#contentWrapper #fs, #sidebarContent #fs, #contentWrapper div [id * = 'myExtraContent'], #sidebarContent div [id * = 'myExtraContent'] {display: block;}

The Kermith workshop (https://translate.googleusercontent.com/translate_c?depth=1&hl=en&prev=search&rurl=translate.google.com&sl=fr&sp=nmt4&u=http://www.sugarbu

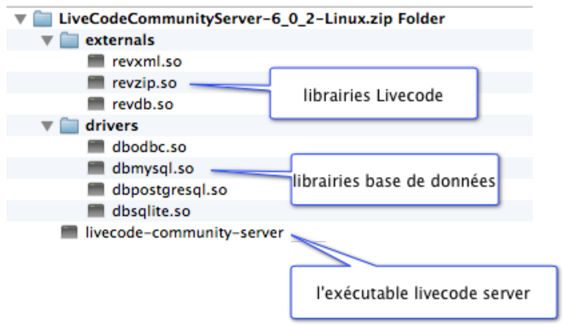
The other way to see supervision ...

LiveCode Server on Linux

The Community version of LiveCode brings a lot of new things as the days go by. For a few days you will find the Community server version on the LiveCode website http://downloads.livecode.com/livecode/server/6_0_2/ (https://translate.googleusercontent.com/translate_c?depth=1&hl=en&prev=search&rurl=translate.google.com&sl=fr&sp=nmt4&u=http://downloads.livecode.com/livecode/server/6_0_2/&xid=25657,15700022,157001
. Using this tool will allow you to use the LiveCode language on a web server in cgi mode. It is even possible to use the code located in the stacks of your programs. On the other hand, the graphic interface of the batteries will not be taken into account.
I propose in this article, the installation of a LiveCode server in a Debian server with Apache.

Installing the LiveCode Server Module

Get the zip file LiveCodeCommunityServer-6_0_2-Linux.zip. Unzip this file, you will have the following tree:


LiveCodeCommunityServer-6_0_2-Linux.zip Folder
externals
revxml.so
revzip.so
revdb.so
librairies Livecode
drivers
dbodbc.so
dbmysql.so
dbpostgresql.so
dbsqlite.so
librairies base de données
livecode-community-server
l'exécutable livecode server

Livecode Server Tree

I chose to copy all this tree in the folder / usr / local / livecodeserver.

Apache Prerequisites

We need to modify the Apache configuration and check if the modules mod_cgi, mod_actions and mod_alias are loaded. Let's load the Apache modules if it's not already done.

```
 a2enmod actions
a2enmod cgi
a2enmod aka
```

Then, change the configuration of Apache. For my example, I preferred the solution to dedicate a folder / usr / local / livecode for the Web server. Let's create the folder.

```
 mkdir / usr / local / livecode
```

Let's add the file livecode.conf in the folder /etc/apache2/conf.d

```
 Alias / livecode / usr / local / livecode /
<Directory "/ usr / local / livecode">
    MultiViews Indexes Options
    AllowOverride None
    Order allow, deny
    Allow from all
    AddHandler livecode-script .lc
    Livecode-script action / livecode-cgi / livecode-server
</ Directory>

<Directory "/ usr / local / livecodeserver">
     ExecCGI Options
     Order allow, deny
     Allow from all
</ Directory>

ScriptAlias / livecode-cgi / livecode-server / usr / local / livecodeserver / livecode-community-server
```

Restart the apache server to take into account this new configuration.

```
service apache2 restart
```

Server test

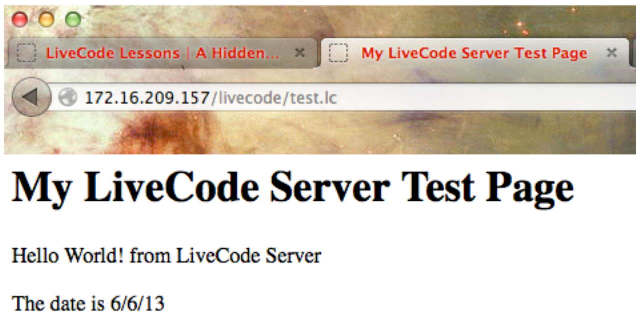
We will create a test page named test.lc in the folder / usr / local / livecode. <? lc?> tags are used to insert LiveCode commands. We will use the put command to send the line in the browser and the date function indicating the current date. These commands are exactly the same as the controls in the Desktop version, which is the strength of this programming environment.

```
 <Html>

<Head>
     <title> My LiveCode Server Test Page
</ Head>

<Body>
     <h1> My LiveCode Server Test Page
<? Lc
     put "<p> Hello World! from LiveCode Server </ p>"
     put "<p> The date is" && the date & "</ p>"
?>
</ Body>

</ Html>
```


LiveCode Lessons | A Hidden...
My LiveCode Server Test Page
172.16.209.157/livecode/test.lc

My LiveCode Server Test Page

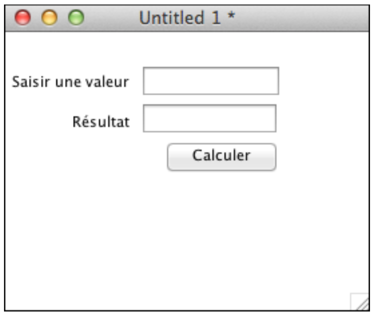
Hello World! from LiveCode Server

The date is 6/6/13

Our test page in Firefox

Using a stack (stack)

In this chapter, we will discover how to use a stack developed for a heavy client type application and transform it into a thin client type web application. Let's start by creating our heavy client. We will use a stack and a standard card, an input area, a label for the result and a button to execute the action. This program will have the complex task of adding 20 to the number entered! The important thing is to show how you can reuse the code for a Web application.



The "heavy client" application

The simple code of the application. The following code will be placed in the card handler.

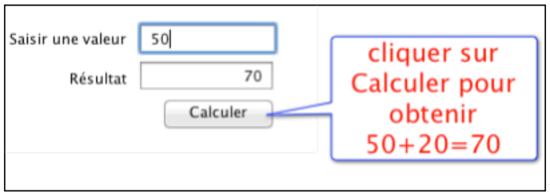
function calculation parameter
local tvaleur
put parameter into tvaleur
add 20 to tvaleur
return tvaleur

end calcul

This code will be placed in the button handler.

on mouseUp
put calcul (the field "TxtValue") into field "Result"
end mouseUp

Back up the stack with the name test_stack.livecode



How the application works

Now we will put our stack into operation in our web server. Copy the livecode file to the / usr / local / livecode folder. We will be able to reuse the code contained in the handler of the card. The graphical interface of the map will be replaced by an HTML form. To use the code of the stack, we will use the following command:

```
start using stack "test_stack.livecode"
```

Let's create the html file capture_stack.lc

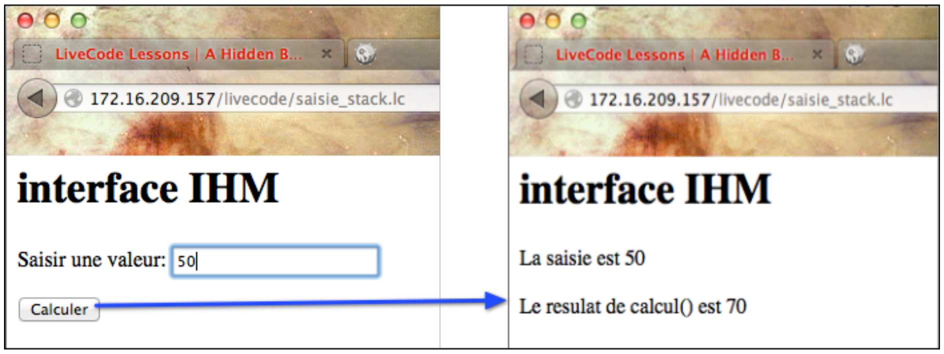
```
 <Html>

<Head>
     <title> Stack Utilization Project </ title>
</ Head>

<Body>
<h1> HMI interface </ h1>
<? Lc
    if $ _POST ["form_submitted"] is true then
        put "<p> The entry is" && $ _POST ["value"] & "</ p>"
          if $ _POST ["value"] is not "" then
               start using stack "test_stack.livecode"
               put "<p> The calculation result () is" && calculation ($ _ POST ["value"]) & "</ p>"
          else
               put "<p> You forgot to enter a value </ p>"
          end if
    else
    ?>
                <form action = "./ entry_stack.lc" method = "POST">
        <p> Enter a value: <input type = "text" name = "value" value = "" /> </ p>
        <input type = "hidden" name = "form_submitted" value = "true" />
        <p> <input type = "submit" value = "Calculate" /> </ p>
                </ Form>
        <? Lc
    end if
?>
</ Body>

</ Html>
```

The page uses the calculation function of the test_stack.livecode stack. A big advantage of this solution is that you can reuse your code for Desktop, Mobile or Server applications.


LiveCode Lessons | A Hidden B...
172.16.209.157/livecode/saisie_stack.lc
interface IHM
Saisir une valeur: 50
Calculer
LiveCode Lessons | A Hidden B...
172.16.209.157/livecode/saisie_stack.lc
interface IHM
La saisie est 50
Le resulat de calcul() est 70

How the web page works

Next article, the use of a database. At your keyboard.
comments powered by Disqus (https://translate.googleusercontent.com/translate_c?depth=1&hl=en&prev=search&rurl=translate.google.com&sl=fr&sp=nmt4&u=http://disqus.com/&xid=25657,15700022,15700124,15700149,15700186,15700191,1