

LIESH

Fco. Javier Bohórquez Ogalla

Índice

1. Descripción	3
2. Estructura de funcionamiento	3
2.1. Kernel LIESH	4
2.2. Programas escritos en LIESH	4
2.2.1. Sentencias básicas del lenguaje	5
2.2.2. Sentencias de comunicación	5
2.2.3. Procesamiento de datos	5
2.2.4. Almacenamiento de datos	5
2.2.5. Presentación de los resultados	6
2.3. Consola LIESH	6
2.4. Demonio LIESH	6
2.5. Screen LIESH	7
3. Bibliotecas dinámicas de LIESH	7
3.1. Biblioteca LIESH para el protocolo HTTP	8
4. Casos prácticos	8
5. Licencias	8
6. Herramientas de desarrollo	9

1. Descripción

LIESH es un software orientado a la explotación de servicios web basados en diversos protocolos (HTTP, FTP...). Con ayuda de LIESH el usuario podrá realizar tareas sobre distintos servicios de red y el contenido que estos ofrecen.

LIESH proporciona un lenguaje de programación interpretado y estructurado que facilitará al programador la realización de programas que trabajen sobre el contenido de distintos servicios de red.

Un programa escrito en LIESH se basará en:

- Comunicación y recopilación de datos desde servidores de red.
- Tratamiento y procesamiento de los datos para obtener información.
- Almacenamiento de los datos.
- Presentación de los resultados.

LIESH representa una capa de abstracción que facilita la comunicación y la obtención de información desde un servicio web.

Gran parte de la información que llega a las casas modernas lo hace a través de un PC conectado a internet. La web es un espacio no centralizado que ofrece multitud de servicios y que está en un constante cambio. LIESH nace de un intento por facilitar la obtención de información de la red, de la necesidad de automatizar tareas que periódicamente se realizan en internet, y del deseo de controlar un mundo que no para de cambiar.

2. Estructura de funcionamiento

LIESH se divide en cinco partes funcionales:

- Kernel o interprete
- Programas de usuario
- Consola
- Demonio
- Screen

2.1. Kernel LIESH

El kernel o interprete LIESH es la aplicación software encargada de interpretar instrucciones o comandos LIESH. Es la parte software más importante del conjunto.

El kernel se encargará de la administración y gestión de los datos y de las comunicaciones, y de realizar las acciones correspondientes a cada instrucción LIESH.

Es posible que en mismo sistema se den concurrentemente varios kernels en ejecución. En tal caso cada uno trabajará de forma independiente a los demás.

El usuario puede comunicarse con el kernel LIESH de varias formas:

- Ficheros de programas de usuario.
- Consola de comandos.
- Demonio.
- Screen.

2.2. Programas escritos en LIESH

LIESH proporciona una serie de símbolos y reglas sintácticas que forman un lenguaje de programación interpretado y estructurado. Haciendo uso de este lenguaje el usuario podrá obtener o enviar datos a un servidor de red basado en distintos protocolos (HTTP, FTP...). El lenguaje facilitará al usuario una serie de sentencias y estructuras para el procesamiento de estos datos y su almacenamiento. Además proporcionará distintos métodos para la presentación de los resultados en pantalla.

Las sentencias o instrucciones que forman un programa serán interpretados por el kernel o interprete de LIESH.

Un fichero de programa LIESH es un fichero formado por un conjunto de sentencias, instrucciones, estructuras y demás elementos del lenguaje LIESH. Estos ficheros pueden ser enviados al kernel para su ejecución de forma secuencial.

A continuación se presenta una clasificación de los conjuntos de sentencias que ofrece LIESH:

2.2.1. Sentencias básicas del lenguaje

LIESH proporciona al usuario sentencias y estructuras típicas de cualquier lenguaje de programación:

- Declaración de variables y constantes de diferentes tipos de datos.
- Declaración de expresiones regulares.
- Operaciones sobre distintos tipos de datos.
- Sentencias selectivas simples y múltiples.
- Sentencias iterativas.
- Definición de funciones y procedimientos.

2.2.2. Sentencias de comunicación

El lenguaje de programación LIESH debe dar soporte para la comunicación con distintos servicios de red. La cantidad de protocolos de comunicación a nivel de aplicación que existen hoy día hacen necesario que esta parte del lenguaje sea extensible y dinámica. Así el soporte para los distintos protocolos existentes se darán a partir de bibliotecas dinámicas que dotarán al lenguaje del potencial necesario para trabajar con el contenido que ofrecen los distintos servicios.

2.2.3. Procesamiento de datos

El procesamiento de los datos se hará a partir de las sentencias básicas del lenguajes y sentencias específicas para cada protocolo, ofrecidas a partir de las bibliotecas dinámicas.

2.2.4. Almacenamiento de datos

El almacenamiento de datos en un programa LIESH se podrá hacer de diferentes formas:

- En ejecución los datos se almacenarán en la memoria principal. Los datos en memoria serán administrados por el kernel LIESH por lo que una vez termine la ejecución de este estos se perderán.

- Los datos podrán ser guardados como ficheros ordinarios. Es por tanto que el lenguaje debe facilitar sentencias para el manejo de ficheros y el sistema de ficheros.
- Los datos podran ser almacenados en tablas de una base de datos, facilitando así el acceso y la agilizando la tarea de búsqueda. Así el lenguaje deberá facilitar la definición y el tratamiento de estructuras que representen a la base de datos.

2.2.5. Presentación de los resultados

Muchas de las sentencias e instrucciones que proporciona LIESH producen un resultado que podrá ser imprimido en pantalla o no, según la sintaxis utilizada por el programador.

LIESH ofrece una biblioteca dinámica y un software denominado Screen LIESH, que será una interfaz gráfica a la cual se podrá redireccionar los datos a imprimir.

2.3. Consola LIESH

El usuario puede comunicarse con el kernel de LIESH por medio de una consola de comandos que recibe instrucciones LIESH. La consola mantendrá una comunicación directa con el kernel, enviándole las instrucciones del usuario. El kernel interpretará la instrucción, realizará la acción correspondiente y en caso de que exista una salida se la enviará a la consola.

La consola LIESH puede ser ejecutada desde el interprete de comandos del sistema operativo.

2.4. Demonio LIESH

El demonio LIESH representa una forma de ejecutar determinados programas LIESH de forma automática.

Al iniciarse, esta aplicación leerá de un directorio una serie de programas LIESH definidos por el usuario. Por cada uno de ellos ejecutará un kernel LIESH independiente para su interpretación.

Cada fichero de programa LIESH podrá añadir directivas para el demonio. Estas directivas se encargan de programar al demonio LIESH para que mande al kernel a interpretar un conjunto de instrucciones cada cierto intervalo de tiempo.

2.5. Screen LIESH

Es un programa que presenta una interfaz gráfica con varios propósitos:

- Sirve como una interfaz gráfica amigable para que los programas LIESH impriman sus resultados.
- Pretende ser un mecanismo de entrada hacia los kernels LIESH en ejecución.
- Representa una forma de acceso a las distintas estructuras de datos que administran y gestionan los distintos kernels en ejecución.

Al iniciarse un kernel este podrá ser asociado a un proceso Screen LIESH, siendo posible asociar más de un kernel a un mismo proceso Screen.

Al lenguaje LIESH se añaden directivas que describen un sistema de interfaz gráfica que serán interpretado y presentado por la aplicación Screen LIESH.

Screen LIESH además implementa un sistema de ventanas emergentes que los programas LIESH pueden utilizar para mostrar información.

3. Bibliotecas dinámicas de LIESH

Como se ha comentado, la potencia y funcionalidad de LIESH puede extenderse por medios de bibliotecas dinámicas. Así existirán bibliotecas que añadan nuevas características al lenguaje, o que añadan una nueva forma de comunicación del kernel.

Por ejemplo, para añadir la capacidad de comunicación con servicios HTTP al lenguaje, se deberá instalar la biblioteca dinámica correspondiente.

Además muchos de los sistemas descritos en el punto anterior se implementan por medio de bibliotecas de este tipo.

Esto hace que el sistema software LIESH sea dinámico y modular, según las necesidades del usuario.

A continuación se muestra un ejemplo de biblioteca dinámica LIESH.

3.1. Biblioteca LIESH para el protocolo HTTP

Esta biblioteca añade funciones para la explotación del servicio HTTP. Algunas de las funcionalidades que presenta son:

- Proporcionar estructuras de datos que permitan mantener de forma óptima una página en HTML en memoria y acceder al contenido de la misma.
- Facilitar un conjunto de sentencias que permiten obtener datos de una página HTML de una forma inteligente y según diferentes criterios. Por ejemplo, se podría obtener todos los enlaces cuyo texto se corresponda con una expresión regular o buscar y descargar una imagen concreta dentro de la página.
- Permitir la comunicación con el servidor mediante métodos GET y POST propios del lenguaje HTML.
- Proveer de mecanismos para acceder a las páginas referenciadas por enlaces, permitiendo así la navegación dentro de un sitio web.

4. Casos prácticos

A continuación se ponen algunos ejemplos de casos prácticos en los cuales el sistema LIESH puede facilitar la tarea del usuario:

- Se desea descargar recursivamente todo el contenido de un servidor FTP.
- Se pretende realizar un programa que se conecte a una web HTTP con noticias de informática, se descargue cada noticia nueva que encuentre y mantenga una base de datos actualizada con todas las noticias.
- Se quiere un programa que se conecte a un repositorio SVN y avise si existen actualizaciones.
- Se necesita desarrollar un software que busque definiciones de palabras en un servidor de red por el protocolo DICT

En el desarrollo de programas LIESH es la creatividad y dedicación del usuario quienes ponen sus límites.

5. Licencias

LIESH es un software desarrollado con licencia GPL versión 3 de GNU.

6. Herramientas de desarrollo

Sistema operativo:	GNU/Linux
Lenguaje de programación:	C++
Compilador:	GCC
Depurador:	GDB
Biblioteca de interfaz gráfica:	Qt4
Biblioteca de protocolos y comunicaciones:	libCURL
Analizador léxico:	Flex
Analizador sintáctico y semántico:	Bison
Lenguaje de scripts:	Bash
Documentación:	L ^A T _E X, Doxygen
Planificación de proyecto:	Planner
Lenguaje de base de datos:	MySQL
Lenguaje de desarrollo web:	HTML, PHP, JavaScript