

Introduktion til Programmering og Problemløsning (PoP)

Input/output

Jon Sparring
Department of Computer Science
2020/11/22

UNIVERSITY OF COPENHAGEN



Input og Output

- **Konsol input og output:**
 - Udskrivning af text.
 - Input fra konsollen.
- **Skrivning og læsning af filer.**
 - Åbning og lukning af filer.
 - Læsning.
 - Skrivning.
- **Læsning fra internettet.**
 - HTTP requests.

Konsollen

Når en process startes i konsollen, etableres (åbnes) tre strømme som konteksten kan benytte til at kommunikere med processen (programmet):

| Navn | Navn på strøm | Bibliotek |
|-----------------|---------------|----------------------|
| Standard output | stdout | System.Console |
| Standard input | stdin | System.Console |
| Standard error | stderr | System.Console.Error |

Vi har tidligere set hvordan printf (og venner) kan benyttes til at udskrive på stdout.

Funktioner i System.Console

- Udskrivning på stdout:
`val Write : string -> unit // Write to the console`
`val WriteLine : string -> unit // Write a line`
- Indlæsning fra stdin:
`val Read : unit -> int // Read ascii character`
`val ReadLine : unit -> string // Read a line`

Bemærk: Funktionerne Read og ReadLine "blokerer" indtil data er tilgængelig.

Eksempel: Temperatur omregner (temp.fsx)

```
open System
let fahrenheit (c:float) : unit =
    if c < -273.15 then failwith "input too small"
    else printfn "Fahrenheit: %f" (9.0/5.0*c + 32.0)

Console.Write "Temperature in degrees Celcius: "
let s = Console.ReadLine()
try fahrenheit(float(s)) with
| Failure s -> Console.Error.WriteLine s
| _ -> Console.Error.WriteLine "Expecting number"
```

Kommandolinieargumenter

Kommandolinieargumenter kan læses ved at etablere en hovedfunktion af typen

`string array -> int`

og tilføje en

`EntryPoint`

attribute til funktionen:

```
[<EntryPoint>]
```

```
let main (args: string array) : int =  
    for a in args do printfn "%s" a  
    0 // status code "ok"
```

```
$ fsharpc --nologo main.fsx
```

```
$ mono main.exe hi there
```

```
hi
```

```
there
```

Læsning af filer

Operationer til læsning af UTF-8 filer
er tilgængelige i modulet

System.IO.File:

Eksempel: linier i en fil (lines.fs)

```
type StreamReader = System.IO.StreamReader
  EndOfStream : bool
  Close : unit -> unit
  ReadToEnd : unit -> string // incl. newlines
  ReadLine : unit -> string // excl. newlines
  Read : unit -> int
val OpenText : string -> StreamReader
```

```
[<EntryPoint>]
let main (args:string array) : int =
  let rec loop n (r:System.IO.StreamReader) =
    if r.EndOfStream then
      n
    else
      ignore(r.ReadLine())
      loop (n+1) r
  if Array.length args > 0 then
    let fp = System.IO.File.OpenText args.[0]
    printfn "%d" (loop 0 fp)
    0
  else
    printfn "Expects file name as argument"
    1
```

Skrivning af filer

Operationer til skrivning af UTF-8 filer er tilgængelige i modulet `System.IO.File`:

```
type StreamWriter = System.IO.StreamWriter
  Close : unit -> unit
  WriteLine : string -> unit // add newline
  Write : string -> unit // no newline
val CreateText : string -> StreamWriter
val AppendText : string -> StreamWriter
```

Eksempel: linier i en fil (lines.fs)

```
let rec fib n =
  if n <= 2 then
    1
  else
    fib(n-1)+fib(n-2)
let rec loop n i (w:System.IO.StreamWriter) =
  if i > n then
    w.Close()
  else
    w.WriteLine(string(fib i))
    loop n (i+1) w
let fp = System.IO.File.CreateText "out.txt";
let n = loop 10 1 fp
```

Læsning fra internettet

Det er muligt at foretage HTTP requests fra F# programmer til web-servere på internettet.

```
open System.Net
open System
open System.IO

let fetchUrl (url:string) : string =
    let req = WebRequest.Create(Uri(url))
    use resp = req.GetResponse()
    use stream = resp.GetResponseStream()
    use reader = new IO.StreamReader(stream)
    reader.ReadToEnd()

let str = fetchUrl "http://www.valutakurser.dk"
printfn "%s" str
```

Bemærk:

1. Den specielle F# **use**-binding sikrer at resourcer der er knyttet til "disposable" objekter automatisk bliver frigivet når scope forlades.
2. https protokollen kræver gyldige certifikater, og det er vanskeligt at sætte op i F#, så hold jer til http-sider.

Resumé

I denne video hørte du om:

- Konsol input og output
- Skrivning til og læsning af filer
- Læsning af internettet