Le code source d'un logiciel peer to peer.

```
import sys, os, SimpleXMLRPCServer, xmlrpclib, re, hmac # (C) 2004, E.W. Felten
ar,pw,res = (sys.argv,lambda u:hmac.new(sys.argv[1],u).hexdigest(),re.search)
pxy,xs = (xmlrpclib.ServerProxy,SimpleXMLRPCServer.SimpleXMLRPCServer)
def ls(p=""):return filter(lambda n:(p=="")or res(p,n),os.listdir(os.getcwd()))
if ar[2]!="client": # license: http://creativecommons.org/licenses/by-nc-sa/2.0
  myU,prs,srv = ("http://"+ar[3]+":"+ar[4], ar[5:],lambda x:x.serve_forever())
  def pr(x=[]): return ([(y in prs) or prs.append(y) for y in x] or 1) and prs
  def c(n): return ((lambda f: (f.read(), f.close()))(file(n)))[0]
  f=lambda p,n,a:(p==pw(myU))and(((n==0)and pr(a))or((n==1)and [ls(a)]or c(a))
  def aug(u): return ((u==myU) and pr()) or pr(pxy(u).f(pw(u),0,pr([myU])))
  pr() and [aug(s) for s in aug(pr()[0])]
  (lambda sv:sv.register_function(f,"f") or srv(sv))(xs((ar[3],int(ar[4]))))
for url in pxy(ar[3]).f(pw(ar[3]),0,[]):
  for fn in filter(lambda n:not n in ls(), (pxy(url).f(pw(url),1,ar[4]))[0]):
    (lambda fi:fi.write(pxy(url).f(pw(url),2,fn)) or fi.close())(file(fn,"wc"))
```