Dateien in Modula-3

- **Dateisystem**
- **Operationen auf Dateien**
  - Lesen
  - Schreiben
- **Dateien in Modula-3**
  - wichtige Datei-Operationen
- **Konvertierung**

Dateien: Zweck

- **Verarbeiten von Daten**
  - Daten müssen in vielen Fällen dauerhaft (***persistent***) gespeichert werden
- **Bisher:**
  - Daten wurden im ***Arbeitsspeicher*** zur Laufzeit des Programms erzeugt
  - Nachdem das Programm beendet ist, sind diese Daten ***verloren***
  - "***flüchtiger***" Speicher
- **Hintergrundspeicher**
  - persistenter Speicher
  - Diskette, CD, Festplatte etc.
- **Frage:**
  - Wie können wir Daten aus dem Arbeitsspeicher in den Hintergrundspeicher bringen und umgekehrt?

Dateisystem

- **Betriebssystem:**
  - stellt ***Dienstleistungen*** zur Verfügung, damit der Umgang mit dem Rechner einfach und mit ***bestimmten Diensten*** möglich ist
  - eine angebotene Dienstleistung des Betriebssystems ist das ***Dateisystem***
- **Dateisystem**
  - erlaubt, den Hintergrundspeicher in ***einzelne Bereiche*** aufzuteilen
  - diese nennt man ***Dateien*** (file)
  - Dateien können in sogenannten ***Verzeichnissen*** gruppiert werden (directory)
  - jede Datei hat einen ***Namen***
  - aus der Sicht des Rechners ist eine Datei eine Folge von ***Informationseinheiten*** (Bytes)
  - ein Verzeichnis verwaltet für alle seine Dateien den Namen und die Position, wo die Dateien physisch auf dem Hintergrundspeicher liegen

Dateien: Zweck und Formen

- **Dateien können**
  - angelegt und gelöscht werden
  - gelesen und geschrieben werden
- **Dabei gibt es zwei Dateiformen:**
  - ***Sequentiell***
    - Daten werden von Anfang bis zum Schluß der Datei ***nacheinander*** gelesen (geschrieben).
    - Es gibt keine Möglichkeit, einen Ausschnitt der Datei zu ***überspringen***
  - ***Direkt***
    - Es kann eine ***Position*** angegeben werden, ab der gelesen (geschrieben) werden soll
    - Wechseln der Position ist ***beliebig*** möglich
- **Das Betriebssystem**
  - kennt für jede Datei einen Zähler, der die aktuelle Position kennzeichnet.

Arbeiten mit Dateien - allgemein

- **Programmiersysteme**
  - bieten in der Regel Möglichkeiten und Mechanismen, um vom Programm aus ***Dateioperationen*** ausführen zu können
  - dabei werden i.d.R. nicht die Funktionen des Betriebssystems benutzt
  - Programmiersysteme stellen eine ***abstrakte Schnittstelle*** zum Dateisystem zur Verfügung

Arbeiten mit Dateien - Wichtige Dateioperationen

- **Datei *öffnen***
  - öffnen zum Lesen
  - öffnen zum Schreiben (von vorne)
  - öffnen zum Schreiben (neue Zeichen werden hinten angehängt)
- **neue Datei *erzeugen***
- ***lesen* und *schreiben***
- **Abfragen des *Dateiendes* (end of file, EOF)**
- **Datei *schließen***
  - um sicherzugehen, daß alle Änderungen in einer Datei persistent sind

Sprachumgebung, E/A-Strom

- **Sprachumgebung von Modula-3**
  - stellt Module zur Verfügung, um Dateien manipulieren zu können
- **Ein- / Ausgabestrom**
  - Mit einem Ein- / Ausgabestrom kann auf eine Datei zugegriffen werden
  - In der Standardbibliothek sind diese definiert
    - `reader, writer`
  - Ein Ein- / Ausgabestrom kann bei seiner Initialisierung mit einer Datei verbunden werden

```
IMPORT IO, Rd, Wr;
VAR eingabestrom : Rd.T;
    ausgabestrom : Wr.T;
…
eingabestrom := IO.OpenRead ("eingabe.txt");
ausgabestrom := IO.OpenWrite("ausgabe.txt");
```


Dateioperationen in Modula-3 - Das Modul IO

- PROCEDURE **EOF** (rd: Rd.T := NIL): BOOLEAN;
  - Liefert TRUE gdw. das Ende der Datei rd erreicht wurde
- PROCEDURE **OpenRead** (f: TEXT): Rd.T;
  - Öffnet die Datei mit dem Namen f zum Lesen und liefert einen Eingabestrom auf ihren Inhalt.
  - Wenn die Datei nicht existiert oder sie nicht gelesen werden kann, so wird NIL geliefert.
- PROCEDURE **OpenWrite** (f: TEXT): Wr.T;
  - Öffnet die Datei mit dem Namen f zum Schreiben liefert einen Ausgabestom auf ihren Inhalt.
  - Wenn die Datei nicht existiert, so wird sie geschaffen.
  - Darf der Prozeß die Datei nicht modifizieren oder erschaffen, dann wird NIL zurückgeliefert.

Dateioperationen in Modula-3 - Das Modul Rd (1)

- Eine Variable vom Typ **Rd.T** identifiziert einen Eingabestrom.
- PROCEDURE **GetChar** (rd: T): CHAR RAISES {EndOfFile, Failure, Alerted};
  - Liest ein Zeichen vom Eingabestrom rd an der aktuellen Position und erhöht dessen aktuelle Position um 1.
- PROCEDURE **EOF** (rd: T): BOOLEAN RAISES {Failure, Alerted};
  - Liefert TRUE gdw. das Ende der Datei rd erreicht ist.
- PROCEDURE **UnGetChar** (rd: T): CHAR RAISES {};
  - Legt das zuletzt gelesene Zeichen auf Rd zurück.

Dateioperationen in Modula-3 - Das Modul Rd (2)

- PROCEDURE **Close** (rd: T) RAISES {Failure, Alerted};
  - Schließt rd,
  - d.h. alle mit rd verbundenen Ressourcen werden freigegeben und closed(rd) wird auf TRUE gesetzt.
- PROCEDURE **GetLine** (rd: T): TEXT RAISES {EndOfFile, Failure, Alerted};
  - Liest so viele Zeichen von rd, bis das Ende von rd erreicht wurde oder ein Zeilenvorschubzeichen gelesen wurde.
  - Die Ausnahme EndOfFile wird generiert, wenn das Ende von rd schon vor der Operation erreicht war.
  - Die gelesenen Zeichen werden zurückgegeben. Das Zeilenvorschubzeichen wird nicht im Ergebnis abgelegt.
- **Weitere Operationen zur**
  - Bestimmung der aktuellen Position von rd
  - Bestimmung der Länge von rd
  - ...

Dateioperationen in Modula-3 - Das Modul Wr (1)

- Eine Variable vom Typ **Wr.T** identifiziert einen Ausgabestrom.
- PROCEDURE **PutChar** (wr: T; ch: CHAR) RAISES {Failure, Alerted};
  - Schreibt das Zeichen ch auf den Ausgabestrom wr und erhöht dessen aktuelle Position um 1.
- PROCEDURE **Close** (wr: T) RAISES {Failure, Alerted};
  - Schließt wr,
  - d.h. alle mit wr verbundenen Ressourcen werden freigegeben und closed(wr) wird auf TRUE gesetzt.
- PROCEDURE **PutText** (wr: T; t: TEXT) RAISES {Failure, Alerted};
  - Schreibt alle Zeichen von t auf wr.

Dateioperationen in Modula-3 - Das Modul Wr (2)

- PROCEDURE **Flush** (wr: T) RAISES {Failure, Alerted};
  - Leert den Puffer von wr.
- **Weitere Operationen zur**
  - Bestimmung der Länge von wr
  - Bestimmung der aktuellen Position von wr
  - ...

Konvertierung

- Rd und Wr haben nur Prozeduren zum ***Lesen und Schreiben von Zeichen*** (Text ist eine Sequenz von Zeichen)
- Sollen Zahlen oder Boolesche Werte gespeichert werden, so müssen diese erst formatiert werden, d.h.
  - Konvertierung der Werte in jeweilige Text-Repräsentation
  - beim Lesen Konvertierung des Textes in jeweiligen Typ („scannen“)
- Dieses leisten die Module Fmt und Scan.

Das Modul Fmt

- Mit Hilfe der Prozeduren der Schnittstelle Fmt können Zahlen und andere Daten ***in TEXT umgewandelt*** werden.
- Dies wird u.a. geliefert für folgende Typen:
  - BOOLEAN (Prozedur Bool)
  - INTEGER (Prozedur Int)
  - CHAR (Prozedur Char)
  - REAL (Prozedur Real)
- Weitere Prozeduren zur Formatierung wie z.B. Längenanpassung, ...

Das Modul Scan

- Mit Hilfe der Schnittstelle Scan können Zahlen und andere Daten ***aus TEXT-Variablen gelesen*** werden.
- Die Prozeduren lesen den TEXT-Parameter und konvertieren dessen Inhalt in Werte des jeweiligen Typs.
- Führende Leerzeichen und bei Zahlen führende Nullen werden weggelassen.
- Dies wird u.a. geleistet für die folgenden Typen:
  - BOOLEAN (Prozedur Bool)
  - INTEGER (Prozedur Int)
  - REAL (Prozedur Real)
  - CHAR (Prozedur Char)

Was haben wir gelernt?

- **Dateien für die persistente Speicherung**
- **Dateisysteme als Teil des Betriebssystems**
- **Dateiformen (sequentielle / direkte Dateien), Modi (Lese-, Schreibdateien)**
- **interne Dateien, externe Dateien und Bindung / Auflösung der Bindung beim Öffnen oder Schließen**
- **Anschluß von Dateien an das Programmiersystem über vordefinierte E/A-Bausteine**
- **Dateioperationen für Dateiformen, Eingabe- oder Ausgabedateien**
- **Konvertierung**

Glossar

- **Hintergrundspeicherarten**
- **flüchtiger, persistenter Speicher**
- **Betriebssystem, Dateisystem, Anschluß von seiten des Programmiersystems**
- **Datei (file), Verzeichnis (directory)**
- **externe, interne Dateien, Namen für beides**
- **Dateiformen, Eingabe-/Ausgabedatei**
- **Dateioperationen, Öffnen, Schließen, Lesen, Schreiben, Positionierung, Abfrage Dateiende**
- **Einschränkung der Operationen bei bestimmten Dateiformen, Ein- oder Ausgabedateien**