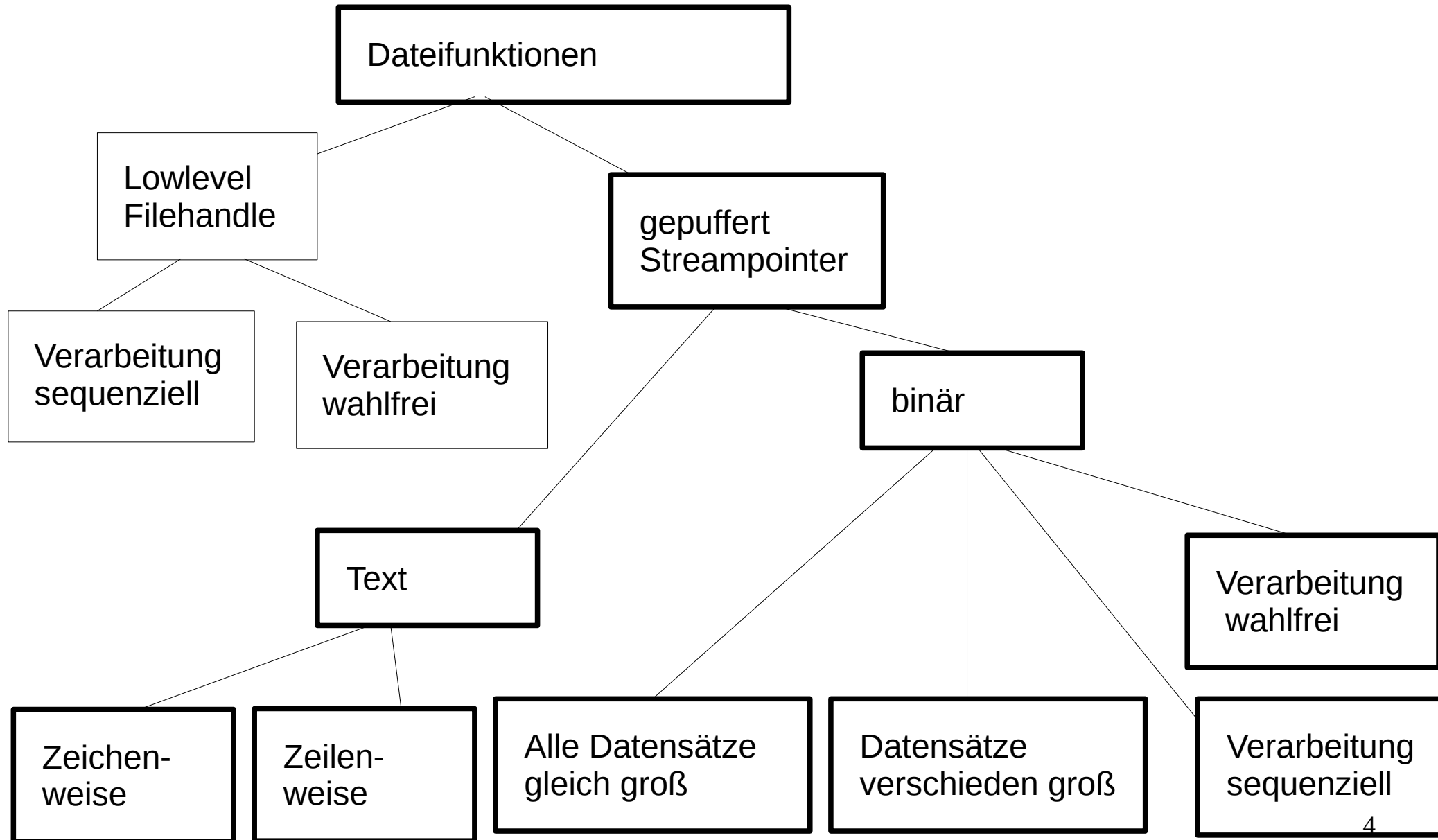
Datenströme

- Bei der Verarbeitung der Daten einer Datei spricht man vom Datenstrom (Stream (I/O))
- Werden die Daten vom Datenträger in den programminternen Speicher gelesen, so spricht man von Eingabestrom
- Werden die Daten auf den Datenträger aus dem programminternen Speicher geschrieben, so spricht man von Ausgabestrom

high level I/O/ low level I/O

- Es wird zwischen high level I/O/ low level I/O unterschieden.
- Low level I/O arbeitet sehr Systemnah, die Verbindung zur Datei wird über einen Integerwert, den sog. Filedeskriptor oder Filehandle(Win) hergestellt.
- High level I/O arbeitet mit einem Streampointer, einem Pointer vom Typ FILE*. high level I/O arbeitet gepuffert.
- Im Weiteren wird zunächst high level I/O betrachtet.

Dateiarbeit in C



Datei Öffnen mit fopen

Dateiname
Incl. Pfad

Modus

```
FILE * pf;  
pf=fopen("myFile.txt","rt");
```

b/t	Texdatei ([t])	Binärdatei(b)
r	Zum Lesen öffnen	
w	Zum Schreiben (überschreiben) öffnen, ggf. erzeugen	
a	Zum Schreiben am Dateiende öffnen, ggf. erzeugen	
r+	Zum Lesen und Schreiben öffnen (ändern)	
w+	Zum Lesen und Schreiben öffnen (überschreiben), ggf. erzeugen	
a+	Zum Anfügen, Lesen, Erzeugen öffnen, ggf. erzeugen	

Lesen / Schreiben von Bytes

<u>Funktion</u>	<u>Verwendung</u>
<code>int fgetc(FILE* stream);</code>	Lesen des nächsten Zeichens (Bytes) als unsigned char oder EOF
<code>int fputc(int c, FILE* stream);</code>	Schreibt das Zeichen c in den Stream

Lesen und Schreiben von Text

<u>Funktion</u>	<u>Verwendung</u>
<code>int fgets(char*s, int n, FILE*stream) ;</code>	Lesen von max. n-1 Zeichen. Das Lesen wird vorher beendet, wenn ein Zeilentrenner gefunden wird, dieser wird mit eingelesen. Eine terminierende 0 wird angefügt.
<code>int fputs(const char*s, FILE* stream) ;</code>	Schreibt die nullterminierte Zeichenkette + ein \n in den Stream.

Lesen und Schreiben von Binärdaten

Funktion	Verwendung
<code>size_t fread(void* ptr, size_t size, size_t n FILE* stream)</code>	Liest aus dem Stream max. n Objekte der Größe size und speichert sie auf die Adresse ptr. Es muß genügend Speicher zur Verfügung stehen, um die Daten aufzunehmen. Der Returnwert liefert die Anzahl der gelesenen Objekte. (feof, oder ferror liefern Fehlerzustand)
<code>size_t fwrite(void* ptr, size_t size, size_t n, FILE* stream)</code>	Schreiben von max. n Objekten der Größe size. Der Returnwert liefert die Anzahl der geschriebenen Objekte.

Positionierung in Datei

Funktion	Verwendung
<code>int fseek(FILE* stream, long offset, int origin);</code>	Setzt die Dateiposition auf den Wert Offset, relativ zuDateiAnfang (origin=SEEK_SET), Dateiende(origin=SEEK_END) oder zum aktuellen Stand(origin=SEEK_CUR)
<code>long ftell(FILE* stream);</code>	Liefert die aktuelle Lese-/Schreibposition
<code>void rewind(FILE* stream);</code>	Stellt die Lese-/Schreibpos. Auf den Dateianfang