



## Navegador

# Lynx

por Reinaldo Marques de Lima

*Uma abordagem mais aprofundada sobre o Navegador em modo texto Lynx, suas funções, configurações, algumas dicas e sincronia com scripts em Shell.*



## Índice

### **Apresentação;**

Sobre este documento e seus direitos autorais.

### **Conhecendo o navegador lynx;**

Um pouco sobre a história, propósito e criação do navegador Lynx.

### **Onde conseguir o Lynx;**

Link da internet e meios de conseguir este programa caso não existe no Sistema Operacional.

### **Comandos mais usuais;**

Aprendendo a navegar em modo texto.

### **Opções do Menu Inicial;**

Iniciando o navegador.

### **Algumas opções de Parâmetros;**

Passando opções pela linha de comando.

### **Variáveis de Ambiente;**

Introdução à configuração do navegador.

### **Executando cgi's em Lynx, você sabia?**

Dicas para auxiliar a criação de cgi's, para administração de servidores web.

### **O arquivo lynx.cfg;**

Uma visão geral sobre o arquivo de configuração que é lido pelo navegador Lynx.

### **Algumas Dicas;**

Dicas para facilitar a interação com o usuário no dia-a-dia.

### **Efetando pesquisas;**

Como automatizar a navegação em sites de busca.

### **Exemplos de Shell-script com Lynx;**

Alguns exemplos bem didáticos e funcionais de scripts em Shell interagindo diretamente com o navegador Lynx.



Olá,

Gostaria de parabeniza-lo por ter adquirido este documento que visa abordar em profundidade um assunto não muito difundido na internet, uma ferramenta poderosa em muitos casos e que foi, na medida do possível, esmiuçada para um levar até o usuário deste documento uma análise concisa sobre o assunto. Esta ferramenta é o navegador modo texto Lynx.

### **Especificações técnicas**

A versão do navegador Lynx utilizada para criação deste documento é: **2.8.5rel.1 de 04 de Fevereiro de 2004**, e o interpretador utilizado para criação dos scripts foi o **bash** na versão **3.00.16**.

Recomendo que, para um bom aproveitamento deste documento, a versão dos programas seja compatível ou superior.

### **Direitos Autorais**

Este documento está devidamente registrado (), portanto não sendo ele "de código aberto", "free", "open source" nem com licença GPL, constituindo crime sobre os direitos autorais a sua distribuição indevida por meio de copias ilegais . A única maneira de se adquirir este documento legalmente é entrando em contato direto com o autor (Eu).

```
+-----+
| Esta copia pertence à:
|
| Podendo assim efetuar impressão para uso
| próprio ou para empréstimo sem fins lucrativos.
|
| Não podendo comercializa-la de qualquer forma
| sendo por meio impresso ou eletrônico.
+-----+
```



### **Conhecendo o Navegador Lynx**

O Lynx é um navegador em modo texto onde você pode se conectar a qualquer site da internet pelo seu terminal de trabalho por exemplo, ele foi desenvolvido na Universidade do Kansas - Estados Unidos para, dentre outras tarefas, possibilitar uma navegação rápida caso se queira ler o conteúdo de uma página sem precisar abrir outro navegador. A seguir veremos um pouco mais que o lynx pode oferecer.

### **Onde conseguir o Lynx**

Caso você não tenha o Lynx instalado em seu computador ele pode ser baixado da internet gratuitamente em instalado manualmente através dos links:

*Site oficial -*

`http://lynx.browser.org/`

*Última versão estável ( até a ultima atualização deste documento ) -*

`http://lynx.isc.org/lynx2.8.5/index.html`

*Versão em testes ( não estável ) -*

`http://lynx.isc.org/current/`

*Mirrors -*

`http://lynx.isc.org/mirrors.html`

Se você utiliza o Debian, ou alguma variação como por exemplo o Ubuntu, basta, na linha de comando como root, digitar:

```
prompt> apt-get install lynx
```

Usando o "smart":

```
prompt> smart install lynx -y
```

Depois de finalizada a instalação, basta chama-lo pela linha de comando:

```
prompt> lynx
```

E Para obter ajuda utilize:

`man lynx`, `lynx -help` e `info lynx`

E seu arquivo de configuração fica em:

`/etc/lynx.cfg`

Existe ainda a versão do lynx para DOS, bom para quem trabalho com os dois ambientes e precisa desse recurso. Ele pode ser conseguido em:

<http://www.fdisk.com/doslynx/lynxport.htm>

### **Comandos mais usuais**

#### **Opções do Menu Inicial**

Ao iniciar o lynx chamando o programa na linha de comando aparecerá uma página com

alguns links de ajuda do navegador, para escolher algum deles basta mover a seta do teclado para cima ou para baixo, seta para direita para seguir o link e seta para esquerda para voltar.

```
-- press space for next page --  
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.  
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list   
menu inicial do lynx
```

Caso queira que seja exibido alguns tópicos de ajuda digite no teclado "h" ( de help ) que, caso você esteja navegando em algum site, mostrará a tela inicial do lynx como se você tivesse ativado ele sem nenhum endereço.

Para buscar algum conteúdo na página a qualquer momento digite "/" seguida do que se deseja achar e de <ENTER>, esta opção é muito parecida com a do VI.

Para acessar algum site primeiro digitamos "g" ( de Go ) no teclado e em seguida digitamos o endereço da URL.

Digitando a letra "p" mostramos as opções de impressão, onde podemos salvar a página, envia-la por e-mail, etc.

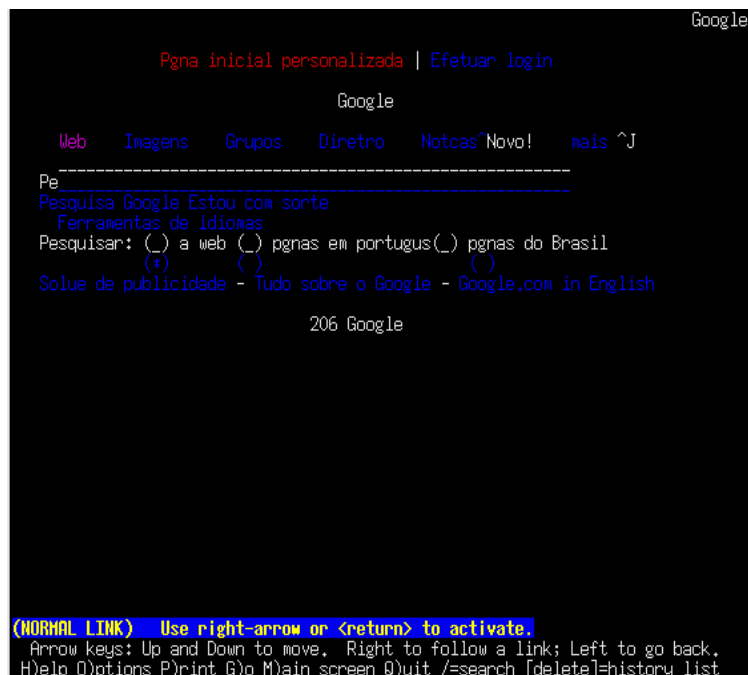
Caso não tenha sido especificado nenhum parâmetro na chamada do programa, digitando "o" teremos as opções de configuração do navegador.

Quando a navegação está bastante avançada e você precisa voltar a página inicial basta digitar "m" para voltar tudo.

Podemos também especificar uma URL logo na chamada do programa, como por exemplo:

```
user@linux~$ lynx www.google.com
```

Página do Google



Caso nenhum parâmetro seja especificado, antes de acessar o site o navegador pergunta se quer aceitar cookies da página à ser acessada.

### **Algumas Opções de Parâmetros**

Aqui veremos alguns opções de parâmetros que podem ser inseridos na linha de comando junto com a chamada do programa para que sejam realizadas determinadas ações. São elas:

**-accept\_all\_cookies** - Use este parâmetro para que o lynx aceite todos os cookies que a página tenta enviar, sabemos que cookies são sempre um assunto polêmico se tratando de internet, vai de cada um aceitar os cookies da página ou não.

**-book** - Habilita uma página em html como bookmark, sendo a primeira página chamada ao iniciar o navegador.

**-case** - Habilita o modo case-sensitive enquanto o lynx estiver ativo nessa navegação.

**-center** - Centraliza o texto da página.

**-color** - Usa, se estiver disponível, o modo colorido que diferencia por cores os links, imagens, textos, etc...

**-connect\_timeout=N** - Define um tempo de conexão, onde N é o tempo em segundos.

**-dump** - Modo que retira o texto da página desabilitando os tags do HTML, muito bom para ser usado em scripts.

**-emacskeys** - Usa o padrão do editor emacs para navegação.

**-error\_file=FILE** - Define um arquivo onde o lynx vai reportar os códigos de erro HTTP.

**-homepage=URL** - Define um endereço URL como página inicial.

**-index=URL** - Define um endereço URL como página inicial, não necessariamente a index da página web.

**-justify** - Justifica o texto da página.

**-nolist** - Desabilita a lista de links destacados no dump.

**-source** - Faz a mesma coisa que o dump, mas mantém as características de HTML da página.

**-use\_mouse** - Permite, se disponível a utilização do mouse na navegação.

**-vikeys** - Usa o padrão do editor vi para navegação.

**-width=NUMBER** - Especifica o tamanho de colunas para formatação do dump, por padrão o número é 80.

### **Variáveis de Ambiente**

Neste tópico abordarei algumas das principais variáveis de ambiente do lynx, que podem ser habilitadas para que o navegador adote um determinado comportamento, listarei as variáveis que mais se enquadram no propósito deste tutorial, que é a utilização do navegador lynx em sincronia com scripts em shell.

Podemos dizer que este tópico é triplo, pois ao final dele abordarei um pouco sobre cgi's no lynx, que vai ser visto com mais ênfase no próximo tópico, e como isto está altamente relacionado com o arquivo de configuração do lynx, este também será abordado com calma mais adiante.



**NOTA:** A maioria das variáveis de ambiente listadas aqui ,por padrão, não estão ativas. Para atribuir um valor basta executar normalmente em linha de comando, use:

```
prompt> VARIÁVEL=valor
```

E para torna-la global use:

```
prompt> export $VARIÁVEL
```

Variável	Descrição
<b>LYNX_CFG</b>	Utiliza outro arquivo de configuração para o navegador, diferente do arquivo localizado em /etc/lynx.cfg.
<b>MAIL</b>	Esta variável, se declarada, irá especificar uma caixa de entradas onde o lynx vai verificar a existência de novos emails.

Existem algumas outras variáveis de ambiente, mas como eu havia dito, para o propósito de criação de scripts, não são necessárias muitas configurações no navegador.

Caso as regras de execução de cgi's for configurada no arquivo lynx.cfg, as seguintes variáveis de ambiente em cgi passam a ser aceitas também pelo navegador lynx:

CONTENT\_LENGTH

CONTENT\_TYPE

DOCUMENT\_ROOT

HTTP\_ACCEPT\_CHARSET

HTTP\_ACCEPT\_LANGUAGE

HTTP\_USER\_AGENT

PATH\_INFO

PATH\_TRANSLATED

QUERY\_STRING

REMOTE\_ADDR

REMOTE\_HOST

REQUEST\_METHOD

SERVER\_SOFTWARE

#### ***Executando cgi's em Lynx, você sabia?***

Pois é meus amigos, é a mais pura verdade, quando você está criando seu script, independente da linguagem, para ser usando como cgi para uma página web, as vezes fica um tanto complicado ficar toda hora abrindo o navegador, e testando o cgi, para tornar as coisas mais dinâmicas eis que surge mais uma vez o lynx para o salvamento.

A principio a configuração padrão do lynx aceita a execução de qualquer cgi, basta chama-lo pela linha de comando:

```
prompt> lynx http://localhost/cgi-bin/dir\_teste/script.cgi
```

Mas para efeito de segurança, e também para ficar mais fácil de aplicar algumas regras na execução do cgi o lynx aceita, através de seu arquivo de configuração, que se especifique algumas regras. Eis aqui o trecho do arquivo que precisa ser editado para aplicar estas regras:

```

.h1 CGI scripts
# These settings control Lynx's ability to execute various types of scripts.

.h2 LOCAL_EXECUTION_LINKS_ALWAYS_ON
.h2 LOCAL_EXECUTION_LINKS_ON_BUT_NOT_REMOTE
# Local execution links and scripts are by default completely disabled,
# unless a change is made to the userdefs.h file to enable them or
# the configure script is used with the corresponding options
# (--enable-exec-links and --enable-exec-scripts).
# See the Lynx source code distribution and the userdefs.h
# file for more detail on enabling execution links and scripts.
#
# If you have enabled execution links or scripts the following
# two variables control Lynx's action when an execution link
# or script is encountered.
#
# If LOCAL_EXECUTION_LINKS_ALWAYS_ON is set to TRUE any execution
# link or script will be executed no matter where it came from.
# This is EXTREMELY dangerous. Since Lynx can access files from
# anywhere in the world, you may encounter links or scripts that
# will cause damage or compromise the security of your system.
#
# If LOCAL_EXECUTION_LINKS_ON_BUT_NOT_REMOTE is set to TRUE only
# links or scripts that reside on the local machine and are
# referenced with a URL beginning with "file://localhost/" or meet
# TRUSTED_EXEC or ALWAYS_TRUSTED_EXEC rules (see below) will be
# executed. This is much less dangerous than enabling all execution
# links, but can still be dangerous.
#
LOCAL_EXECUTION_LINKS_ALWAYS_ON:FALSE
LOCAL_EXECUTION_LINKS_ON_BUT_NOT_REMOTE:FALSE

.h2 TRUSTED_EXEC
# If LOCAL_EXECUTION_LINK_ON_BUT_NOT_REMOTE is TRUE, and no TRUSTED_EXEC
# rule is defined, it defaults to "file://localhost/" and any lynxexec
# or lynxprog command will be permitted if it was referenced from within
# a document whose URL begins with that string. If you wish to restrict the
# referencing URLs further, you can extend the string to include a trusted
# path. You also can specify a trusted directory for http URLs, which will
# then be treated as if they were local rather than remote. For example:
#
#     TRUSTED_EXEC:file://localhost/trusted/
#     TRUSTED_EXEC:http://www.wfbr.edu/trusted/
#
# If you also wish to restrict the commands which can be executed, create
# a series of rules with the path (Unix) or command name (VMS) following
# the string, separated by a tab. For example:
#
# Unix:
# ====
#     TRUSTED_EXEC:file://localhost/<tab>/bin/cp
#     TRUSTED_EXEC:file://localhost/<tab>/bin/rm
# VMS:
# ===
#     TRUSTED_EXEC:file://localhost/<tab>copy
#     TRUSTED_EXEC:file://localhost/<tab>delete
#
# Once you specify a TRUSTED_EXEC referencing string, the default is
# replaced, and all the referencing strings you desire must be specified
# as a series. Similarly, if you associate a command with the referencing
# string, you must specify all of the allowable commands as a series of
# TRUSTED_EXEC rules for that string. If you specify ALWAYS_TRUSTED_EXEC
# rules below, you need not repeat them as TRUSTED_EXEC rules.
#
# If EXEC_LINKS and JUMPFILe have been defined, any lynxexec or lynxprog

```

```
# URLs in that file will be permitted, regardless of other settings. If
# you also set LOCAL_EXECUTION_LINKS_ON_BUT_NOT_REMOTE:TRUE and a single
# TRUSTED_EXEC rule that will always fail (e.g., "none"), then *ONLY* the
# lynxexec or lynxprog URLs in JUMPFILe (and any ALWAYS_TRUSTED_EXEC rules,
# see below) will be allowed. Note, however, that if Lynx was compiled with
# CAN_ANONYMOUS_JUMP set to FALSE (default is TRUE), or -restrictions=jump
# is included with the -anonymous switch at run time, then users of an
# anonymous account will not be able to access the jumps file or enter
# 'j'ump shortcuts, and this selective execution feature will be overridden
# as well (i.e., they will only be able to access lynxexec or lynxprog
# URLs which meet any ALWAYS_TRUSTED_EXEC rules).
#
TRUSTED_EXEC:none
```

## .h2 ALWAYS\_TRUSTED\_EXEC

```
# If EXEC_LINKS was defined, any lynxexec or lynxprog URL can be made
# always enabled by an ALWAYS_TRUSTED_EXEC rule for it. This is useful for
# anonymous accounts in which you have disabled execution links generally,
# and may also have disabled jumps file links, but still want to allow
# execution of particular utility scripts or programs. The format is
# like that for TRUSTED_EXEC. For example:
#
# Unix:
# ====
# ALWAYS_TRUSTED_EXEC:file://localhost/<tab>/usr/local/kinetic/bin/usertime
# ALWAYS_TRUSTED_EXEC:http://www.more.net/<tab>/usr/local/kinetic/bin/who.sh
# VMS:
# ===
# ALWAYS_TRUSTED_EXEC:file://localhost/<tab>usertime
# ALWAYS_TRUSTED_EXEC:http://www.more.net/<tab>show users
#
# The default ALWAYS_TRUSTED_EXEC rule is "none".
#
ALWAYS_TRUSTED_EXEC:none
```

## .h2 TRUSTED\_LYNXCGI

```
# Unix:
# =====
# TRUSTED_LYNXCGI rules define the permitted sources and/or paths for
# lynxcgi links (if LYNXCGI_LINKS is defined in userdefs.h). The format
# is the same as for TRUSTED_EXEC rules (see above), but no defaults are
# defined, i.e., if no TRUSTED_LYNXCGI rules are defined here, any source
# and path for lynxcgi links will be permitted. Example rules:
#
# TRUSTED_LYNXCGI:file://localhost/
# TRUSTED_LYNXCGI:<tab>/usr/local/etc/httpd/cgi-bin/
# TRUSTED_LYNXCGI:file://localhost/<tab>/usr/local/www/cgi-bin/
#
# VMS:
# =====
# Do not define this.
#
TRUSTED_LYNXCGI:none
```

## .h2 LYNXCGI\_ENVIRONMENT

```
# Unix:
# =====
# LYNXCGI_ENVIRONMENT adds the current value of the specified
# environment variable to the list of environment variables passed on to the
# lynxcgi script. Useful variables are HOME, USER, etc... If proxies
# are in use, and the script invokes another copy of lynx (or a program like
# wget) in a subsidiary role, it can be useful to add http_proxy and other
# *_proxy variables.
#
# VMS:
# =====
# Do not define this.
```

```
#
#LYNXCGI_ENVIRONMENT:

.h2 LYNXCGI_DOCUMENT_ROOT
# Unix:
# =====
# LYNXCGI_DOCUMENT_ROOT is the value of DOCUMENT_ROOT that will be passed
# to lynxcgi scripts. If set and the URL has PATH_INFO data, then
# PATH_TRANSLATED will also be generated. Examples:
#     LYNXCGI_DOCUMENT_ROOT:/usr/local/etc/httpd/htdocs
#     LYNXCGI_DOCUMENT_ROOT:/data/htdocs/
#
# VMS:
# =====
# Do not define this.
#
#LYNXCGI_DOCUMENT_ROOT:
```

---

Para aplicar as regras necessárias para execução de cgi's pelo navegador lynx basta modificar as linhas:

```
LOCAL_EXECUTION_LINKS_ALWAYS_ON:FALSE
LOCAL_EXECUTION_LINKS_ON_BUT_NOT_REMOTE:FALSE
TRUSTED_EXEC:none
ALWAYS_TRUSTED_EXEC:none
TRUSTED_LYNXCGI:none
#LYNXCGI_ENVIRONMENT:
#LYNXCGI_DOCUMENT_ROOT:
```

### O arquivo lynx.cfg

Já que estamos falando de configuração, que tal dar um passeio por alguns trechos relevantes do arquivo lynx.cfg e comentar algumas de suas funções?

Então, aperte os cintos 8^).

Antes de mais nada, quero reafirmar que listarei alguns trechos mais relevantes do arquivo lynx.cfg, unicamente para o propósito de criação de scripts, pois o arquivo é enorme, e não faz sentido coloca-lo aqui, porque tornaria a leitura entediante, e sem sentido, além de fugir do escopo.

---

```
.h2 CONNECT_TIMEOUT
# Specifies (in seconds) connect timeout. Default value is rather huge.
#CONNECT_TIMEOUT:18000
```

*Determina o tempo de conexão do navegador, se configurado tem tempo default de 5 horas, altere para o tempo desejado, em segundos.*

```
.h2 CASE_SENSITIVE_ALWAYS_ON
# The default search type.
# This is a default that can be overridden by the user!
#
#CASE_SENSITIVE_ALWAYS_ON:FALSE
```

*Altera a definição de case sensitive, para reconhecer parâmetros maiúsculos e minúsculos.*

```
.h2 DEFAULT_BOOKMARK_FILE
```

```
# DEFAULT_BOOKMARK_FILE is the filename used for storing personal bookmarks.
# It will be prepended by the user's home directory.
# NOTE that a file ending in .html or other suffix mapped to text/html
# should be used to ensure its treatment as HTML. The built-in default
# is lynx_bookmarks.html. On both Unix and VMS, if a subdirectory off of
# the HOME directory is desired, the path should begin with "./" (e.g.,
# ./BM/lynx_bookmarks.html), but the subdirectory must already exist.
# Lynx will create the bookmark file, if it does not already exist, on
# the first ADD_BOOKMARK attempt if the HOME directory is indicated
# (i.e., if the definition is just filename.html without any slashes),
# but requires a pre-existing subdirectory to create the file there.
# The user can re-define the default bookmark file, as well as a set
# of sub-bookmark files if multiple bookmark file support is enabled
# (see below), via the 'o'ptions menu, and can save those definitions
# in the .lynxrc file.
#
#DEFAULT_BOOKMARK_FILE:lynx_bookmarks.html
```

*Especifica uma página em html para ser utilizada como bookmark do navegador.*

```
.h2 BLOCK_MULTI_BOOKMARKS
# If BLOCK_MULTI_BOOKMARKS is set TRUE, multiple bookmark support will
# be forced off, and cannot to toggled on via the 'o'ptions menu. The
# compilation setting is normally FALSE, and can be overridden here.
# It can also be set via the -restrictions=multibook or the -anonymous
# or -validate command line switches.
#
#BLOCK_MULTI_BOOKMARKS:FALSE
```

*Especifica uma página em html para ser utilizada como bookmark do navegador para páginas que NÃO podem ser acessadas.*

```
.h1 Proxy
#http_proxy:http://some.server.dom:port/
#https_proxy:http://some.server.dom:port/
#ftp_proxy:http://some.server.dom:port/
#gopher_proxy:http://some.server.dom:port/
#news_proxy:http://some.server.dom:port/
#newspost_proxy:http://some.server.dom:port/
#newsreply_proxy:http://some.server.dom:port/
#snews_proxy:http://some.server.dom:port/
#snewspost_proxy:http://some.server.dom:port/
#snewsreply_proxy:http://some.server.dom:port/
#nntp_proxy:http://some.server.dom:port/
```

```
#wais_proxy:http://some.server.dom:port/  
#finger_proxy:http://some.server.dom:port/  
#cso_proxy:http://some.server.dom:port/  
#no_proxy:host.domain.dom
```

Define o endereço de servidores proxy, para diversas tecnologias.

### Algumas Dicas:

Dica show, e simples. Para baixar um site da Web direto para o VI:

```
:r !lynx -dump http://sitedaweb.com (funciona com servidor local inclusive)
```

O navegador lynx tem algumas particularidades na sua execução em linha de comando como por exemplo:

- Colocar a linha de comando em background quando existe um endereço com um "&" no meio.

Exemplo: <http://www4.climatempo.com.br/site/espelho.php?estados=SP&pg=capitais&pc=estadao>

Para resolver isto basta colocar uma barra invertida antes do "&" negando-o como se faz em expressões regulares por exemplo, ficando desta maneira:

<http://www4.climatempo.com.br/site/espelho.php?estados=SP\&pg=capitais\&pc=estadao>

Assim a linha de comando reconhece o "&" como um caracter do próprio endereço web.

### Efetuando pesquisas

Uma maneira bem legal de efetuar pesquisas em páginas como a do google por exemplo, é passar todo o preenchimento dos campos de pesquisa pelo endereço que vai na linha de comando, por exemplo:

Pelo navegador ( no meu caso o Mozilla Firefox ), vamos pesquisar a palavra "Linux" no site do google usando as opções de pesquisa, ( Palavra Linux, páginas em português do Brasil ) e mandar pesquisar. Note que a barra de endereço mostra algo assim:

"<http://www.google.com.br/search?hl=pt-BR&q=Linux&btnG=Pesquisa+Google&meta=cr%3DcountryBR>"


Agora se dermos um **Ctrl+i** ( ou no navegador de sua preferência, encontre a opção "Propriedades da Página" ) na página inicial do google aparecerá uma caixa com informações sobre a página, selecionando a aba **Form** podemos ver umas colunas onde mostra-se algo mais ou menos assim:

Label	Field Name	Type	Current Value
	hl	hidden	pt-BR
	q	text	Linux
	btnG	submit	Pesquisa Google

	btnl	submit	Estou com Sorte
a web	meta	radio	
páginas em português	meta	radio	lr=lang_pt
páginas do Brasil	meta	radio	cr=countryBR

Ao analisar os campos que o google usa e a linha que aparece no navegador podemos perceber que o mecanismo de busca do google simplesmente completa os campos para a pesquisa, onde no campo **text** vai a palavra ou expressão que se está procurando, nada mais natural. A partir dai conseguimos criar um script para efetuar esta mesma pesquisa sem precisar do navegador, não conseguimos? "Claro que sim..."

Então vamos lá:

	<p>Script: busca_google.bsh</p> <p>Script simples que efetua pesquisas no site do google.</p>
<pre>#!/bin/bash # # Script simples que faz pesquisas no site '<a href="http://www.google.com">http://www.google.com</a>' # # Criado para simples didática, sem tratativas de erro ou melhorias similares # LYNX="/usr/bin/lynx" WORD=\$1 SEARCH=\$( \$LYNX -dump <a href="http://www.google.com.br/search?hl=pt-BR&amp;q=\$WORD">http://www.google.com.br/search?hl=pt-BR&amp;q=\$WORD</a> \&amp;btnG=Pesquisa+Google \&amp;meta=cr%3DcountryBR   sed 's/\[[^\]]*\ //') # Um 'sedzinho' no final para remover os lixos deixados pelo lynx echo "\$SEARCH"</pre>	

### Exemplos de Shell-script com Lynx

Talvez o mais conhecido exemplo que circula na internet de como utilizar o poder do lynx em um script sejam as **FunçõesZZ** <http://funcoeszz.net/> escritas em conjunto entre o Aurélio e o Thobias, um programa que usa e abusa do lynx para buscar informações em sites e mostra-las na tela do terminal. Claro que não vou nem arriscar a fazer algo equivalente, mas tentarei dar alguns exemplos de utilização dessas duas ferramentas ( lynx e shell-script ) em conjunto.

Aqui mostrarei dois códigos de scripts que eu fiz para buscar noticias de sites da Web usando lynx, o script "apinfo.bsh" e o "pesquisa\_grupo.bsh", e um script criado em parceria com o Julio Neves que tenta acessar um site congestionado a força bruta:

```
#!/bin/bash
#####[ Ficha: ]#####
#
```

```

# Nome: Apinfo.bsh #
# Escrito por: Reinaldo Marques de Lima #
# Criado em: 03/2006 #
# Ultima atualizaçãõ: 24/04/2006 #
# #
#####[ Descrição: ]#####
# #
# Script que acessa o site www.apinfo.com (site de ofertas de emprego para#
# profissionais de informatica) e gera uma pagina html com as ofertas do #
# dia. #
# #
#####

#( Variaveis ).....#

LYNX=$(which lynx)
GREP=$(which grep)
HEAD=$(which head)
SED=$(which sed)
CUT=$(which cut)

#( Funcao apinfo, acessa o site e gera um tmp com links das ofertas )....#

apinfo(){

URL=http://www.apinfo3.com/pp7w.htm

#####[ Filtra usando o canivete ( grep, head, cut...) ]#####
#$LYNX -source $URL | $GREP \<p\>\<a | $HEAD -2 | $CUT -d\" -f2 > /tmp/apinfo.tmp
#
#####[ Aqui usa só o sed ]#####
$LYNX -source $URL | $SED '/<p><a href=.*!/d;
s/<p><a href=\"\(.*\)\">/\1/g' | sed '1,2!d' > /tmp/apinfo.tmp

txt
}

#( Fucao txt acessa os link usando array, filtra a pagina e gara um txt)..#

txt(){

LINKS=$(cat /tmp/apinfo.tmp)

```

```

for i in ${LINKS[*]}; do

    $LYNX -dump $i | $SED 's/\[[^]]*\]/g;
    /References/q' >> /tmp/apinfo.txt

done

html
}

#( Funcao html pega o txt e converte em um html, facil de visualizar )....#

html(){

DATA=$(date +%d/%m/%Y)
DATA2=$(date +%d_%m_%Y)
TEXTO=$( $SED 's/[_]\{10,\}/<HR NOSHADE SIZE=1>/g;
/References/d' /tmp/apinfo.txt)

TITLE="<TITLE>
Ofertas de emprego do dia $DATA
</TITLE>"

HEAD="<HEAD>
$TITLE
<H1>
<CENTER>
<B>
Script que acessa o Site Apinfo.com
</B>
</CENTER>
</H1>
<H2>
<CENTER>
<B>
Ofertas de emprego do dia $DATA
<HR NOSHADE SIZE=1>
</B>
</CENTER>
</H2>
</HEAD>"

BODY="<BODY bgcolor="white">

```

```

<FONT SIZE=3 FACE=verdana>
<PRE>
$TEXTO
</PRE>
</FONT>
</BODY>"

HTML="<HTML>
$HEAD
$BODY
</HTML>"

printf "$HTML" > apinfo_$DATA2.html

rm /tmp/apinfo.tmp
rm /tmp/apinfo.txt

}

#( ultima linha, chama a primeira funcao que desencadeia o processo )....#

apinfo

#( fim ).....#

```

---

```

#!/bin/bash
#####[ Ficha: ]#####
#
# Script: Pesquisa Grupo
# Escrito por: Reinaldo Marques de Lima
# Criado em: 05/03/2006
# Ultima Atualizacao: 05/03/2006
#
#####[ Descricao: ]#####
#
# Script que acessa a pagina do grupo de shell-script e gera uma pagina
# em html com o resultado da pesquisa.
# O nome do arquivo gerado eh 'pesquisa_(chave de pesquisa).html'.
#
# TODO:
# Gerar pagina com todos os resultados da pesquisa, atualmente o script

```

```

# so pega a primeira pagina de resultados. #
# #
#####

#( Teste ).....#

[ -z $1 ] && echo "erro: use $0 [palavra]: sed ( por exemplo)" && exit

#( Variaveis ).....#

WORD=$1
SED=$(which sed)
CAT=$(which cat)
LYNX=$(which lynx)
URL=http://search.gmane.org/search.php?group=gmane.org.user-
groups.programming.shell.brazil\&query=

#( Funcao 'grupo', acessa a pagina e gera um tmp ).....#

grupo(){

$LYNX -source $URL$WORD | $SED -n '/<A HREF=.*>/p' | $SED 's/<A
HREF="\(.*\)">.*\1/;/^http:\\\.*\d;s/<B>\(.*\)/\1/' > /tmp/links.tmp

txt
}

#( Funcao 'txt', usa o tmp criado como array para gerar a pagina ).....#

txt(){

LINKS=$( $CAT /tmp/links.tmp )

for i in ${LINKS[*]}; do

    $LYNX -dump $i | $SED '1,27d;
    /\[paint-list-id\.php?group=gmane\.org\.user-
groups\.programming\.shell\.b$/q;
    s/\[[^\]]*\]/g' | $SED 's/ \[30\].*/<HR NOSHADE SIZE=1>/' >>
/tmp/links.txt

#[ Caso aqui ^ de erro, coloque tudo em uma linha soh ].....#

done

```

```
html
}
```

```
##( Funcao 'html', pega o txt e insere tags para gerar a pagina html ).....#
```

```
html(){
```

```
TEXTO=$(SED 's/[_]\{10,\}/<HR NOSHADE SIZE=1>/g' /tmp/links.txt)
```

```
TITLE="<TITLE>
```

```
Resultado da Pesquisa com a palavra: $WORD
```

```
</TITLE>"
```

```
HEAD="<HEAD>
```

```
$TITLE
```

```
<H1>
```

```
<CENTER>
```

```
<B>
```

```
Script que acessa o site do grupo de Shell-Script
```

```
</B>
```

```
</CENTER>
```

```
</H1>
```

```
<H2>
```

```
<CENTER>
```

```
<B>
```

```
Resultado da Pesquisa com a palavra: $WORD
```

```
<HR NOSHADE SIZE=1>
```

```
</B>
```

```
</CENTER>
```

```
</H2>
```

```
</HEAD>"
```

```
BODY="<BODY bgcolor="white">
```

```
<FONT SIZE=3 FACE=verdana>
```

```
<PRE>
```

```
$TEXTO
```

```
</PRE>
```

```
</FONT>
```

```
</BODY>"
```

```
HTML="<HTML>
```

```
$HEAD
```

```
$BODY
</HTML>"
```

```
echo "$HTML" > pesquisa_${WORD}.html
```

```
rm /tmp/links.tmp
```

```
rm /tmp/links.txt
```

```
}
```

```
##( Aqui eh chamada a primeira funcao ).....#
```

```
grupo
```

```
##( fim ).....#
```

```
#!/bin/bash
```

```
#####[ Ficha: ]#####
```

```
# #
```

```
# Script: force_lynx.bsh #
```

```
# Escrito por: Reinaldo Marques de Lima ( Plastico ) e Julio Cezar Neves #
```

```
# Criado em: 24/05/2006 #
```

```
# Ultima Atualizacao: 26/05/2006 #
```

```
# #
```

```
#####[ Descricao: ]#####
```

```
# #
```

```
# Script que tenta acessar um determinado site repetidas vezes, ideal #
```

```
# para acessar sites congestionados atraves do navegador lynx, que demora #
```

```
# menos para carregar uma pagina. #
```

```
# #
```

```
#####[ How: ]#####
```

```
# #
```

```
# O script recebe na linha de comando uma URL a ser acessada, com numero #
```

```
# definido de vezes ou nao (em caso negativo inicia-se um loopin eterno), #
```

```
# que abre um novo terminal, ja chamando o navegador lynx e a URL pela #
```

```
# linha de comando, tentando acessar a URL atraves de forza bruta. #
```

```
# Para isso o usuário precisa finalizar o terminal ao perceber a demora #
```

```
# no acesso. #
```

```
# #
```

```
#####
```

```
# #
```

```
##( Variaveis ).....#
```

```

#                                                                 #
PING=$(which ping)          #####
KEY=0                       # indice inicial que controla o while;
URL=$1                      # endereco a ser acessado;
LYNX=$(which lynx)         # teste de portabilidade;
                            #####
#                                                                 #
#( Testes ).....#
#                                                                 #

[ $1 ] || {
    echo "$0: Erro: Use: $0 URL [N]: Onde N eh o numero de tentativas
(vazio=infinito). "
    exit 1; }

ping -c1 "$1" > /dev/null || { echo "URL inválida ou sem rede"; exit 1; }

TRY=${2:-1}    # Se $2 estiver definido TRY recebe $2, senao -1

trap "clear; echo fim da conexão, Tchau; exit" 0 2 3 # Para interromper loop
infinito

#                                                                 #
#( Funcao ).....#
#                                                                 #
tentar()
{
    while [ $KEY != $TRY ]; do
        xterm -T $URL -e $LYNX --accept_all_cookies $URL &> /dev/null
        let KEY++
    done

    read -p "Nao foi possivel acessar $URL. Quer tentar novamente? (N/s) " RESPOSTA
    RESPOSTA=${RESPOSTA:-n}
    [ `echo $RESPOSTA | tr N n` = n ] &&
    {
        echo "fim da conexão, Tchau"
        exit
    }
    KEY=0    # reinicia o indice do while
    tentar  # e chama novamente a funcao
}
tentar

```