§7.0-7.6: Applications (strings)

•devo

19 Oct 2005
CMPT14x
Dr. Sean Ho
Trinity Western University

Reminders:

- ***journals in folder***
- ***quiz ch6 today***

Review of ch6

- Libraries
  - M2 Standard Libraries:
    - I/O
      - **STextIO: ReadRestLine vs ReadString; ReadToken**
      - **Channels and redirection**
    - Math
  - Library module vs. program module
  - DEFINITION vs. IMPLEMENTATION
  - Accessor (set/get) functions

Quiz ch6 (3 questions, 20 marks, 10 minutes)

- Name 3 out of the 4 Standard Library modules we know
- What are the two parts (files) of a library module?
- In your own words, describe the roles of those two parts
- A program needs to copy text one character at a time from one file into another file. Put the following building blocks in the correct order to do this:

ReadChar	OpenOutput	OpenInput
WriteChar	CloseOutput	CloseInput

Quiz ch6 answers (#1-2)

- Name 3 out of the 4 Standard Library modules we know
  - STextIO, SWholeIO, SRealIO, RealMath
  - Also SIOResult (but not RedirStdIO)
- What are the two parts (files) of a library module?
  - DEFINITION, IMPLEMENTATION

Quiz ch6 answers (#3-4)

- In your own words, describe the roles of those two parts
  - DEF: public interface of the library, declares what is available for import
  - IMP: actual code, bodies of procedures
- A program needs to copy text one character at a time from one file into another file.
  - OpenInput OpenOutput
  - ReadChar
  - WriteChar
  - CloseOutput CloseInput

What's on for today (7.0-7.6)

- Strings: manipulating text
  - Null-terminating strings
  - String concatenation and length functions
  - The Strings Standard Library module
    - Comparing strings
- Application: cryptography (substitution cipher)
  - DEFINITION module
  - IMPLEMENTATION using some library-internal helper functions

Applications: manipulating text

- Recall that strings in M2 are just ARRAYs of CHAR:

```
VAR myName : ARRAY [0..14] OF CHAR;
```

- But the array is not always completely filled:

```
myName := “AppleMan”;
WriteString (myName);
```

- How does WS know where the string ends?
- Strings are null-terminated:
  - Append CHR(0) (same as “”) to end
  - Anything past the termination char is ignored

A	p	p	l	e	M	a	n	Ø						

String concatenation

- ISO M2 overloads the "+" operator to concatenate string literals and string constants:

```
WriteString ("Hello " + "World!");
```

  - Doesn't work with string variables
  - Doesn't work with characters
- ISO M2 also provides a built-in LENGTH function:

```
myName := "AppleMan";
myNameLen := LENGTH (myName);     (* 8 *)
```

  - LENGTH counts chars up to the null-terminator, not the total length of the ARRAY
- How would you implement these yourself?

Standard Library module: Strings

```
DEFINITION MODULE Strings;
TYPE
    String1 = ARRAY [0..0] OF CHAR;            (* 1-char string *)
PROCEDURE Length (stringVal: ARRAY OF CHAR): CARDINAL;
    (* same as built-in function LENGTH *)
PROCEDURE Assign (src: ARRAY OF CHAR;
  VAR dst: ARRAY OF CHAR);
    (* copy from one string into another *)
PROCEDURE Concat (src1, src2: ARRAY OF CHAR;
  VAR dst: ARRAY OF CHAR);
    (* concatenate two src strings and put result in dst *)
END Strings.
```

String comparison

- M2 type rules prevent us from simply doing

```
IF (myName = yourName) OR (myName < yourName)
  ...
```

- The Strings library provides comparison functions:

```
TYPE
    CompareResults = (less, equal, greater);
```

  - Compare (s1, s2: ARRAY OF CHAR): CompareResults;
  - Equal (s1, s2: ARRAY OF CHAR): BOOLEAN;
  - How would you implement these?
- Strings has other handy procedures; go look
  - Search and replace, extract substring, etc.

Cryptography example

- Cæsar substitution cipher:
  - Key: e.g., QAZXSWEDCVFRTGBNHYUJMKIOLP
  - Cleartext: input text to encrypt
  - Ciphertext: output encrypted text
  - Encoding: replace each letter in source with corresponding letter from code key
  - Decoding: same, using the decode key
- ROT13 was an example of a substitution cipher
  - Key: NOPQRSTUVWXYZABCDEFGHIJKLM

Write a Substitution cipher library

- What public interface do we want for the library?

```
DEFINITION MODULE Substitution;
TYPE CodeString = ARRAY [0..25] OF CHAR;

PROCEDURE Encode (src: ARRAY OF CHAR;
  VAR dst: ARRAY OF CHAR; key: CodeString);

PROCEDURE Decode (src: ARRAY OF CHAR;
  VAR dst: ARRAY OF CHAR; key: CodeString);

END Substitution.
```

Implementing Substitution

- In the implementation it is handy to have some helper functions for internal use: these will not be exported:

```
IsLetter (ch: CHAR) : BOOLEAN;
    (* check if it's a letter or some other character *)
AlphaPos (ch: CHAR) : CARDINAL;
    (* index of a letter in the range 0..25 *)
DecodeKey (enckey: CodeString; deckey: CodeString);
    (* create a decode key from an encoding key *)
```

TODO items

- Homework due Fri: 6.11 #28
- Quiz ch7 on Fri
- Lab #6 next week: 7.14 #(22 / 32 / 37)
- Reading: through end of book for Fri
- 140 Final next week W-Th (two parts)