Full Stack Development with REXX

Using REXX as a programming language for end-to-end application development.

Powered by debian GNU/Linux

Powered by httpd APACHE

REXX

PostgreSQL

Full Stack Development with REXX

From the earliest days of the World Wide Web, REXX has been used as an application language for CGI, The Common Gateway Interface. CGI is an early standard for writing dynamic web content.

Powered by debian GNU/Linux

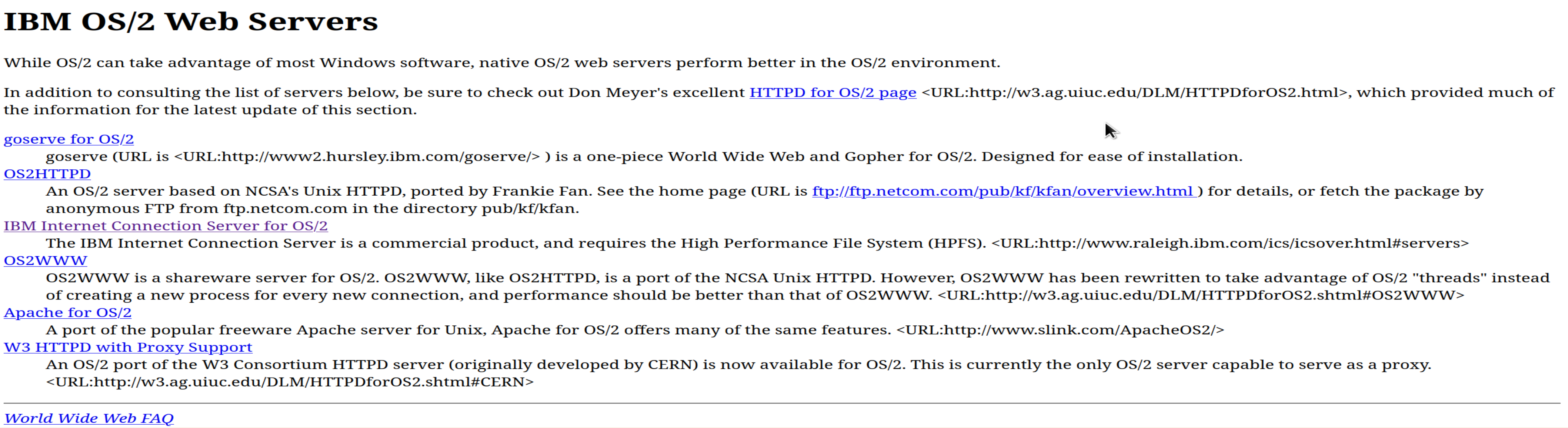
Powered by httpd APACHE

REXX

PostgreSQL

Full Stack Development with REXX

IBM was very early to provide web servers and related applications. REXX support was included in many of these early products, particularly, on OS/2



IBM OS/2 Web Servers

While OS/2 can take advantage of most Windows software, native OS/2 web servers perform better in the OS/2 environment.

In addition to consulting the list of servers below, be sure to check out Don Meyer's excellent HTTPD for OS/2 page <URL:http://w3.ag.uiuc.edu/DLM/HTTPDforOS2.html>, which provided much of the information for the latest update of this section.

goserve for OS/2
: goserve (URL is <URL:http://www2.hursley.ibm.com/goserve/>) is a one-piece World Wide Web and Gopher for OS/2. Designed for ease of installation.

OS2HTTPD
: An OS/2 server based on NCSA's Unix HTTPD, ported by Frankie Fan. See the home page (URL is ftp://ftp.netcom.com/pub/kf/kfan/overview.html) for details, or fetch the package by anonymous FTP from ftp.netcom.com in the directory pub/kf/kfan.

IBM Internet Connection Server for OS/2
: The IBM Internet Connection Server is a commercial product, and requires the High Performance File System (HPFS). <URL:http://www.raleigh.ibm.com/ics/icsover.html#servers>

OS2WWW
: OS2WWW is a shareware server for OS/2. OS2WWW, like OS2HTTPD, is a port of the NCSA Unix HTTPD. However, OS2WWW has been rewritten to take advantage of OS/2 "threads" instead of creating a new process for every new connection, and performance should be better than that of OS2WWW. <URL:http://w3.ag.uiuc.edu/DLM/HTTPDforOS2.shtml#OS2WWW>

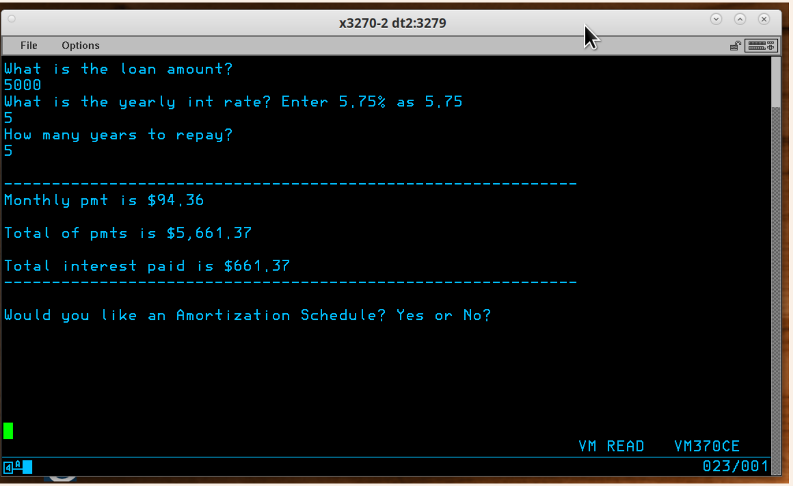
Apache for OS/2
: A port of the popular freeware Apache server for Unix, Apache for OS/2 offers many of the same features. <URL:http://www.slink.com/ApacheOS2/>

W3 HTTPD with Proxy Support
: An OS/2 port of the W3 Consortium HTTPD server (originally developed by CERN) is now available for OS/2. This is currently the only OS/2 server capable to serve as a proxy. <URL:http://w3.ag.uiuc.edu/DLM/HTTPDforOS2.shtml#CERN>

World Wide Web FAQ

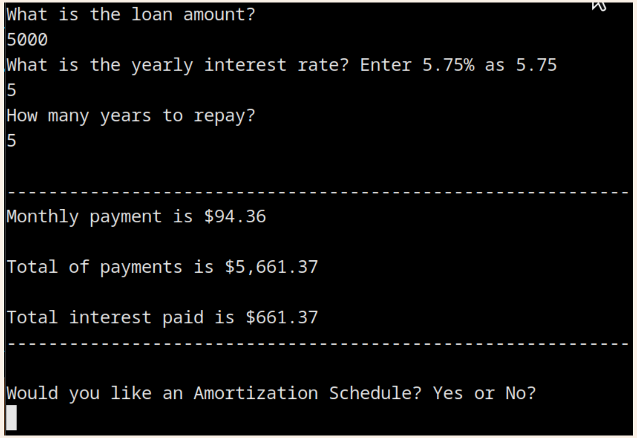
Full Stack Development with REXX

End user interfaces of the early 90s



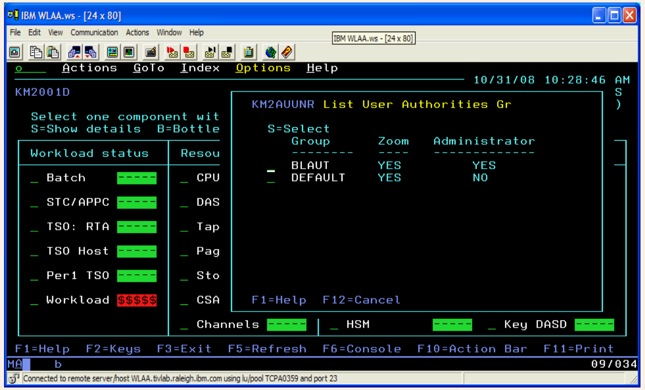
```
What is the loan amount?
5000
What is the yearly int rate? Enter 5.75% as 5.75
5
How many years to repay?
5

------------------------------------------------------------------
Monthly pmt is $94.36

Total of pmts is $5,661.37

Total interest paid is $661.37
------------------------------------------------------------------

Would you like an Amortization Schedule? Yes or No?



                                                  VM READ    VM370CE
                                                             023/001
```

VM/CMS



```
What is the loan amount?
5000
What is the yearly interest rate? Enter 5.75% as 5.75
5
How many years to repay?
5

-----------------------------------------------------------------------

Monthly payment is $94.36

Total of payments is $5,661.37

Total interest paid is $661.37
-----------------------------------------------------------------------

Would you like an Amortization Schedule? Yes or No?
```

DOS, OS/2, Unix



```
 o___  Actions  GoTo  Index  Options  Help
 ---------------------------------------------------  10/31/08 10:28:46 AM
 KM2001D                 KM2AUUNR List User Authorities Gr                S
                                                                          )
   Select one component wit
   S=Show details  B=Bottle   S=Select
                                Group      Zoom    Administrator
  Workload status    Resou      --------   ----    -------------
                              _ BLAUT      YES          YES
   _ Batch     -----   _ CPU  _ DEFAULT    YES          NO
   _ STC/APPC  -----   _ DAS
   _ TSO: RTA  -----   _ Tap
   _ TSO Host  -----   _ Pag
   _ Per1 TSO  -----   _ Sto
   _ Workload  $$$$$   _ CSA  F1=Help  F12=Cancel
                       _ Channels -----  | _ HSM        -----  _ Key DASD -----
 F1=Help  F2=Keys  F3=Exit  F5=Refresh  F6=Console  F10=Action Bar  F11=Print
MA    b                                                                  09/034
```

Connected to remote server/host WLAA.tivlab.raleigh.ibm.com using lu/pool TCPA0359 and port 23

IBM CUA (Common User Access). A standard for creating graphical type applications in a 3270 terminal.

Powered by debian GNU/Linux

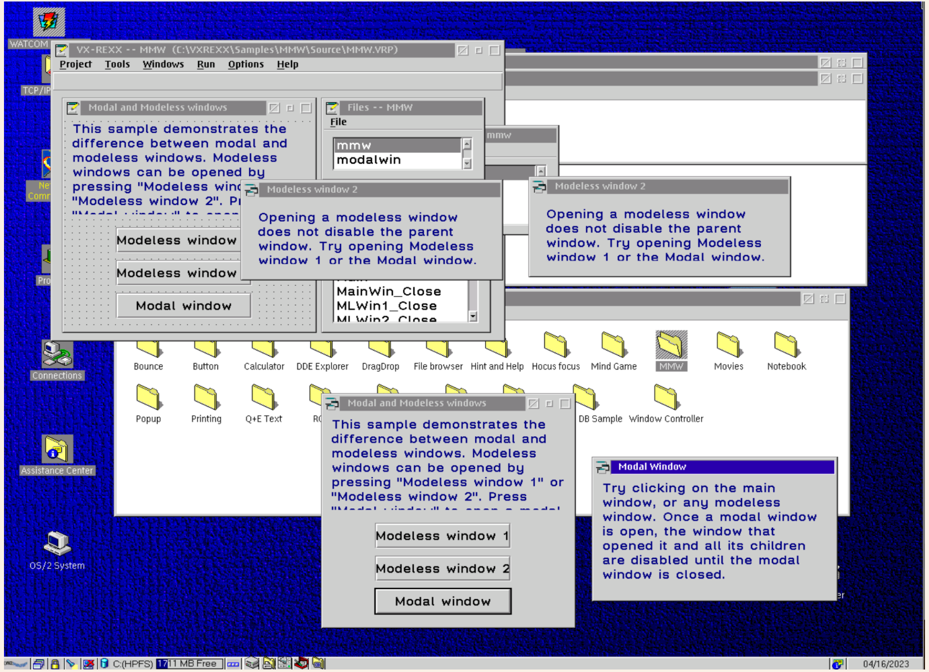
Powered by httpd APACHE

REXX

PostgreSQL

Full Stack Development with REXX

Mid 90s GUI development on OS/2 using VX-REXX. OS/2 provided some powerful REXX development tools.



This sample demonstrates the difference between modal and modeless windows. Modeless windows can be opened by pressing "Modeless window 1" or "Modeless window 2". Press

Opening a modeless window does not disable the parent window. Try opening Modeless window 1 or the Modal window.

Try clicking on the main window, or any modeless window. Once a modal window is open, the window that opened it and all its children are disabled until the modal window is closed.

In the 1990s, OS/2 was a REXX developer's dream. In my humble opinion, it was (and possibly still is), the richest, most tightly integrated REXX platform to date.

Powered by debian GNU/Linux

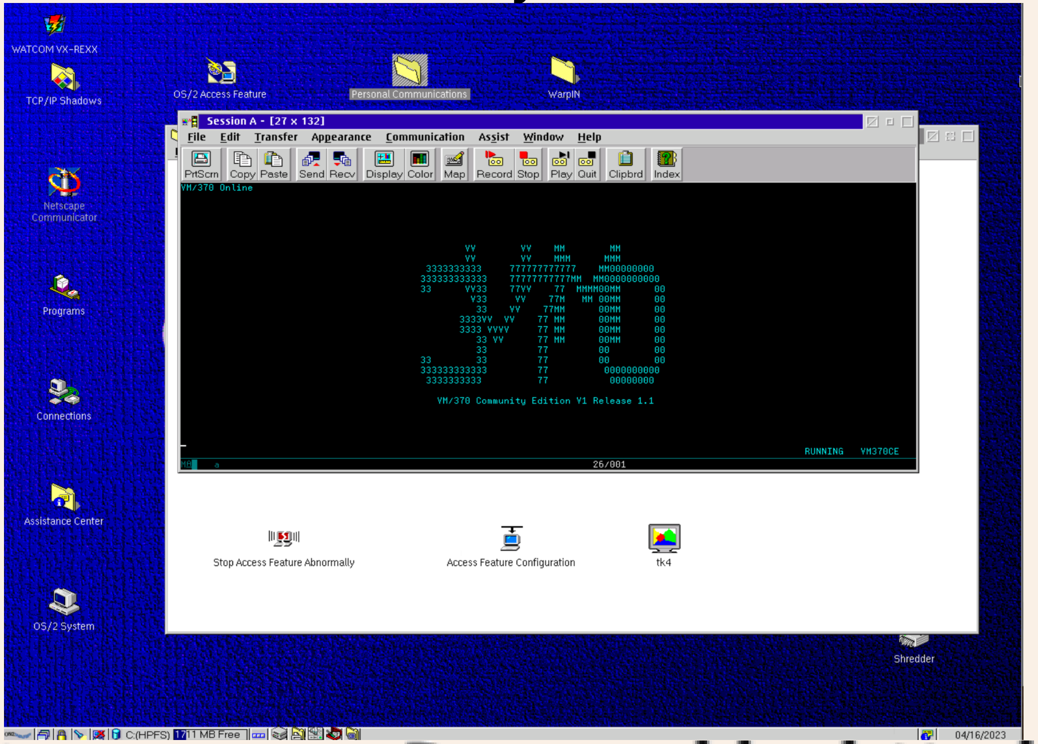
Powered by httpd APACHE

REXX

PostgreSQL

Full Stack Development with REXX

OS/2 Goes the way of the DODO, or so we thought.



```
VM/370 Online

        VV        VV    MM        MM
        VV        VV    MMM      MMM
 3333333333     777777777777    MM00000000
333333333333    77777777777MM  MM0000000000
33      VV33    77VV    77  MMMM00MM      00
         V33     VV     77M   MM 00MM      00
          33    VV    77MM      00MM      00
       3333VV  VV    77 MM      00MM      00
       3333 VVVV     77 MM      00MM      00
          33 VV      77 MM      00MM      00
          33         77         00        00
33        33         77         00        00
333333333333         77          0000000000
 3333333333          77           00000000

   VM/370 Community Edition V1 Release 1.1

                                                    RUNNING   VM370CE
                                   26/001
```

In the late 1990’s, the writing was on the wall for the demise of OS/2 and all of the Amazing REXX tools that could be used to develop not only applications, but rich user interfaces.

Powered by debian GNU/Linux

Powered by httpd APACHE

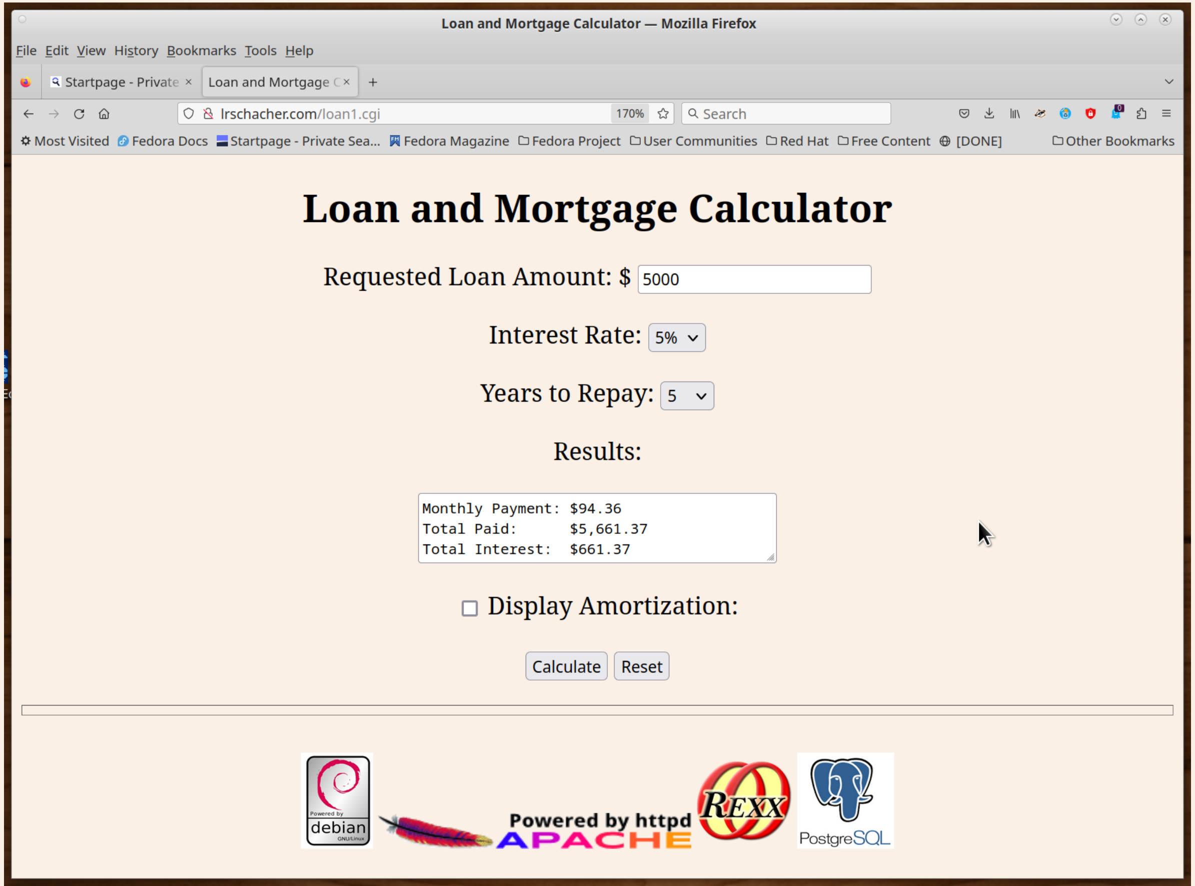
REXX

PostgreSQL

Full Stack Development with REXX

The Web Browser is the new terminal!

As previously stated, from the earliest days of the World Wide Web, REXX could be used for developing CGI applications.



Loan and Mortgage Calculator

Requested Loan Amount: $ 5000

Interest Rate: 5%

Years to Repay: 5

Results:

```
Monthly Payment:  $94.36
Total Paid:       $5,661.37
Total Interest:   $661.37
```

☐ Display Amortization:

Calculate Reset

The World Wide Web replaced the previous days of Floppy Disc and Sneaker Net. REXX CGI applications could often provide a reasonable replacement for simple desktop applications.

This application, is the same back end code as shown on slide 4 that ran on VM/CMS, DOS, OS/2, Windows, and Unix in terminals.

Full Stack Development with REXX

The LAMP Stack

From Wikipedia, the free encyclopedia

LAMP (Linux, Apache, MySQL, PHP/Perl/Python) is an acronym denoting one of the most common software stacks for many of the web's most popular applications. However, LAMP now refers to a generic software stack model and its components are largely interchangeable.

Each letter in the acronym stands for one of its four open-source building blocks:

- Linux for the operating system
- Apache HTTP Server
- MySQL for the relational database management system
- PHP, Perl, or Python programming language

Powered by debian GNU/Linux

Powered by httpd APACHE

REXX

PostgreSQL

Full Stack Development with REXX

Introducing the DARP stack!

- Debian GNU / Linux
- Apache HTTP Server
- REXX (ooRexx 5.0)
- PostgreSQL Database Management Server

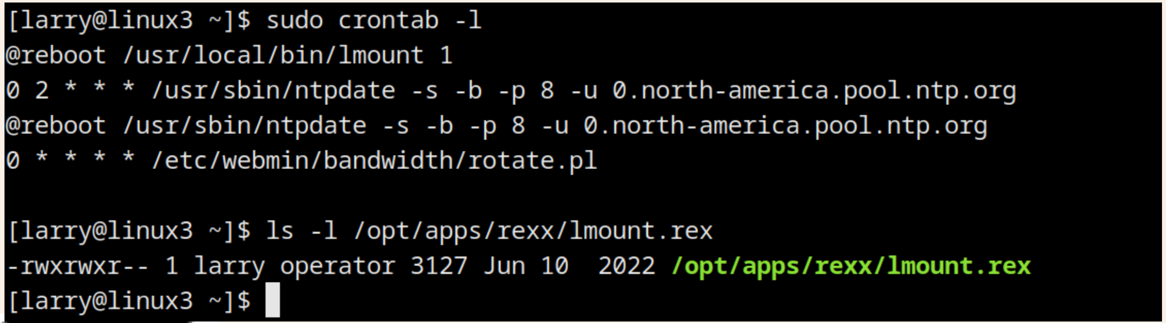
Full Stack Development with REXX

REXX and the operating system.

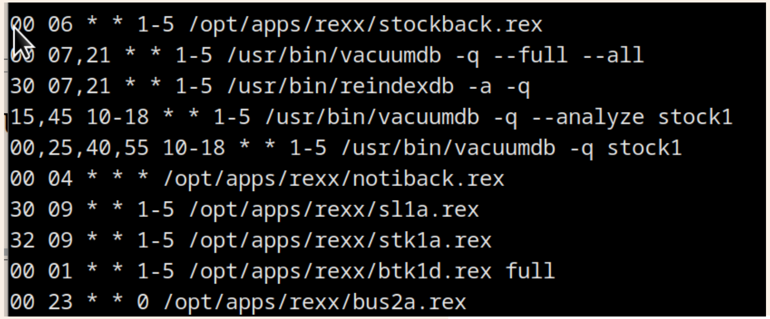
From its earliest inception, the REXX programming language has been tightly integrated with the operating system. This remains the same and so much more with modern REXX interpreters such as Regina and ooRexx on modern systems such as Windows and Linux.

In the case of the latest Debian GNU / Linux distribution, just about anything you can do in a bash script, perl program etc, You can do in REXX.

REXX is useful for both System Administration and user space tasks.



```
[larry@linux3 ~]$ sudo crontab -l
@reboot /usr/local/bin/lmount 1
0 2 * * * /usr/sbin/ntpdate -s -b -p 8 -u 0.north-america.pool.ntp.org
@reboot /usr/sbin/ntpdate -s -b -p 8 -u 0.north-america.pool.ntp.org
0 * * * * /etc/webmin/bandwidth/rotate.pl

[larry@linux3 ~]$ ls -l /opt/apps/rexx/lmount.rex
-rwxrwxr-- 1 larry operator 3127 Jun 10  2022 /opt/apps/rexx/lmount.rex
[larry@linux3 ~]$
```



```
00 06 * * 1-5 /opt/apps/rexx/stockback.rex
00 07,21 * * 1-5 /usr/bin/vacuumdb -q --full --all
30 07,21 * * 1-5 /usr/bin/reindexdb -a -q
15,45 10-18 * * 1-5 /usr/bin/vacuumdb -q --analyze stock1
00,25,40,55 10-18 * * 1-5 /usr/bin/vacuumdb -q stock1
00 04 * * * /opt/apps/rexx/notiback.rex
30 09 * * 1-5 /opt/apps/rexx/sl1a.rex
32 09 * * 1-5 /opt/apps/rexx/stk1a.rex
00 01 * * 1-5 /opt/apps/rexx/btk1d.rex full
00 23 * * 0 /opt/apps/rexx/bus2a.rex
```

Powered by debian GNU/Linux

Powered by httpd APACHE

REXX

PostgreSQL

Full Stack Development with REXX

REXX and the operating system.

```
#!/usr/local/bin/rexx
/* Sync local files from server */
path_string = "/usr/kerberos/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/opt/apps/rexx:/opt/apps/cgi-rexx"
pvalue = VALUE("PATH",path_string,"ENVIRONMENT")
rs.0 = 6
rs.1 = "/lhome/larry/docs/ /home/larry/docs/"
rs.2 = "/media2/ /media1/"
rs.3 = "/opt/lapps/ /opt/apps/"
rs.4 = "/opt/ldata/ /opt/data/"
rs.5 = "/netpub/ /pub/"
rs.6 = "/l1backup1/linux1/ /backup1/linux1/"
IF netmount() = 0 THEN
DO i = 1 to rs.0
   SAY "rsync" rs.i
   "sudo rsync -grupPElvot --delete" rs.i
END /* do i */
RETURN rc
```

Powered by debian GNU/Linux

Powered by httpd APACHE

REXX

PostgreSQL

Full Stack Development with REXX

REXX and the Apache HTTP server

From the earliest days of the World Wide Web, REXX has been used to parse and process CGI (Common Gateway Interface) and to create dynamic web content.

There is also an Apache Module called mod_rexx, which allows REXX to control all aspects of Apache responses and also provides dynamic content called Rexx Server Pages (RSP)

Many of the old CGI tools and libraries still work under the latest version of Linux Apache, or can be slightly modified to run just fine.

I recently installed the latest mod_rexx under Debian, and It tested just fine.

I still use an old CGIPARSE.CMD from OS/2 on my modern Debian GNU / Linux Apache server as well.

Powered by debian GNU/Linux

Powered by httpd APACHE

REXX

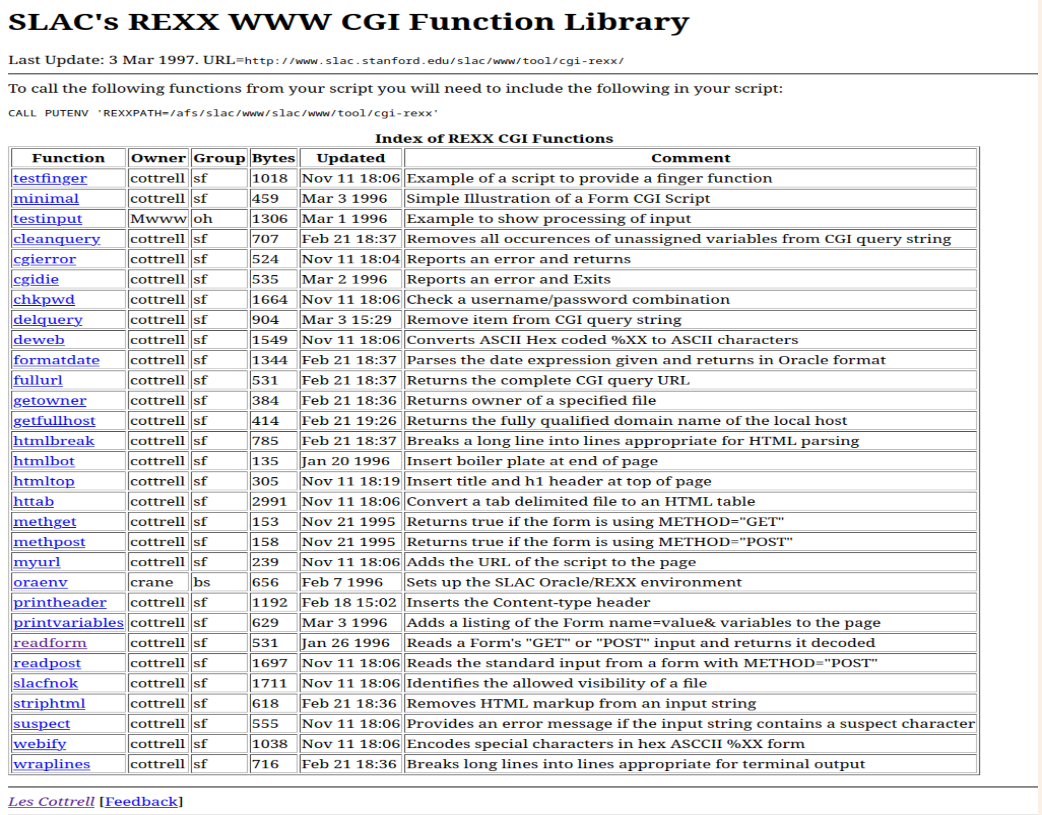
PostgreSQL

Full Stack Development with REXX

REXX and the Apache HTTP server

```
/* Do not modify below this line --  Important parsing code...
   From Frankie Fan's OS2HTTPD archive
*/
ParseQueryString: procedure expose Parms. NFields
  Parse arg P
  i = 1
  do while ((P \= '') & (i < 1000))
     Parse var P Parms.Text.i '&' rest
     Parse var Parms.Text.i Parms.Tag.i '=' Parms.KeyVal.i
     Parms.Tag.i = translate( Parms.Tag.i)
     Parms.XVal.i=DecodeKeyVal( Parms.KeyVal.i)
     P = rest
     i = i + 1
  end
  NFields = i - 1
  return NFields

DecodeKeyVal: procedure
  parse arg Code
  Text=''
  Code=translate(Code, ' ', '+')  /* Convert + signs to spaces */
  rest='%'
  do while (rest\='')
     Parse var Code T '%' rest
     Text=Text || T
     if (rest\='') then
      do
        ch = left( rest,2)
        c=X2C(ch)                /* Hex to character conversion */
        Text=Text || c
        Code=substr( rest, 3)
      end
  end
  return Text
```

Full Stack Development with REXX



SLAC's REXX WWW CGI Function Library

Last Update: 3 Mar 1997. URL=`http://www.slac.stanford.edu/slac/www/tool/cgi-rexx/`

To call the following functions from your script you will need to include the following in your script:

```
CALL PUTENV 'REXXPATH=/afs/slac/www/slac/www/tool/cgi-rexx'
```

Index of REXX CGI Functions

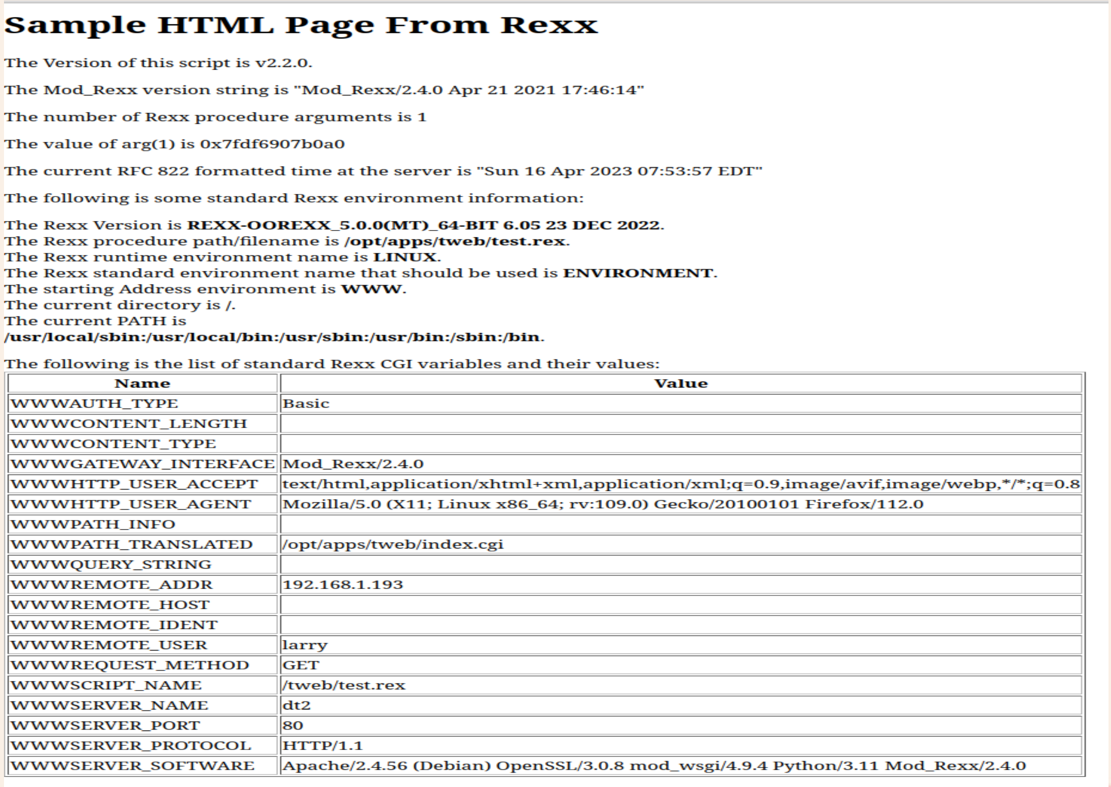
Function	Owner	Group	Bytes	Updated	Comment
testfinger	cottrell	sf	1018	Nov 11 18:06	Example of a script to provide a finger function
minimal	cottrell	sf	459	Mar 3 1996	Simple Illustration of a Form CGI Script
testinput	Mwww	oh	1306	Mar 1 1996	Example to show processing of input
cleanquery	cottrell	sf	707	Feb 21 18:37	Removes all occurences of unassigned variables from CGI query string
cgierror	cottrell	sf	524	Nov 11 18:04	Reports an error and returns
cgidie	cottrell	sf	535	Mar 2 1996	Reports an error and Exits
chkpwd	cottrell	sf	1664	Nov 11 18:06	Check a username/password combination
delquery	cottrell	sf	904	Mar 3 15:29	Remove item from CGI query string
deweb	cottrell	sf	1549	Nov 11 18:06	Converts ASCII Hex coded %XX to ASCII characters
formatdate	cottrell	sf	1344	Feb 21 18:37	Parses the date expression given and returns in Oracle format
fullurl	cottrell	sf	531	Feb 21 18:37	Returns the complete CGI query URL
getowner	cottrell	sf	384	Feb 21 18:36	Returns owner of a specified file
getfullhost	cottrell	sf	414	Feb 21 19:26	Returns the fully qualified domain name of the local host
htmlbreak	cottrell	sf	785	Feb 21 18:37	Breaks a long line into lines appropriate for HTML parsing
htmlbot	cottrell	sf	135	Jan 20 1996	Insert boiler plate at end of page
htmltop	cottrell	sf	305	Nov 11 18:19	Insert title and h1 header at top of page
httab	cottrell	sf	2991	Nov 11 18:06	Convert a tab delimited file to an HTML table
methget	cottrell	sf	153	Nov 21 1995	Returns true if the form is using METHOD="GET"
methpost	cottrell	sf	158	Nov 21 1995	Returns true if the form is using METHOD="POST"
myurl	cottrell	sf	239	Nov 11 18:06	Adds the URL of the script to the page
oraenv	crane	bs	656	Feb 7 1996	Sets up the SLAC Oracle/REXX environment
printheader	cottrell	sf	1192	Feb 18 15:02	Inserts the Content-type header
printvariables	cottrell	sf	629	Mar 3 1996	Adds a listing of the Form name=value& variables to the page
readform	cottrell	sf	531	Jan 26 1996	Reads a Form's "GET" or "POST" input and returns it decoded
readpost	cottrell	sf	1697	Nov 11 18:06	Reads the standard input from a form with METHOD="POST"
slacfnok	cottrell	sf	1711	Nov 11 18:06	Identifies the allowed visibility of a file
striphtml	cottrell	sf	618	Feb 21 18:36	Removes HTML markup from an input string
suspect	cottrell	sf	555	Nov 11 18:06	Provides an error message if the input string contains a suspect character
webify	cottrell	sf	1038	Nov 11 18:06	Encodes special characters in hex ASCCII %XX form
wraplines	cottrell	sf	716	Feb 21 18:36	Breaks long lines into lines appropriate for terminal output

Les Cottrell [Feedback]

Much of Les Cotrell’s CGI Function library also runs just fine with ooRexx 5 on Apache 2.4 with only minor modifications

https://www.slac.stanford.edu/slac/www/tool/cgi-rexx/

Full Stack Development with REXX



Sample HTML Page From Rexx

The Version of this script is v2.2.0.

The Mod_Rexx version string is "Mod_Rexx/2.4.0 Apr 21 2021 17:46:14"

The number of Rexx procedure arguments is 1

The value of arg(1) is 0x7fdf6907b0a0

The current RFC 822 formatted time at the server is "Sun 16 Apr 2023 07:53:57 EDT"

The following is some standard Rexx environment information:

The Rexx Version is **REXX-OOREXX_5.0.0(MT)_64-BIT 6.05 23 DEC 2022**.
The Rexx procedure path/filename is **/opt/apps/tweb/test.rex**.
The Rexx runtime environment name is **LINUX**.
The Rexx standard environment name that should be used is **ENVIRONMENT**.
The starting Address environment is **WWW**.
The current directory is /.
The current PATH is **/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin**.

The following is the list of standard Rexx CGI variables and their values:

Name	Value
WWWAUTH_TYPE	Basic
WWWCONTENT_LENGTH	
WWWCONTENT_TYPE	
WWWGATEWAY_INTERFACE	Mod_Rexx/2.4.0
WWWHTTP_USER_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
WWWHTTP_USER_AGENT	Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/112.0
WWWPATH_INFO	
WWWPATH_TRANSLATED	/opt/apps/tweb/index.cgi
WWWQUERY_STRING	
WWWREMOTE_ADDR	192.168.1.193
WWWREMOTE_HOST	
WWWREMOTE_IDENT	
WWWREMOTE_USER	larry
WWWREQUEST_METHOD	GET
WWWSCRIPT_NAME	/tweb/test.rex
WWWSERVER_NAME	dt2
WWWSERVER_PORT	80
WWWSERVER_PROTOCOL	HTTP/1.1
WWWSERVER_SOFTWARE	Apache/2.4.56 (Debian) OpenSSL/3.0.8 mod_wsgi/4.9.4 Python/3.11 Mod_Rexx/2.4.0

The latest version of mod_rexx also runs just fine with ooRexx 5 on Apache 2.4

https://sourceforge.net/projects/modrexx/

Full Stack Development with REXX

REXX and the PostgreSQL DBMS server

SQL and DBMS tools and libraries have been available for REXX on just about every platform from the earliest days of REXX.

I think you would call this an Easter Egg.
I wonder if Mark Hessling remembers that around 1999, I was having trouble compiling his REXX/SQL library on my boxed set, Red Hat Linux 6.2, server running in my basement. I created a telnet account for him (in the days before ssh) and he kindly compiled REXX/SQL for me. I used REXX/SQL for years! Cheers Mark!

Powered by debian GNU/Linux

Powered by httpd APACHE

REXX

PostgreSQL

Full Stack Development with REXX

REXX and the PostgreSQL DBMS server

```
#!/usr/local/bin/rexx
/*  replicate PostgreSQL database */
st = TIME()
PARSE ARG database shost thost bfile
IF database = "" THEN database = "notialerts"
IF bfile = "" THEN bfile = "/tmp/b1.backup"
IF shost = "" THEN shost = "debian2"
IF thost = "" THEN thost = "linux3"
rc = SysFileDelete(bfile)
SAY "backing up" database
cmd = "pg_dump -h" shost "-f" bfile "-Fc" database
SAY cmd
ADDRESS "bash" cmd
SAY "RC for backup was" rc
IF rc = 0 THEN
DO
   cmd = "pg_restore -h" thost "-c -d" database bfile
   SAY cmd
   ADDRESS "bash" cmd
   SAY "RC for restore was" rc
END /* do */
ELSE SAY "Backup failed."
rc = SysFileDelete(bfile)
SAY "RC for delete is" rc
SAY "Program complete" st TIME()
RETURN 0
```

Full Stack Development with REXX

Advanced CGI Technique

Most CGI programs use an html form and call a separate CGI program. This pair of processes necessitates that the user press back or otherwise take some action to return to the html form page in order to use different options. The samples in mod_rexx demonstrate an alternative technique of using recursion in a CGI. The CGI draws the page and calls itself on submit. The CGIPARSE routine is called at the top of the page to process any GET or POST parameter values and modifies the display based on user chosen values. This allows an application to be processed entirely by the server and can be used as an alternative to client side coding such as java-script

Full Stack Development with REXX

A real world application. A PUT options screener.

This application has several back end processes that run during U. S. Stock market hours, overnight, and weekly and loads market data into a PostgreSQL database. The web front end, does dynamic queries based on options chosen by the user. The application uses the recursive CGI technique mentioned above to keep the user in the same web page.

Stock Reports

Search for out of the money PUT options

Search parameters

Stock Symbol: □

Maximum Strike Price $25.00 Premium rate 0

Maximum Moneyness 75% Days to Expire 7

Sort by Stock, Expiration Date, Strike ● Lowest Strike Only ○ All Strikes

● Bid only ○ Any Volume ○ All available options

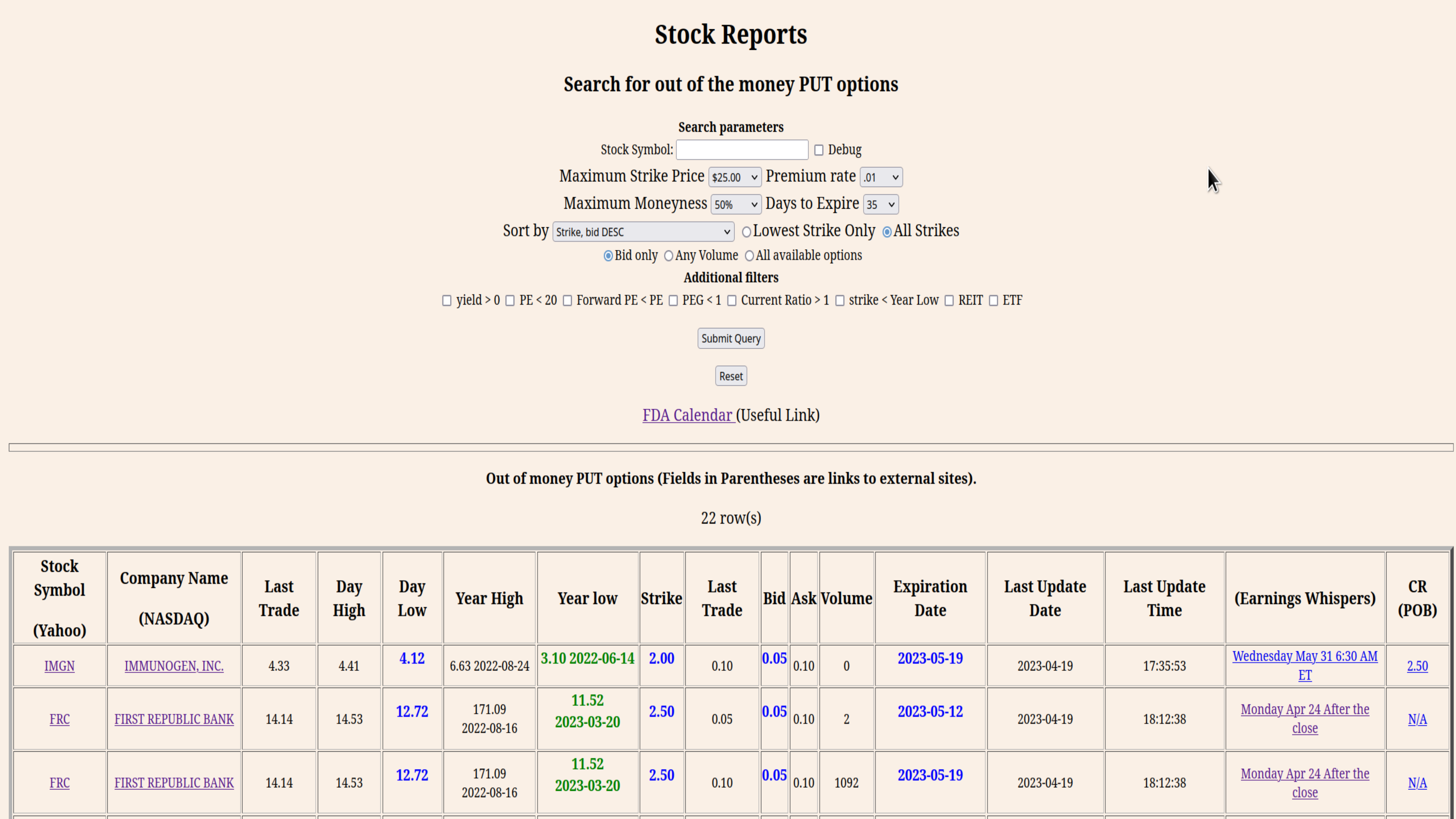
Additional filters

□ yield > 0 □ PE < 20 □ Forward PE < PE □ PEG < 1 □ Current Ratio > 1 □ strike < Year Low □ REIT □ ETF

Submit Query

Reset

FDA Calendar (Useful Link)



Stock Reports

Search for out of the money PUT options

Search parameters

Stock Symbol: [ ] □ Debug

Maximum Strike Price $25.00 Premium rate .01

Maximum Moneyness 50% Days to Expire 35

Sort by Strike, bid DESC ○ Lowest Strike Only ● All Strikes

● Bid only ○ Any Volume ○ All available options

Additional filters

□ yield > 0 □ PE < 20 □ Forward PE < PE □ PEG < 1 □ Current Ratio > 1 □ strike < Year Low □ REIT □ ETF

FDA Calendar (Useful Link)

Out of money PUT options (Fields in Parentheses are links to external sites).

22 row(s)

Stock Symbol (Yahoo)	Company Name (NASDAQ)	Last Trade	Day High	Day Low	Year High	Year low	Strike	Last Trade	Bid	Ask	Volume	Expiration Date	Last Update Date	Last Update Time	(Earnings Whispers)	CR (POB)
IMGN	IMMUNOGEN, INC.	4.33	4.41	4.12	6.63 2022-08-24	3.10 2022-06-14	2.00	0.10	0.05	0.10	0	2023-05-19	2023-04-19	17:35:53	Wednesday May 31 6:30 AM ET	2.50
FRC	FIRST REPUBLIC BANK	14.14	14.53	12.72	171.09 2022-08-16	11.52 2023-03-20	2.50	0.05	0.05	0.10	2	2023-05-12	2023-04-19	18:12:38	Monday Apr 24 After the close	N/A
FRC	FIRST REPUBLIC BANK	14.14	14.53	12.72	171.09 2022-08-16	11.52 2023-03-20	2.50	0.10	0.05	0.10	1092	2023-05-19	2023-04-19	18:12:38	Monday Apr 24 After the close	N/A

Full Stack Development with REXX

About the Presenter.

Larry has over 40 years of Enterprise IT experience. He started his career as a S/370 Mainframe operator. He first learned Rexx in 1993 on VM/CMS, and has since programmed on just about every Rexx platform. In his spare time, Larry enjoys retro computing and likes to run Legacy IBM operating systems using the Hercules Emulator.

larryschacher@gmail.com