GetEz

ABSTRACT

Internet is the resource center in the present day life. One can get information about anything on the internet. The most striking feature of the internet is that one can not only view the information but also download them and use/view them in the local system. Thus efficient downloading is the need of the hour. There are many tools that provide efficient downloading. WGET is one such tool that provides efficient download. WGET is a GNU utility.

GETEZ is an interface to the wget tool. Wget is a free utility for non-interactive download of files from the Web. It supports HTTP, HTTPS, and FTP protocols, as well as retrieval through HTTP proxies. Our project aims at providing an interface to the wget tool.

The application is developed using glade as the front end and gtk+ as the back end. Basic C functions are used to implement the various functionality of the wget tool. The application implements almost all the options that are available with the wget tool. The user has to choose the options according to the type and necessity of the download. The download is done in the efficient manner. The user has to specify the URL to be downloaded and can also specify the location as to where the files are to be downloaded. The time of download is displayed.

1 INTRODUCTION

GETEZ is an interface to the wget tool. WGET is the GNU free utility that is used to download files from the internet. The main aim of our project is to provide an interface to the wget tool so that it becomes more easy for the user to perform efficient downloading of the files. The application implements all the mostly used options of the wget. Thus it becomes easy for the user to choose the necessary options for the particular download. The application is developed using the glade as the front end and the gtk+ as the back end. The options are got from the user. The user is required to specify the URL from which the download is to be done. The user may also mention the destination location in the local system as to where the downloaded files are to be saved.

1.1 Wget

This is the free utility which we have implemented in this project. The interface is designed in such a manner that almost all the options that are available with the wget tool are implemented here. The wget tool is a non-interactive download utility. By non-interactive we mean that the it does the job in background. So the user need not be logged on during the download. The wget supports HTTP, HTTPS and the FTP protocols, as well as HTTP proxies. Wget can follow links in HTML and XHTML pages and create local versions of remote web sites, fully recreating the directory structure of the original site. Thus wget is a very useful tool for performing the download efficiently.

1.2 Glade

Glade is an application for creating graphical user interfaces that uses the GTK+ and GNOME libraries. Glade allows to rapidly develop these interfaces, and can create source code in a variety of languages that will construct the interfaces.
Glade is composed of several windows which serves a particular purpose. The main window contains the menu, toolbar and a list of top-level windows. The palette lists the various user interface objects which can be used to build an interface. The property editor is used to manipulate the properties of widgets, such as their size, color, etc. Typically a glade project begins with the construction of the user interface. While doing this the property editor is used to manipulate the various packing settings, dimensions, etc., of the widgets. When the interface is finished, the project is saved and the C source files are built. These source files will be used to generate the user interface. Integration with the project logic then follows.

1.3 gtk+

gtk+ is a multi-platform toolkit for creating graphical user interfaces. It contains common and complex widgets, such as file selection, and color selection widgets. GTK+ was initially developed as a widget set for the GIMP (GNU Image Manipulation Program). It has grown extensively since then, and is today used by a large number of applications, and is the toolkit used by the GNOME desktop project. GTK+ has been designed from the ground up to support a range of language bindings, not only C/C++. Using GTK+ from languages such as Perl and Python (especially in combination with the Glade GUI builder) provides an effective method of rapid application development.

2 WGET - AN OVERVIEW

As mentioned above the main aim of our project is to implement the wget tool. This section will give an overview of the wget tool. The wget is a free utility for non-interactive download of files from the web.

2.1 Features

- Wget is non-interactive, meaning that it can work in the background, while the user is not logged on. This allows to start a retrieval and disconnect from the system, letting Wget finish the work. By contrast, most of the Web browsers require constant user's presence, which can be a great hindrance when transferring a lot of data.

- Wget can follow links in HTML and XHTML pages and create local versions of remote web sites, fully recreating the directory structure of the original site. This is sometimes referred to as "recursive downloading." While doing that, Wget respects the Robot Exclusion Standard (`/robots.txt'). Wget can be instructed to convert the links in downloaded HTML files to the local files for offline viewing.

- File name wildcard matching and recursive mirroring of directories are available when retrieving via FTP. Wget can read the time-stamp information given by both HTTP and FTP servers, and store it locally. Thus Wget can see if the remote file has changed since last retrieval, and automatically retrieve the new version if it has. This makes Wget suitable for mirroring of FTP sites, as well as home pages.

- Wget has been designed for robustness over slow or unstable network connections; if a download fails due to a network problem, it will keep retrying until the whole file has been retrieved. If the server supports re getting, it will instruct the server to continue the download from where it left off.

- Wget supports proxy servers, which can lighten the network load, speed up retrieval and provide access behind firewalls. Wget also supports the passive FTP downloading as an option.

- Built-in features offer mechanisms to tune which links one may wish to follow.

- The retrieval can be conveniently traced. These representations can be customized to user preferences.

2.2 General Downloading Process

How wget works can be understood from the above features. The most important feature of the wget is that it runs in the background and thus the user need not be logged in while the download takes place. Large files generally take a lot of time to download. Although the download speed is dependent on the bandwidth capacity, on an average as the file size is larger the download time takes quite a long time. Another feature that adds to the long time during download is that if the source file to be downloaded is in several steps down the main link it may take some time to go until the depth necessary. There are many more reasons for the downloading time to be high.

Now the usual process is like the user logs in and request for a particular page. The user's presence during the download is usually necessary for during the download some user inputs may be necessary. If any problem occurs then the user is notified by a warning message telling that some problem has occurred and that further download may not be possible. Now the user has to start all over from the beginning once again. This is the usual process and it is really tiresome for the user, especially if the file is big.

2.3 How wget works

What happens during a download was explained in the previous section. Now how wget works? As mentioned above, wget is a non-interactive , download utility. By non-interactive we it is meant that the user intervention will not be necessary during the download operation. The user executes the wget command with the options of his choice (i.e. those options that are necessary for that particular download). Once wget starts running, the download will begin and continue as a background operation. Thus the user need not be logged on to the system.

The user has to just mention the URL from which the download has to take place and also as to where the files must be downloaded. The user also mentions the other options that he wants to enable during the download. By other options we mean the other functionality that one can enable during the download. There are many options available with the wget.

Now once the wget command gets executed, it will start the download. This process will be going on in the background and thus the user will not be having any hindarance in his work. The user may even log off and go but the downloading will continue to process. This is the power of wget.

Files of larger size can be downloaded efficiently using wget. We can just leave the files to be downloaded and carry on with our jobs. Once the files are downloaded completely we will have a complete directory setup similar to the file structure in the specified URL. Thus viewing the files will be very easy since we just have to follow the links. We can also specify in the options field as to whether the links in the HTML files are to be maintained. If this option is checked then the links will converted in such a way that they will point to the file in the local system file structure and mostly in the directory of the downloaded file. Thus when this page is viewed in the browser this will look similar to the file as in the web page and the links will remain. Now if one clicks on this link, the file will be correctly opened since we have made the ;link to point to the file in the local system. What happens is that when the files are being downloaded, the links will be made to point to the local file. The destination path will be changed to here. Thus when you follow the link the file opens correctly.

Wget was mainly designed for robustness over slow and unstable network connections. If a download fails due to some problems in the network, it will keep retrying until the whole file has been retrieved. If the server supports re getting, it will instruct the server to continue the download from where it left off. Thus we can say that wget provides reliable download of files.

3 GETEZ - BASIC DESIGN

The basic design section explains the interface design of the getez. The interface is designed using the glade tool available with the Dedian Sarge flavour of LINUX. The interface can be explained as follows.

There are two tabs available between which the user can switch. The first tab shows the basic options that are available with the getez. The mandatory information that are needed to perform the download are given here. The basic necessary information are as follows:

1. URL (The address from where files are to be downloaded).
2. Destination (The local location where the downloaded files are to be stored).

3.1 Basic Options

1. Index File Only

This option when enabled ensures that only the index page of the required URL is retrieved. The rest pages are not downloaded. Only the index page is downloaded. The index page is the first page in every web site. Thus this option when enabled will download only the index or the first page. Even when the "RECURSIVE" option is enabled if this option is chosen then only the index page will be retrieved.

2. All Files

This is the other choice if "INDEX ONLY" option is not chosen. This option when enabled will ensure that all files are downloaded. The index file together with all the files till the destination file specified by the user is reached are retrieved and stored in the local system.

3. No Clobber

If a file is downloaded more than once in the same directory, getez behavior depends on whether this option is chosen or not. In certain cases, the local file will be "clobbered", or overwritten, upon repeated download. In other cases it will be preserved. When getez is run without this option being run then downloading the same directory will result in the original copy of the FILE being preserved and the second copy named 'FILE.1'. If that file is downloaded again, the third copy will be named 'FILE.2' and so on.

When "no clobber" is specified, this behavior is suppressed, and getez will refuse to download newer copies of 'FILE'. When running with this option causes the original version to be preserved and any newer copies on the server to be ignored.

4. Continue

Continue getting a partially - downloaded file. This is useful when we want to finish up a download started by a previous instance of getez, or by another program. We dont have to choose this option if we want the current invocation of getez to retry downloading a file should the connection be lost midway through. This is the default behavior. This option only affects resumption of downloads started prior to this invocation of getez, and whose local files are still sitting around.

While using with this option any file that is bigger on the server than locally will be considered an incomplete download. This option can thus be used to download just the new portion that is been appended to a data collection or log file.

5. Convert Links

After the download is complete, convert the links in the document to make them suitable for local viewing. This affects not only the visible hyperlinks, but any part of the document that links to external content, such as embedded images, links to style sheets, hyperlinks to non-HTML content, etc.

Each link will be changed in one of the two ways:

* The links to files that have been downloaded by getez will be changed to refer to the file they point to as a relative link.
Example: if the downloaded file `/foo/doc.html' links to `/bar/img.gif', also downloaded, then the link in `doc.html' will be modified to point to `../bar/img.gif'. This kind of transformation works reliably for arbitrary combinations of directories.

* The links to files that have not been downloaded by getez will be changed to include host name and absolute path of the location they point to.
Example: if the downloaded file `/foo/doc.html' links to `/bar/img.gif' (or to `../bar/img.gif'), then the link in `doc.html' will be modified to point to `http://HOSTNAME/bar/img.gif'.

Because of this, local browsing works reliably: if a linked file was downloaded, the link will refer to its local name; if it was not downloaded, the link will refer to its full Internet address rather than presenting a broken link. The fact that the former links are converted to relative links ensures that you can move the downloaded hierarchy to another directory.

Note that only at the end of the download can Wget know which links have been downloaded. Because of that, the work done by this option will be performed at the end of all the downloads.

The various basic options are thus explained. Once these options are chosen the user can start the download by clicking the "DOWNLOAD" button. This means that the user can start the download even without choosing the advanced options. The basic options are just enough for a normal download. If the user does not chose any option but starts the download just by specifying the URL and the destination then default options are automatically included. The default options will mostly satisfy all the user needs during a download. Thus the download can be started just by giving the URL name and the destination location.

3.2 Advanced Options

The advanced options are available with in the next tab. The advanced options are chosen to enable complex options with the download. These options are not generally used but in very special cases their choice is of greater significance.

1. Number of Tries

Here the user can specify the number of times the getez will retry to download a particular file. If 0 is specified then it means that it has to retry "INFINITE" times. The default is to retry 20 times, with the exception of fatal errors like "CONNECTION REFUSED" or "NOT FOUND". The user can specify the number of retries.

2. Limit Rate

Here the user can specify the download speed that he wants. That is if the user wishes that this download operation must not consume the entire bandwidth then he may specify the speed with which this download operation must take place. For example if 20kb is specified then it means that the files will be retrieved at the rate of 20 kb/s. The amount can be specified by the appropriate suffix - 'k' for kilobytes and 'm' for megabytes.

The limiting is implemented by sleeping the appropriate amount of time after a network read that took less time than specified by the rate. Eventually this strategy causes the TCP transfer to slow down to approximately the specified rate. However, it may take some time for this balance to be achieved, so limiting the rate doesn't work well with very small files generally.

3. Turn PROXY ON/OFF

Turn proxy support on or off. The proxy is on by default if the appropriate environment variable is defined.

"Proxies" are special-purpose HTTP servers designed to transfer data from remote servers to local clients. One typical use of proxies is lightening network load for users behind a slow connection. This is achieved by channeling all HTTP and FTP requests through the proxy which caches the transferred data. When a cached resource is requested again, proxy will return the data from cache. Another use for proxies is for companies that separate (for security reasons) their internal networks from the rest of Internet. In order to obtain information from the Web, their users connect and retrieve remote data using an authorized proxy.

4. CACHE ON/OFF

When set to off, disable server-side cache. In this case, Wget will send the remote server an appropriate directive (`Pragma: no-cache') to get the file from the remote service, rather than returning the cached version. This is especially useful for retrieving and flushing out-of-date documents on proxy servers. Caching is allowed by default.

5. COOKIES ON/OFF

When set to off, disable the use of cookies. Cookies are a mechanism for maintaining server-side state. The server sends the client a cookie using the `Set-Cookie' header, and the client responds with the same cookie upon further requests. Since cookies allow the server owners to keep track of visitors and for sites to exchange this information, some consider them a breach of privacy. The default is to use cookies; however, _storing_ cookies is not on by default.

6. Don't Remove Listing Files

Don't remove the temporary `.listing' files generated by FTP retrievals. Normally, these files contain the raw directory listings received from FTP servers. Not removing them can be useful for debugging purposes, or when we want to check on the contents of remote server directories.

Note that even though getez writes to a known filename for this file, this is not a security hole in the scenario of a user making `.listing' a symbolic link to `/etc/passwd' or something and asking `root' to run getez in his or her directory. Depending on the options used, either getez will refuse to write to `.listing', making the globbing/recursion/time-stamping operation fail, or the symbolic link will be deleted and replaced with the actual `.listing' file, or the listing will be written to a `.listing.NUMBER' file.

Even though this situation isn't a problem, though, `root' should never run getez in a non-trusted user's directory.

7. GLOB ON/OFF

Turn FTP globbing on or off. Globbing means we may use the shell-like special characters ("wildcards"), like `*', `?', `[' and `]' to retrieve more than one file from the same directory at once, like:

ftp://gnjilux.srk.fer.hr/*.msg

By default, globbing will be turned on if the URL contains a globbing character. This option may be used to turn globbing on or off permanently. The user may have to quote the URL to protect it from being expanded. Globbing makes getez look for a directory listing, which is system-specific. This is why it currently works only with Unix FTP servers (and the ones emulating Unix `ls' output).

4 HOW TO USE GETEZ

Now that we saw the various options available with the getez now it is time to see how getez works. This application is used for downloading files. So the basic requirement is very obvious - Access to the internet is mandatory. This basic requirement is very obvious and does not need any more mention in this document. However if files are to be got from a computer in the local network then it is required that the computer on which this application will run should also be in the network.

It is time to run the application. On running getez the window appears.

The basic options are available with this window. The necessary inputs that are needed for the download are to be given in the two text fields available. There are labels indicating what values are to be given in the text fields. The two inputs that

are required are mentioned in the beginning itself. They are the URL and the destination location.

The basic options follow. The options are available as either checkboxes or radio buttons. The user has to choose the options of his choice and carry on. This interface is designed in such a manner that it is easier for the user to use. If the user is satisfied with the basic options and feels that these options are enough for his download then he can carry on with the download. However the advanced options are available in the next tab.

The advanced options are also available as either checkboxes or radio buttons. The "NUMBER OF TRIES" options is available is a drop down list from which the user can choose any value of his choice. However the user can give the value manually also. The "LIMIT RATE" option has to be given as a numerical value with the proper suffix in the text field available.

Once the user has finished choosing the options of his choice he can start the download by clicking the "DOWNLOAD" button. This button is present in the first window. So the user after choosing the advanced options has to "COME BACK" to the first window and then click the download button. On clicking the button the download begins. It can be seen that the download is performed as a background process and any sign of download taking place will not be seen. The user can now safely return to his job. The user can return once the download is complete. This application finally shows the time taken for the download is displayed.

5 FUTURE ENHANCEMENTS

This application is an interface to the wget tool. We have tried to implement almost all the options that are available with the wget tool. The options should satisfy the user's needs for a effective download. The user may not need more than these options. However more options can be added.

As said before after the download is complete the time taken for the download to complete is displayed. However no sign of download taking place is shown during the actual download. We are planning to keep a PROGRESS BAR that will indicate how far the download has taken place. The progress bar will show the progress in the download. One can thus know the percentage of download from the progress bar. This feature is yet to be implemented in getez.

More advanced options like a "DOWNLOAD ACCELARATOR" can also be implemented. Although there are as such no direct way we need to implement this feature using the other options that are available. The getez with the accelerator will be a full fledged download tool.

6 CONCLUSION

The getez application is an interface to the wget tool. In order to do this application a thorough study of the wget tool for download was done. The various options that are available with the wget tool are studied and they are implemented in this application.

The interface is designed using glade and the code for implementing the options are done using the gtk+. The implementation is mostly done using the C language. For this a study on how to use glade and interface it with gtk+ was also made.

Thus this application is an user friendly interface for the fastest and the most efficient download tool - wget.

7 AUTHORS

Senthil Kumaran S <stylesen@gmail.com>
Pre-Final Year, Bachelor of Engineering,
Department of Computer Science and Engineering,
Thiagarajar College of Engineering,
Madurai – 6250015
INDIA.

Sri Ranjani K <ranjani.14@gmail.com>
Pre-Final Year, Bachelor of Engineering,
Department of Computer Science and Engineering,
Thiagarajar College of Engineering,
Madurai – 6250015
INDIA.

getez@gmail.com

Feel free to post your doubts and queries to us!!!