§8.5-8.12: Stream I/O

•devo

2 Nov 2005
CMPT14x
Dr. Sean Ho
Trinity Western University

Reminders:

- ***journals in folder***
- ***hw due today***

Generic swap

```
PROCEDURE Swap (VAR a, b : ARRAY OF LOC);        (* any type! *)
VAR
    temp : LOC;
    max, count : CARDINAL;
BEGIN
    IF CanSwap (a, b)
        THEN
            FOR count := 0 TO HIGH (a)    (* swap one LOC at a time *)
                DO
                    temp := a [count];
                    a [count] := b [count];
                    b [count] := temp;
                END;
        END;
END Swap;
```

What's on for today (8.6-8.12)

- Sequential streams: StreamFile driver
  - StringIO, WholeIO, RealIO libraries
- Rewindable streams: SeqFile driver
  - Reread and Rewrite
  - File modes: read/write/old
- Binary streams: RawIO driver
- Standard Channels (StdInChan, StdOutChan)

Restricted stream I/O

- The StreamFile library opens and closes sequential files
- The TextIO, RealIO, etc. libraries contain the same procedures as their S-equivalents, but
  - Each procedure has an extra param specifying the file channel to use:

```
VAR
    out: ChanId;                    (* file channel *)
    result : OpenResults;
StreamFile.Open (out, “output.txt”, write, result);
WholeIO.WriteCard (out, myCard, 0);
StreamFile.Close (out);
```

Standard input, standard output

- The standard input and output channels (usually keyboard and screen) are file channels:

```
WholeIO.ReadCard (StdChans.StdInChan(), myCard);
WholeIO.WriteCard (StdChans.StdOutChan(), myCard,
  0);
```

- With StreamFile, you can have multiple channels open at the same time
  - e.g., output to screen and file simultaneously
  - Now we can understand how RedirStdIO works

Rewindable sequential stream I/O

- The SeqFile library opens and closes rewindable sequential streams:
  - OpenRead, OpenWrite, OpenAppend
  - Reread (cid: ChanId): rewind to beginning
  - Rewrite (cid: ChanId): clear file and start over
- Open streams with a combination of modes:
  - Read, write, old
    - Old: ok to overwrite (clobber) existing files
- If opened read+write, use Reread/Rewrite to switch between reading and writing

Example: rewindable file

- Read from keyboard, store in file, read back:

```
FROM SeqFile IMPORT
    ChanId, OpenWrite, write, read, old, Close, Reread,
      OpenResults;
FROM WholeIO IMPORT
    ReadCard, WriteCard;
FROM StdChans IMPORT
    StdInChan, StdOutChan;

VAR
    file : ChanId;
    result : OpenResults;
```

Example: rewindable file, cont.

```
BEGIN
    OpenWrite (file, “output.txt”, read+write+old, result);
    IF result = opened
        THEN
            WriteString (StdOutChan(), “Type a number: ”);
            WriteLn (StdOutChan());
            ReadCard (StdOutChan(), myCard);
            WriteCard (file, myCard, 0);

            Reread (file);          (* rewind file and start reading *)
            ReadCard (file, myCard);
        END;
    Close (file);
```

Binary I/O

- The analogue to StreamFile/SeqFile for binary streams is RawIO
  - Buffer is used for temporary storage of stream data in-transit over a channel
  - Read (cid, VAR to: ARRAY OF LOC);
  - Write (cid, from: ARRAY OF LOC);
    - Reads/writes binary data to/from buffer
    - cid is channel id (program file): IOChan.ChanID
    - Get result of read/write with ReadResults (cid):
      - **IOConsts.ReadResults: allRight, wrongFormat, endOfInput**

Review of today (8.6-8.12)

- Sequential streams: StreamFile driver
  - StringIO, WholeIO, RealIO libraries
- Rewindable streams: SeqFile driver
  - Reread and Rewrite
  - File modes: read/write/old
- Binary streams: RawIO driver
- Standard Channels (StdInChan, StdOutChan)
- Low-level device-independent I/O: IOChan
  - (just be aware that StreamFile/SeqFile/etc. use IOChan for even lower-level stuff)

TODO items

- Quiz ch8 on Fri
- Lab 7 due next week: 8.13 #(53 / 60 / 62)
- Reading: through §9.6 for Fri