Upload Files to Servers. Case Study and Implementation

Lorentz JÄNTSCHI

Technical University Cluj-Napoca, RO, http://lori.academicdirect.ro

Abstract

In the field of client-server applications one of most complex application is to transfer binary files to a server. The problem of files transfer can be solves in many ways. Most complex one is to write a C/C++ script for transfer, and most simple, maybe, is using PHP language.

The paper presents main problems and a solution of client server software developing concerning the upload procedures and finally presents a full example of PHP set of programs that make storing and retrieval of binary files from a web server.

Keywords

Client-server application, File upload, Database binary field.

1. Introduction

In field of data processing it exist a large set of software but few are free. Even for free software, another problem it appears, operating system license and portability of the software. As example, to use well-known Microsoft Excel software, you must have Microsoft Windows and Microsoft Excel license and portability of Excel program is restricted to Windows platform. To import Excel files in another programs or platforms, conversion modules are needed, and conversion is not totally in most of the cases.

Platform independent and free software is a real alternative to this. First step to build free software is to install a free operating system. Unix-like operating systems are known to free, but even here exist licensed software. In order to select a free Unix-like operating system, best offer come from BSD family.

The most secure and license check for installed software is NetBSD. The NetBSD detect so called "license agreement" and do not permit to install a software if the software contain unacceptable license agreement (different form free) and software can be installed only if the acceptance is explicitly stipulated by the user in configuration files of the system. Another advantage of NetBSD operating system is his huge portability under various hardware platforms from i386 family to Sun and Macintosh machines.

By another hand, most full featured operating system of BSD family is FreeBSD. One of the advantages of the FreeBSD operating system is his software portability. With adequate packages, under FreeBSD, can be executed Hercules, CP/M, DOS, Windows, Linux and Sun-OS programs. Another advantage of FreeBSD system comes from easiest to install and uses.

Once you have an operating system installed, the next step is to choose a proper programming language for software developing.

Here, some major problems must be solved. In terms of programming, portability of resulted program can be a problem. As example, if we are chose to implement the algorithm in Visual Basic, the execution of the program is restricted to Windows machines. If Perl is our choice, a Unix-based machine is necessary to run program.

Even if we chouse to implement the program in C language, we will have serious difficulties to compile the programs on machines running with different operating systems.

The complexity of program building is also a serious reason in language chousing. Is known that C based languages is poor in simplicity and necessity more time to expend for application building than other languages.

Other questions require an answer: We want a server-based application or client based application? For a server-based application we want a server side application or a client side application?

As example, a client side application can have disadvantage of execution on client, and dependence of processing speed by calculation power of client machine. If we prefer this variant, a java script or visual basic script is proper to use.

A server side application requires a web server installed. The area of web servers is also a large set, but few have multiplatform capability. If we want a full-featured web server, Apache is our solution.

Under Apache, we have the possibility to execute programs already compiled in C, Fortran and Java, under Unix machines we can directly execute Perl programs, and, most important, under all operating system platforms we can execute PHP programs if we previously install PHP language and module binaries.

The advantage of PHP programs consist in his portability under most of operating system platforms and internal compilation feature that do not necessity the compilation “by hand” from the user.

The disadvantage can derive from the same internal compilation that consumes supplementary time in execution. But, this disadvantage can be partially eliminated through installing a PHP proxy that store compiled programs and next execution of the unmodified program use this compiled binary. In terms of program developing PHP is easy to use, the language borrow syntax from C, Pascal, Basic and Perl, but do not borrow the complex declaration syntax from them.

The pointer mechanism is absolute. Thus, a variable used as a string, can be exploited as an integer or real if the value represent a number such that. Class constructing is also available and PHP posses a strong library of database connectivity. Modular programming, recursivity and graphics are at home! Module loading of compiled programs in other languages and execution of binary programs is also available.

System services such as mail service are easy to exploit in PHP scripts. A very easy mechanism to link PHP scripts to HTML scripts make PHP language to be one of the best. Shell executing commands make PHP a useful platform for system administration (PhpSysInfo, WebAdmin, PhpMyAdmin, PhpPgAdmin). As a conclusion, PHP is our choice!

PHP programs are puts on a web server into a data folder and executed by them using PHP module (mod_phpX, X=3,4). The output of the PHP program is in HTML style and can be viewed by any web client (Microsoft Internet Explorer, Mozzilla, Opera, Netscape, Konkueror).

2. Interface

To submit binary files to a server, a typical interface is required. This interface links the client computer browser (like Internet Explorer) to the web daemon from server computer (like Apache). Depending on information type, communication between client and server are different. Thus, two methods of data transmission are well known, Post and Get methods. Get method transmit data directly through URL (universal resource locator) but is limited to ASCII characters and do not accept binary data. Post method open a direct connection to server of which type (called encryption type) depends on supplementary parameters specifies.

For binary data (as if pictures, archives, word or excel files) combined (or not) with regular ASCII data, always ENCTYPE='multipart/form-data', such that a simple example for a HTML (hypertext markup language) form for submitting binary data is:

```
<form ENCTYPE='multipart/form-data' method='post' action='up_file.php'>
File:<input type='file' name='file'>
<input type='image' src='Go.wmf' alt='Go!'>
</form>
```

where up_file.php is the script that preload binary file from web daemon and process them.

3. Data Storage

To save information on server, client must pass security checks. Security of web server depends on operating system platform and configuration files, web daemon type and configuration files and script grants.

Most of the operating systems securities settings are embed into a operating system database that manage all files from system, and all that we have to do is to specify the script grants. An example of script grants is:

```
-rwsr-sr-x  1 lori  lori  21258 Jun 12  2002 lori.cgi
```

There are 10 bits for file grants, first (0) are rezerved for type (- for files, d for directories), following three (1-3) for super user (root) grants (-/w: deny/allow rewriting of file; -/x/s: deny execution/executable/system file; -/r: deny/allow reading of file), 4-6 bits for user (lori) grants (in same meaning) and 7-9 bits for any other user (like www, mysql, anonymous) grants.

A grant of system file allows file (and implicitly user execution) to create other files and save onto server.

4. PHP File Management

PHP file management is quite different from any other file management system (maybe is very close to asp file management). The difference consists in inexistence of binary file (up_file.php is an ASCII file). Execution of an PHP are preceded by internal compilation of them.

As consequence, the "+s" attribute cannot be applied and a PHP program cannot create files/directories on server, but can modify them (files and directories contains) if these are writable by any user. In terms of security, this type of solution is very bad, because any hacker can exploit the PHP to create his own files on server or modify existent files.

Anyway, to storage files to server without any specialized management system is not an efficient technology. That is the reason of inexistence in PHP of like these instruments of file writing.

More, with some efforts, as example by writing a C/C++ script for file writing, and combining PHP script with C/C++ script, it is possible to make writing on server directly from a PHP program.

However, alternative is more elegant and efficient! Database connectivity of PHP with PostGresSQL or MySQL makes this procedure simple.

All that we have to do the user is to create a MySQL database, create a table into database, to specify the fields (one of them must be a binary field: BLOB).

If security reasons impose, a specific user can be defines into MySQL database management system with specific grants only for the newer created table.

The default location of php.ini file is /usr/local/etc/ directory of web server. Some important configuration options can be set in this file:

```
max_execution_time = 600; Maximum execution time of each script, in seconds
memory_limit = 64M; Maximum amount of memory a script may consume (8MB)
```

```
post_max_size = 16M; Maximum size of POST data that PHP will accept
file_uploads = On; Whether to allow HTTP file uploads
upload_max_filesize = 8M; Maximum allowed size for uploaded files
```

A simple PHP can write now the data into MySQL table:

```
<?
 $data=$_FILES['file']['name'];echo($data."<BR>");
 $data=$_FILES['file']['size'];echo($data."<BR>");
 $data=$_FILES['file']['type'];echo($data."<BR>");
 $data=$_FILES['file']['tmp_name'];echo($data."<BR>");
 $file = "";
 $fp = fopen($data, "rb"); while(!feof($fp)) $file .= fread($fp, 1024); fclose($fp);
 $file = addslashes($file); $file = addcslashes($file,"\0");
 include '../passwords.php';
 $c=mysql_connect($host,$mysql_user,$password);
 if(!$c) die("cannot make database connection");
 $t=mysql_select_db("binary_files"); if(!$t) die("cannot SELECT the database");
 $q="INSERT INTO `table1` (`file`) VALUES ('".$file."')";
 $r=mysql_query($q); if(!$r) die("cannot make INSERT into database");
 echo("<script>alert('Upload Ok.'); </script>");
?>
```

5. Data Retrieval

For select a binary data from table, a primary key onto an ASCII field is required. If nr is *primary key* and *auto_increment* is simple to select a record using this field values. The PHP program is:

```
<?
 include '../passwords.php';
 $c=mysql_connect($host,$user,$password); if(!$c) die("not dbase connection");
 $t=mysql_select_db("binary_files"); if(!$t) die("cannot SELECT the database");
 $q=sprintf("SELECT `file` FROM `table1` WHERE 1 AND `nr` LIKE");
 $q.=" '".$HTTP_POST_VARS['nr']."' LIMIT 0,300");
 $t=mysql_query($q); if(!$t) die("cannot SELECT file from the articles table");
 $r=mysql_fetch_array($t); if(!$r) die("no field in table");
 mysql_free_result($t);
 if (strpos($r["file"],"PDF")==1) header("Content-type: application/pdf\n");
   else header("Content-type: application/msword\n");
 header("Content-Transfer-Encoding: binary\n");
 header("Content-length: ".strlen($r["file"])."\n");
 print($r["file"]);
?>
```

6. Journals Database Structure

To implement a client-server application for two electronic journals (LEJPT and LJS) a database called journals was created using phpMyAdmin graphical interface (see figure 1).

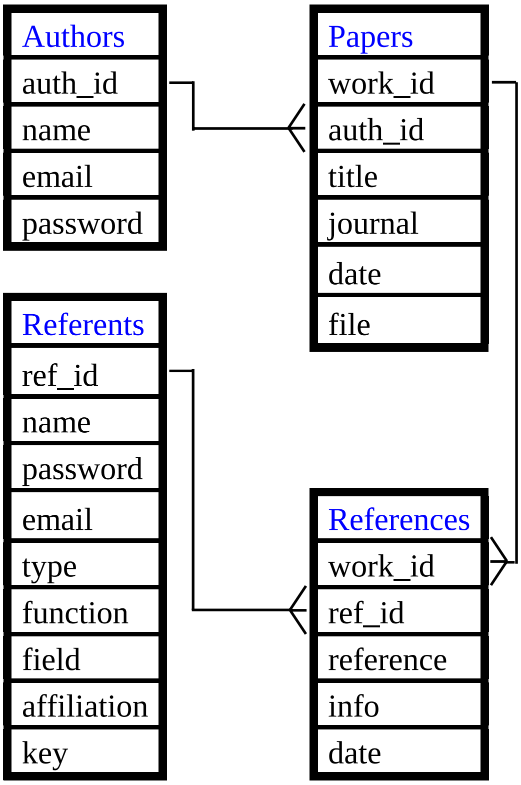


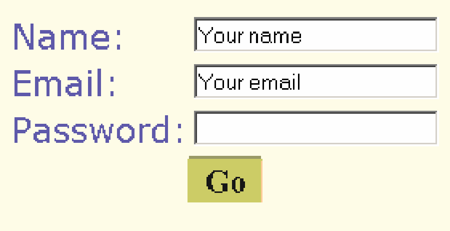
Fig. 1. Journals Database Structure

Four tables (authors, papers, referents, references) were created inside journals database. Table authors contain 4 fields related to corresponding author (name: name of the author, email: e-mail of the author, password: password of the author to enter in author's room, auth_id: a unique key generated at new author registration using the $_SERVER["UNIQUE_ID"] feature of apache web server). Table papers contain 5 fields related to the submitted paper (auth_id: author foreign key to identify the corresponding author, title: title of paper, journal: journal to which paper is submitted – LEJPT or LJS, date: a field which contain date and time of submission in format year:month:day:hour:minutes: seconds using date("y.m.d H:i:s", time()) PHP function, useful to sort papers by date and file: a binary field which contain the submitted material, in varied format – DOC, PDF or other type, in maximum size of about 16 Mb).

Table referents contain all information's about referents using 9 fields (ref_id: primary key generated by web server at creation time, name: first and second name of the referent, password: password of the referent to enter in referent's room, email: correspondence e-mail of the referent, type: type of the referent – two values are allowed here: referent or chief redactor, function: academic title of the referent, field: specialization or professional field of the referent, affiliation: institution where referent are affiliated and key: a numeric field used for setting the display order). The table references contain foreign keys from table referents (ref_id) and table papers (work_id), a field to store the date and time (date), a field to store the reference of paper identified by work_id and referent identified by ref_id (reference) and key: a numeric field for reference state of the paper: in processing (0), modifications required (1), rejected (2) or accepted (3).

7. Journals Management

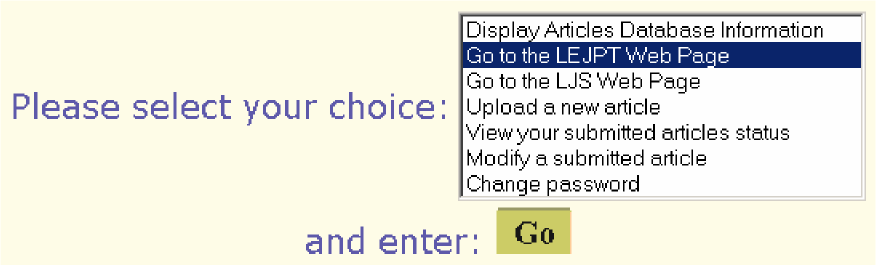
Thee category types of actions are available on database journals: author registration, author login, referent login. Two files contain general configurations for all PHP scripts: passwords.php (host, user and password for read/write in journals mysql database, warning, alert, email and script go back functions) and config.php (colors and styles). At author registration time, a post form is displayed as in figure 2.



Name: Your name
Email: Your email
Password:

Fig. 2. Author registration script

The corresponding author fills the text boxes with name, email and desired password and pressing the Go button the data are submitted to the web server and authors.php script write the information's into authors table of database journals and automatically assign the auth_id unique key. After registration, the author can login into author's room, using his passwords, where it has multiple choices as in figure 3.

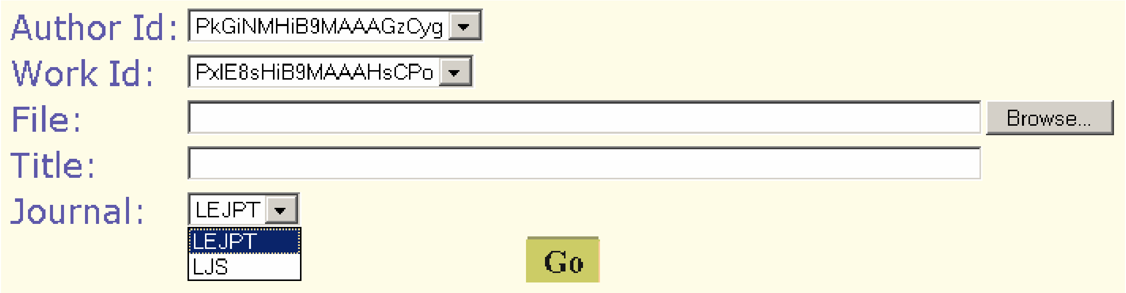


Please select your choice: Go to the LEJPT Web Page

Display Articles Database Information
Go to the LEJPT Web Page
Go to the LJS Web Page
Upload a new article
View your submitted articles status
Modify a submitted article
Change password
and enter:
Go

Fig. 3. Author's options

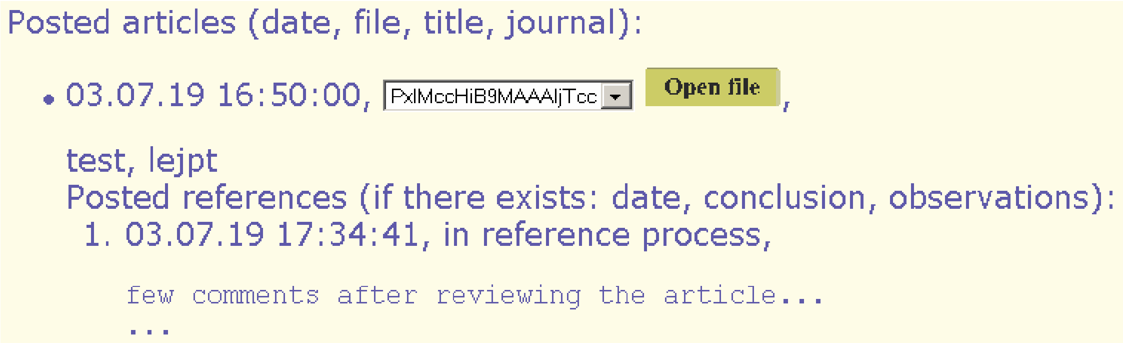
Article database information is a collection of free download articles from few most known publishing houses and until now contains about 15.000 articles free to display, download and print and is a feature of registered authors. Following two options goes to the journal main web pages. Upload a new article feature send a form for submitting an article as in figure 4.



Author Id: PkGiNMHiB9MAAAGzCyg
Work Id: PxIE8sHiB9MAAAHsCPo
File:
Title:
Journal: LEJPT

Fig. 4. Submitting electronic papers

The up_a.php script connect to mysql server and make the uploading of file to the web server and put the electronic paper into binary field of journals mysql database. Automatically referents are informed by email about new article arrival, article characterized by work_id primary key. View your submitted articles status option send a web page which contain all articles submitted by author with title, date of submission, reference status and referents opinions (if available). Name of referents are hidden and protected. The author has possibility to view the submitted article and read the references of submitted articles (see figure 5).



Posted articles (date, file, title, journal):

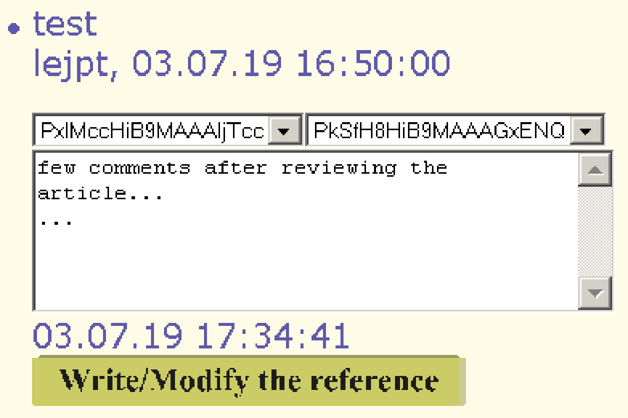
- 03.07.19 16:50:00, PxIMccHiB9MAAAljTcc ▾ Open file ,

  test, lejpt
  Posted references (if there exists: date, conclusion, observations):
  1. 03.07.19 17:34:41, in reference process,

     few comments after reviewing the article...
     ...

Fig. 5. Author feature of open submitted article and to read the references

Modify a submitted article feature is useful when the paper need corrections. After the lecturing of referents opinion, the author has choice to modify (replace) the material with new one. Key of the article remains unchanged, but after replacement, the referents which previously make reference of the article are informed by email. Change password option allow author to change his password for author's room. The referents can login using his passwords into referent's room where posted articles are displayed using work_id and ref_id foreign keys as in figure 6.



- test
  lejpt, 03.07.19 16:50:00

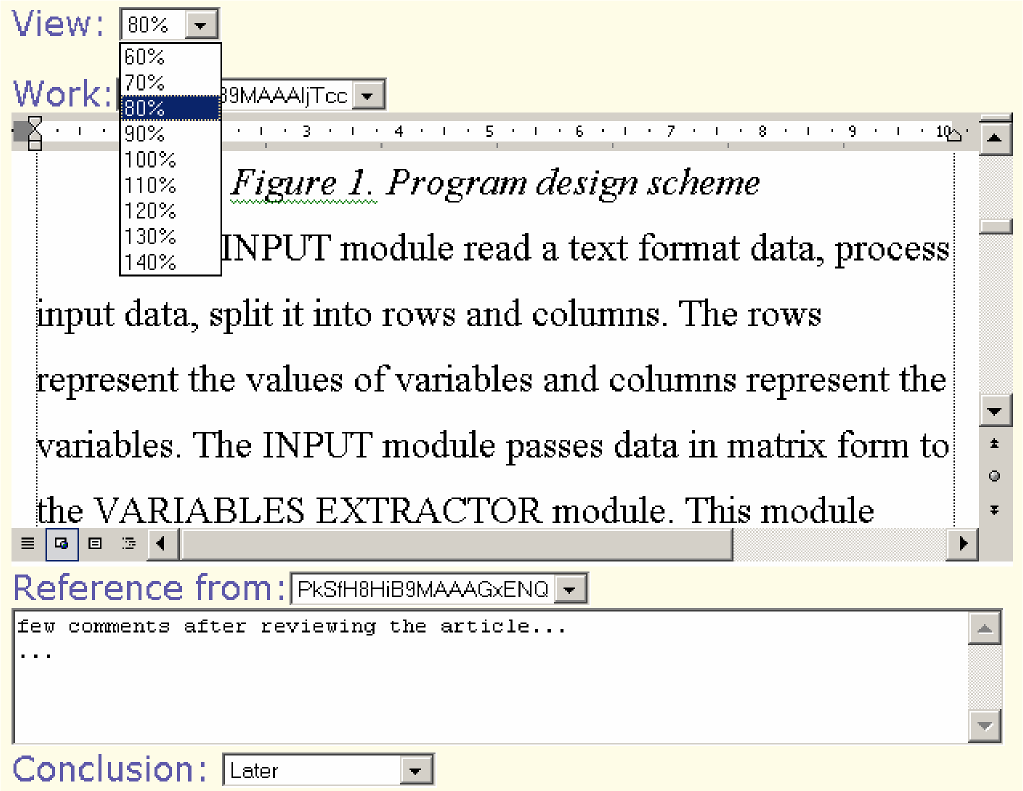
  PxIMccHiB9MAAAljTcc ▾ PkSfH8HiB9MAAAGxENQ ▾

  few comments after reviewing the
  article...
  ...

  03.07.19 17:34:41

Fig. 6. The "chamber" of referent room

Any time before acceptance, the referent can write or modify the reference (figure 6). To modify or write the reference, reference.php program assist the reference process and display the selected paper into a new window inside an IFRAME as in figure 7.



View: 80%

Work: 39MAAAljTcc

Figure 1. Program design scheme

INPUT module read a text format data, process input data, split it into rows and columns. The rows represent the values of variables and columns represent the variables. The INPUT module passes data in matrix form to the VARIABLES EXTRACTOR module. This module

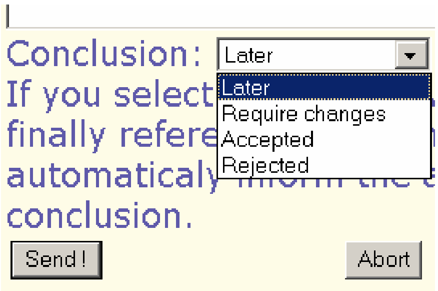
Reference from: PkSfH8HiB9MAAAGxENQ

```
few comments after reviewing the article...
...
```

Conclusion: Later

Fig. 7. Making of the reference

After the acceptance of the article (see figure 8) from one referent, the article are automatically hidden for all referents in future logins.



Conclusion: Later

If you select
finally refere
automaticaly
conclusion.

Fig. 8. Submission of the reference

The implemented system allows both authors and referents to interact via a client-server mechanism. More, the chief redactor it can see the results and make all required procedures for accepted papers.

8. Conclusions

A simplest way to make client-server applications using PHP language was describes. Binary data storage and data retrieval was solves using MySQL database server daemon. The download PHP script sends the header with content-type information to the web client (in present case Adobe Acrobat PDF file type is default one and Microsoft Word DOC file type is alternative one) and web client creates process and execute the specific program for editing and/or viewing the downloaded binary file. Using of the PHP features for management of electronic journals make the editorial job easy and fast.

References

1. http://www.zend.com
2. http://www.phpbuilder.com
3. http://www.mysql.com
4. http://www.FreeBSD.org
5. http://sourceforge.net/projects/phptriad
6. http://www.adobe.com
7. http://www.microsoft.com