ソフトウェア概論 B

数学科 吉開範章, 渡辺俊一 (栗野 俊一)

おしらせ

□資料

http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino

□ネットワークケーブルは...

○前にあります。

□ソフトウエア概論Bの定期試験

○1月29日の1限目

前回のまとめ：構造体

- 構造体
  - 異なる型のデータをまとめて扱う方法
    - cf. 配列：同じ型のデータをまとめて扱う方法
  - 構造体の宣言 (構造体 person の宣言)
    - struct person { char name[NAME_SIZE]; int age };
  - 構造体型変数の定義 (構造体型変数 myfriend の宣言)
    - struct person myfriend;
  - 構造体型メンバの参照 (「.」 演算子)
    - myfriend.age = 20; // myfriend のメンバの age を参照
  - 構造体型へのポインタと、メンバの参照 (「->」 演算子)
    - (*p).m == p -> m
- 構造体の応用
  - 関数値としての構造体：関数の値として構造体型の値が利用可
  - 構造体の中の構造体：構造体型のメンバに再び構造体型が利用可
  - 時間を表現する構造体：システムで様々な構造体型を利用
    - struct tm;

ファイル

- ファイルとは？
  - 計算機内の外部記憶装置に保存されている一塊のデータ
    - 外部記憶装置：ハードディスク / DVD-ROM / USB メモリ etc..
  - OS (Windows/Linux など..) が管理する情報の単位
    - Explorer で表示されるアイコンは、ファイルに対応している事が多い
- ファイルの属性
  - 管理属性 (Explorer の「詳細表示」)
    - 名前 + . + 拡張子 (パス名 : フォルダ名 + ファイル名)

      explorer.exe (c:\windows\explorer.exe)

      拡張子は、ファイルの種別を表現するために付けられる事が多い
    - 作成日時
  - データ(内容)
    - バイトの並び (巨大な char 型配列と考えてよい)
    - ファイルサイズ (内容の大きさ / 配列サイズ)

ファイルの種類

- ファイルの種類 (厳密ではない)
  - 用途別
    - プログラム (explorer.exe) 何かの動作を記述 / 実行可能
    - データ (system.ini) 何かの情報を保持 / プログラムから利用
  - 形式別
    - テキスト : 「メモ帳」で開いて、人間が読めるもの一般
    - バイナリ : テキストファイル以外のもの全て
  - 管理別
    - システムファイル : OS (Windows) が利用
    - ユーザファイル : 利用者が自分で作る
- 様々な拡張子とその意味
  - .exe : 実行ファイル
  - .txt : テキストファイルの一般的な名前
  - .doc : Microsof Word の文章ファイル
  - .c : C 言語のプログラムファイル

ファイルの特性

- ファイルの特性 (データの保持能力として)
  - 永続性：プログラムの終了後もデータが保存されている
    - 電源を切ってもなくならない
  - 独立性：他のプログラムと共有できる
    - cf. explorer で操作が可能
    - 複数のプログラム間で情報を受け渡しができる
  - 大容量：メモリに比べて、大容量 (二桁程度大きい)
    - メモリ 2G / ハードディスク 60G / USB メモリなら事実上無限？
  - 低速度：メモリに比べて、低速 (二桁程遅い)

「ファイル処理」が必要な訳

- 従来のプログラム
  - 入力はキーボード
    - 大量のデータを入力するのは辛い / メモ帳で入力して copy & past ?
  - 出力は画面
    - 一画面内に入らないとスクロールして消えてしまう / copy & past してメモ帳に保存 ?
  - プログラムを終了するとデータは消失
    - 複数のプログラムでデータのやり取りができれば...
  - 扱えるデータサイズに限界
    - 主メモリに入らないデータは扱えない
- 「ファイル」の属性を利用して、より便利に
  - 永続性 : プログラムの扱うデータをファイルに保存して再利用
    - 復活の呪文 / セーブデータ
  - 独立性 : 複数のプログラムで共通のデータを共有出来る
  - 大容量 : 大きなデータも扱える
  - 低速度 : 使い所を考えないと、悲惨な事に..

ファイル処理 (Text p.289)

- C 言語からみた「ファイル」
  - ストリーム : 入出力の対象
    - cf. キーボード (入力) / 画面 (出力) / ファイル (入出力)
- ストリームとは
  - バイトデータ (char) 並び
  - 「順番(シーケンシャル)」に並んでおり、その順に届く
    - cf. 配列は、どこからでも利用できる (ランダムアクセス)
  - 入出力 (I/O : Input/Output) を抽象化したもの
- C 言語からファイルを利用するための仕組
  - ファイルポインタ (FILE 型)
    - ファイル内のデータを扱う為の窓口
  - I/O ライブラリ
    - FILE 型を経由してデータをやり取りするための関数群
- ファイル処理をするためには
  - FILE 型と I/O ライブラリを学習し、使いこなす必要がある

ファイルの利用(サンプル 1)

- □ファイル型の利用 (サンプル 1)
  - ○入出力を行うプログラムを変更してみる
    - ▹キーボードからの入力を画面に出力するプログラム[001]
    - ▹ファイルからの入力を画面に出力するプログラム[002]
    - ▹キーボードからの入力をファイルに出力するプログラム[003]
    - ▹ファイルからの入力を画面をファイルに出力するプログラム[004]
- □サンプルの確認方法
  - ○入力ファイルは予め、「メモ帳」で作成しておく
    - ▹in002.txt / in004.txt
    - ▹ファイルを作成せずに、sample-002/004 を実行するとエラーになる
  - ○出力ファイルの結果は、「メモ帳」で内容を確認
    - ▹out003.txt / out004.txt
    - ▹実行前に、この名前のファイルが存在しない事を確認
    - ▹実行後に、作成日時も確認

ファイルポインターを利用した入出力

- ファイルポインタの宣言
  - (FILE *) 型の変数(ファイルポインタ)の宣言する
- ファイルのオープン
  - ファイルとファイルポインターは fopen で開く
  - ファイル名と、I/O を指定
    - 入力の場合は "r" / 出力の場合は "w"
  - ファイルのオープンは失敗する事がある (NULL が返る)
    - オープンに失敗した場合は、基本的にエラーとして終了する
- ファイルを利用した文字(バイトデータ)の入出力
  - 入力には、fgetc を利用する (cf. getchar)
  - 出力には、fputc を利用する (cf. putchar)
- ファイルのクローズ
  - ファイルポインターは利用が終ったら fclose で閉じる
    - 実は、クローズでも失敗する事がある (が、今回は、省略)

標準入出力

- 標準入出力のファイルポインタ
  - 標準入力 : stdin 「標準」でキーボードに結び付けられている
  - 標準出力 : stdout 「標準」で画面に結び付けられている
  - 標準エラー出力 : stderr 「標準」で画面に結び付けられている
    - stdout と独立でありエラー出力用に準備されている
  - 標準入出力のファイルポインタは別の場所で open/close される
    - 自分で open/close する必要がないファイルポインター
- 標準入出力を利用したプログラム [005]
  - getchar() は、fgetc (stdin)
  - putchar(ch) は、fputc (ch, stdout)
- ファイルポインタを利用した抽象化 [006/007]
  - ファイルの I/O とキーボード/画面を統合できる

fprintf/fscanf

- fprintf
  - printf のストリーム対応版
    - printf (..) は、fprintf (stdout, ..) と同じ
- fscanf
  - scanf のストリーム対応版
    - scanf (..) は、fscanf (stdin, ..) と同じ
- プログラムのファイル対応[008/009]
  - printf/scanf を fprint/fscanf に変更すれば良い

バイナリモード/テキストモード

- □テキストモード
  - ○基本はテキスト型のファイルを処理するモード
    - ▹fgetc/fputc, fprintf/fscanf
  - ○ファイルとの I/O で、文字列との変換を行う
    - ▹プログラム内 : バイナリ形式
    - ▹ファイル内 : テキスト形式
  - ○変換を行うために効率が悪い
    - ▹テキストが可変長な事も問題
    - ▹先頭から読むしかない
- □バイナリモード
  - ○プログラム内のデータをそのままファイルで利用
    - ▹open 時に "b" を追加指定
    - ▹I/O には、fread/fwrite を利用する
  - ○データのサイズが固定長
    - ▹ランダムアクセスが可能となる
    - ▹fseek で、直接アクセスができる

課題

□課題は、次の Web Page の内容を参照してください。

http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino

おわり

終了