42 - Fichiers textes

Programmation Concurrente - LI330
Université P. & M. Curie - année scolaire 2013/2014

PrC

La notion de «fichiers standards»

- Certains environnements définissent des «fichiers standards»
  - Entrée standard : fichier en lecture seulement
  - Sortie standard : fichier en écriture seulement
  - Ces «fichiers standards» sont toujours ouverts

- Dans un langage de programmation, ces fichiers sont très souvent considérés comme étant «par défaut»
  - Si on ne précise pas de paramètre aux primitives d'entrées/sorties, les opérations correspondantes s'effectuent sur ces fichiers
  - On peut parfois réaffecter les fichiers par défaut
    - Ada: fichiers d'entrée et de sortie standard
    - C: fichiers d'entrée, de sortie et d'erreur standard (Unix)

Ada.Text_Io (primitives de gestion)


```
procedure Create (File : in out File_Type;
                  Mode : in File_Mode := Out_File;
                  Name : in String := "";
                  Form : in String := "");
procedure Open (File : in out File_Type;
                Mode : in File_Mode;
                Name : in String;
                Form : in String := "");
procedure Close (File : in out File_Type);
procedure Delete (File : in out File_Type);
procedure Reset (File : in out File_Type;
                 Mode : in File_Mode);
procedure Reset (File : in out File_Type);
function Mode (File : in File_Type) return File_Mode;
function Name (File : in File_Type) return String;
function Is_Open (File : in File_Type) return Boolean;
function End_Of_File (File : in File_Type) return Boole
function Form (File : in File_Type) return String;
```

"" => fichier temporaire

Doit intégrer les contraintes de l'OS

Types déclarés:
Count
Positive_Count

Ada.Text_Io (primitives de lecture/écriture - 1)


```
-- Sauts de lignes en lecture ou en écriture
procedure New_Line (File    : in File_Type;
                    Spacing : in Positive_Count := 1);
procedure New_Line (Spacing : in Positive_Count := 1);
procedure Skip_Line (File    : in File_Type;
                     Spacing : in Positive_Count := 1);
procedure Skip_Line (Spacing : in Positive_Count := 1);
-- Lecture
procedure Get (File : in File_Type;
               Item : out Character);
procedure Get (Item : out Character);
procedure Get (File : in File_Type;
               Item : out String);
procedure Get (Item : out String);
```

Ada.Text_Io (primitives de lecture/écriture - 2)


```
-- Lecture d'une ligne
procedure Get_Line (File : in File_Type;
                    Item : out String;
                    last : out Natural);

procedure Get_Line (Item : out String;
                    Last : out Natural);

-- Écriture
procedure Put (File : in File_Type;
               Item : in Character);

procedure Put (Item : in Character);

procedure Put (File : in File_Type;
               Item : in String);

procedure Put (Item : in String);

-- Écriture d'une ligne
procedure Put_Line (File : in File_Type;
                    Item : in String);

procedure Put_Line (Item : in String);
```

Ada.Text_Io (manipulation des fichiers standards)


```
-- contrôle des descripteurs de fichiers d'entrée et de sortie standards
procedure Set_Input (File : in File_Type);
procedure Set_Output(File : in File_Type);
procedure Set_Error (File : in File_Type);

function Standard_Input  return File_Type;
function Standard_Output return File_Type;
function Standard_Error  return File_Type;

function Current_Input   return File_Type;
function Current_Output  return File_Type;
function Current_Error   return File_Type;
```

Ada.Text_Io (autres fonctions)

- Paquetages inclus
  - Integer_Io
  - Float_Io
  - Fixed_Io
  - Enumeration_Io
  - Etc...
- Trouver la spécification complète de Text_Io
  - http://www.adahome.com/rm95/rm9x-A-10-01.html