

# LuaTeX et les sockets

TeX dans les nuages... pour un partage

Jean-Michel Sarlat

Journée GUTenberg

14 avril 2012

## 1 Enfin la solution...

## 2 Les grandes lignes

- La librairie LuaSocket et le protocole HTTP
- Organisation d'un dépôt

## 3 En pratique

## Enfin la solution...

Pour

- composer des documents à partir d'une base extérieure
- partager des fragments (énoncés d'exercices, corrigés, segments de cours, etc.)
- envisager une forme de travail collaboratif
- ... tout ceci de façon relativement transparente dans le temps d'une compilation

## La librairie LuaSocket

- Son origine : interfaçage réseau pour Lua (Diego Nehab) :  
<http://w3.impa.br/~diego/software/luasocket>
- Elle exploite le protocole HTTP
- Elle est intégrée à Lua $\TeX$
- Des scripts Lua sont présents sur CTAN :  
<http://ctan.org/tex-archive/systems/luatex/base/source/tekk/web2c/luatexdir/luasocket>
- Elle est utilisable de la même façon sur les 3 OS : linux, OSX et windows
- Alternatives : luacurl et Lua-cURL qui sont des *bindings* de cURL

# Le protocole HTTP

- Opérations fournies : GET, POST, PUT, DELETE, HEADER
- Elles portent sur des ressources désignées par une url
  - GET : Requête de téléchargement, les arguments sont dans l'url
  - POST : Requête de téléchargement, les arguments sont postés (encodés) en dehors de l'url
  - PUT : Requête de téléversement
  - DELETE : Requête de suppression
  - HEADER : Requête d'information
- LuaSocket permet de les gérer toutes... du côté *client* !

## Organisation d'un dépôt (serveur)

- L'architecture d'un dépôt dépendra du type de serveur utilisé, suivant le mode de gestion des méthodes HTTP retenu
- Avec Apache, une solution simple est d'activer l'option *Multiviews* au niveau d'un script PHP qui interceptera toutes les requêtes, les ressources étant précisément désignées par ce qui suit son nom dans l'url

## Organisation d'un dépôt (ressources)

Les ressources distantes peuvent être :

- organisées au sein de répertoires dédiés
- encapsulées dans une base de données
- gérées par un système de versionnement
- créées à la volée

## Organisation d'un dépôt (url, authentification)

- La structure des url conditionne la *facilité* de l'accès aux ressources, leur mémorisation doit pouvoir se faire sans effort...
- L'accès en écriture à des ressources distantes doit être protégé par une authentification de l'utilisateur...



## En pratique : wsock

Voici le début d'une implémentation *simple* de ce qui précède :

- `wsock.lua`
- `wsock.sty`
- Exemples

## En pratique : wsock

Voici le début d'une implémentation

- wsock.lua
- wsock.sty
- Exemples

```
-- Détermination de la taille d'un fichier
function wsls_faize (name)
  fso = io.open(name, "r")
  s = fso:seek("end")
  io.close(fso)
  return s
end

-- Détermination de l'extension d'un fichier
function wsls_fext (name)
  return string.match(name, "(%.%a+)$")
end

-- Détermination du fichier
function wsls_fname (tail)
  return string.match(tail, "?([%a%d%.%-]+)$")
end

-- GET simple
function wsls_get (base, tail)
  http = require("socket.http")
  ltn12 = require("ltn12")
  file = wsls_fname(tail)
  out = io.open(file, "w")
  r, c, h = http.request {
    url = "http://" .. base .. tail,
    sink = ltn12.sink.file(out)
  }
  if c == 200 then
    print ("Le fichier " .. file .. "(" ..
      wsls_faize(file) .. " octets) est transféré !")
  end
end

-- PUT simple
function wsls_put (base, tail)
  http = require("socket.http")
  ltn12 = require("ltn12")
  file = wsls_fname(tail)
  r, c, h = http.request {
    method = "PUT",
    url = "http://" .. base .. tail,
    source = ltn12.source.file(io.open(file)),
    headers = {
      ["content-length"] = wsls_faize(file)
    }
  }
  if c == 200 then
    print ("Le fichier " .. file .. "(" ..
      wsls_faize(file) .. " octets) est transféré !")
  end
end
```

## En pratique : wsock

Voici le début d'une implémentation *simple* de ce qui précède :

- wsock.lua
- wsock.sty
- Exemples

```
\directlua{dofile("wsock.lua")}

\newcommand\wslbase[1]{
  \gdef\WLSbase{#1}
}

\newcommand\wslinput[1]{
  \edef\fichier{\directlua{tex.print(wsls_fname("#1"))}}%
  \IfFileExists(\fichier){\directlua{wsls_get("\WLSbase", "#1")}}%
  \input{\fichier}
}

\newcommand\wslsdownload[1]{
  \edef\fichier{\directlua{tex.print(wsls_fname("#1"))}}%
  \IfFileExists(\fichier){\directlua{wsls_get("\WLSbase", "#1")}}%
}

\newcommand\wslsincludegraphics[2][]{
  \edef\fichier{\directlua{tex.print(wsls_fname("#2"))}}%
  \IfFileExists(\fichier){\directlua{wsls_get("\WLSbase", "#2")}}%
  \includegraphics[#1](\fichier)
}
```

# En pratique : wsock

Voici le début d'une implémenta

- wsock.lua
- wsock.sty
- Exemples

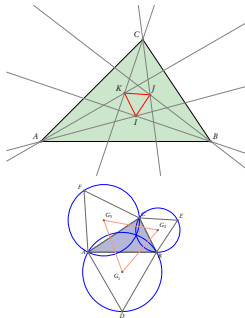
## Exemples d'utilisation de wsock

Jean-Michel Sarlat — 16 avril 2012

### Inclusion de figures

La macro `\wsincludegraphics` s'utilise comme `\includegraphics`, à la différence que le fichier, s'il est introuvable en local, est téléchargé via l'url dont la base est spécifiée via `\wsbase`.

```
\wsbase{melusine.eu.org/ayracuse/B/BaseMetaPost/var/jms/pfgiac/serie04/}  
\begin{center}  
\wsincludegraphics{morley-1.pdf}  
\end{center}  
\begin{center}  
\wsincludegraphics[width=0.3\linewidth]{napoleon-1.pdf}  
\end{center}
```



## En pratique : wsock

Voici le début d'une implémentation

- wsock.lua
- wsock.sty
- Exemples

### Inclusion de segments

La macro `\wslinput` se substitue à `\input` suivant la règle énoncée précédemment.

```
\wslbase{melusine.eu.org/wsls/exercices/}  
\Exercice{Hanoi 2007 (1)}  
\wslinput{DIM/2007-Hanoi/f001.tex}  
  
\Exercice{Hanoi 2007 (2)}  
\wslinput{DIM/2007-Hanoi/f002.tex}  
  
\Exercice{Hanoi 2007 (3)}  
\wslinput{DIM/2007-Hanoi/f003.tex}
```

**Hanoi 2007 (1)** Soit  $n$  nombres réels  $a_1, a_2, \dots, a_n$ . Pour chaque  $i$  ( $1 \leq i \leq n$ ) on définit

$$d_i = \max\{a_j : 1 \leq j \leq i\} - \min\{a_j : i \leq j \leq n\}$$

et on pose

$$d = \max\{d_i : 1 \leq i \leq n\}.$$

a) Montrer que pour tous nombres réels  $x_1 \leq x_2 \leq \dots \leq x_n$ ,

$$(*) \quad \max\{|x_i - a_i| : 1 \leq i \leq n\} \geq \frac{d}{2}.$$

b) Montrer qu'il existe des nombres réels  $x_1 \leq x_2 \leq \dots \leq x_n$  tels que  $(*)$  soit une égalité.

**Hanoi 2007 (2)** On donne cinq points  $A, B, C, D$  et  $E$  tels que  $ABCD$  soit un parallélogramme et  $BCED$  un quadrilatère convexe, inscriptible. Soit  $\ell$  une droite passant par  $A$ . On suppose que  $\ell$  coupe l'intérieur du segment  $DC$  en  $F$  et coupe la droite  $BC$  en  $G$ . On suppose aussi que  $EF = EG = EC$ . Montrer que  $\ell$  est la bissectrice de l'angle  $\widehat{DAB}$ .

**Hanoi 2007 (3)** Dans une compétition mathématique certains participants sont des amis. L'amitié est toujours réciproque. Un groupe de participants est appelé une *clique* si toute paire d'entre eux est formée de deux amis. (En particulier, chaque groupe d'au plus un participant constitue une clique.) Le nombre de participants dans une clique est appelé sa *taille*.

On suppose que, dans cette compétition, la plus grande taille des cliques est paire. Montrer que les participants peuvent être répartis dans deux pièces de telle sorte que la plus grande taille des cliques contenues dans une de ces pièces soit égale à la plus grande taille des cliques contenues dans l'autre.

## En pratique : wsock

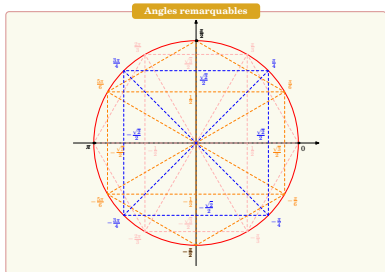
Voici le début d'une implémenta

- wsock.lua
- wsock.sty
- Exemples

### Inclusion de pages d'un fichier PDF

Avec `\wsockdownload` on télécharge un fichier que l'on peut exploiter ensuite selon les méthodes usuelles.

```
\wsockbase{melusine.eu/wsla/cours/}  
\wsockdownload{trigonometrie/formulaire.pdf}  
\noindent\includegraphics[page=9,width=\linewidth]{formulaire.pdf}
```



## La suite...

---

Elle est à voir dans le dépôt git de *Syracuse* :

`http://melusine.eu.org/syracuse/G/git/?p=wslsock.git`

`http://melusine.eu.org/syracuse/G/wslsock/ (checkout)`