

- Klassische Fehlerquelle: Verarbeitung von **EOF**
  - **EOF** (End of File): **Negative** Ganzzahl-Konstante (häufig -1) aus **stdio.h**
  - Rückgabewert von Bibliotheksfunktionen, wenn das Dateiende **überschritten** wurde  
z.B. `int fgetc (FILE *stream)` aus **stdio.h**

```
FILE      *file = fopen(..., "rb");
signed char ch = '\0';
while( (ch = fgetc(file)) != EOF ) { ... }
```

$(ch = fgetc(file)) \quad != \quad EOF$

$\underbrace{\hspace{1.5cm}}_{-1 \text{ (signed char)}} \quad \underbrace{\hspace{1.5cm}}_{-1 \text{ (int)}}$

**implizite unäre Konvertierung:**

$-1 \text{ (int)} \quad -1 \text{ (int)}$

**implizite binäre Konvertierung:**

$-1 \text{ (int)} \quad != \quad -1 \text{ (int)}$

```
FILE      *file = fopen(..., "rb");
unsigned char ch = '\0';
while( (ch = fgetc(file)) != EOF ) { ... }
```

$(ch = fgetc(file)) \quad != \quad EOF$

$\underbrace{\hspace{1.5cm}}_{255 \text{ (unsigned char)}} \quad \underbrace{\hspace{1.5cm}}_{-1 \text{ (int)}}$

**implizite unäre Konvertierung:**

$255 \text{ (int)} \quad -1 \text{ (int)}$

**implizite binäre Konvertierung:**

$255 \text{ (int)} \quad != \quad -1 \text{ (int)}$

*Endlosschleife*

- **Weiteres Problem:** Für Binärdateien wird der gesamte Wertebereich eines 8bit-char benötigt, **EOF** entspräche hier also einem "echten" Symbol
- **Lösung:** Zuweisung auf **int**-Variable