

# Distribución GNU/Linux para uso en los laboratorios de la carrera de Electrónica y Redes de Información

Giler E. Cristian, Santamaría S. Alexander y Mafla G. Enrique.

Ingeniería Electrónica y Redes de Información.  
Escuela Politécnica Nacional. Quito-Ecuador

**Resumen**— El presente proyecto describe la creación a partir de código fuente de una distribución GNU/Linux del tipo LiveCD adaptada a las necesidades de los laboratorios de computación de la carrera de Electrónica y Redes de Información.

El proyecto se compone de tres partes: El análisis de requerimientos y de la situación actual de los laboratorios respecto al sistema operativo basado en Linux. La descripción de todos los procesos, subprocesos y tareas involucrados en la implantación del sistema GNU/Linux. Las pruebas para verificar el funcionamiento del sistema.

El primer capítulo contiene el marco teórico respecto a los componentes principales de un sistema GNU/Linux. Consta además el análisis de la problemática actual de los laboratorios. Se utiliza la metodología ITIL para el análisis de situación actual y de requerimientos con sus respectivos resultados.

En el segundo capítulo se describen todos los procesos seguidos para la construcción del sistema y las utilidades que tiene el mismo. Además se incluye el proceso para la construcción del LiveCD como producto final basado en GNU/Linux, llamado TESIX-0.9.

En el tercer capítulo se incluyen todas las pruebas realizadas, divididas en 2 grupos: pruebas para verificar que se cumpla el alcance del proyecto y pruebas para verificar que se cumplan los requerimientos funcionales.

## I. INTRODUCCIÓN – METODOLOGÍA DEL PROYECTO

La metodología para el desarrollo del proyecto considera cinco etapas. A continuación se enumeran y describen las etapas a seguir:

**Análisis de la situación actual de los laboratorios--** Esta etapa contempla la realización de entrevistas. Las entrevistas se realizan a las personas encargadas del manejo de los laboratorios y a los docentes que hacen uso del sistema operativo Linux en los mismos. Por medio de las entrevistas se analiza la manera en que se gestionan los servicios de TI respecto del sistema operativo Linux en uso. Para este propósito se utiliza la metodología de ITIL (*Information Technologies Infrastructure Library*).

**Análisis de requerimientos del sistema--** Para el análisis de requerimientos del sistema se realizan encuestas a los profesores que utilizan Linux para impartir sus clases en los laboratorios de la carrera. También se realizan encuestas a una muestra significativa de los estudiantes de la carrera de Redes. El objetivo de la encuesta es obtener los datos necesarios para la construcción del sistema.

**Diseño del sistema--** En el diseño se describen: las características del núcleo a implantar, las bibliotecas básicas del sistema, y las capas que conforman el sistema. Además, en el diseño se presentan la estructura de directorios, con sus respectivos permisos y la metodología que se usa para instalar los paquetes de software.

**Construcción del sistema--** Una vez realizados los análisis de: situación actual y de requerimientos, se obtiene la información de qué utilitarios de sistema y de usuario serán parte del sistema operativo. Para ello se utiliza la sección

correspondiente al diseño del sistema como una guía de desarrollo.

La construcción del sistema se realiza sobre un entorno virtual. La máquina virtual usada para el fin mencionado es VMware®. La utilización del entorno de máquina virtual se justifica debido a la flexibilidad de poder pausar el proceso de creación del sistema en cualquier instante.

Una distribución anfitriona provee el software base necesario para la creación del sistema.

**Pruebas del sistema**-- Esta sección contempla las pruebas para verificar que se cumplan los requerimientos funcionales obtenidos de las encuestas de análisis de requerimientos. Para ello se comprobará el correcto funcionamiento de las herramientas que se incluyen en el sistema por cada materia que utiliza Linux. Además se realizan pruebas para corroborar el cumplimiento del alcance del proyecto.

## II. RESULTADOS DE LA ENCUESTA DE ANÁLISIS DE REQUERIMIENTOS

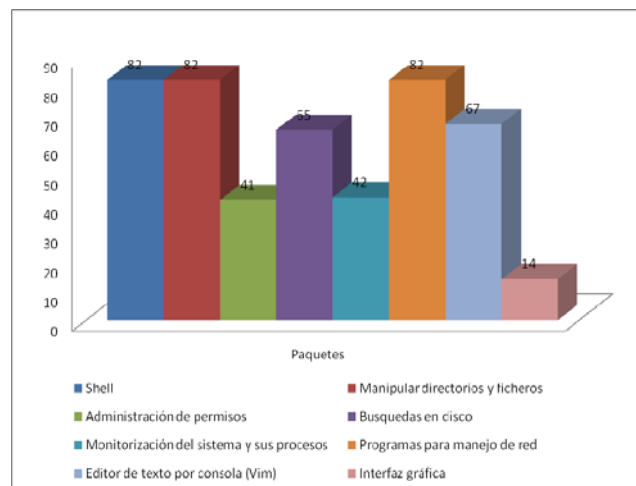
Considerando un nivel de confianza del 90% y un error muestral del 10%, la encuesta se la realizó a 82 personas, 77 estudiantes y 5 profesores.

Las siguientes preguntas reflejan los requerimientos básicos y funcionales para el sistema:

Respuesta a la pregunta 1. Las siguientes materias utilizan el sistema basado en Linux para sus clases prácticas:

- Sistemas Operativos
- Gestión de Redes
- Aplicaciones Distribuidas
- Seguridad de Redes
- Redes WLAN
- Hardware de Conectividad
- Redes e Intranets

Respuesta a la pregunta 7. Componentes básicos del sistema operativo basado en Linux.



**FIGURA 1. COMPONENTES BÁSICOS DEL SISTEMA OPERATIVO**

Como se puede observar comenzando por: Shell, programas para manipular ficheros y directorios, y programas para el manejo de red que fueron considerados como básicos por todos los encuestados. Todos los demás paquetes sugeridos que fueron tomados en cuenta por más del 50% de encuestados.

Debido al alto porcentaje de aceptación de los paquetes mostrados en el gráfico se considerará a todo el grupo dentro de la nueva distribución a excepción de la interfaz gráfica. Esta interfaz es utilizada necesariamente sólo por un paquete del análisis de requerimientos por lo que no será considerada en el sistema final.

Respuesta a la pregunta 8. Programas necesarios para impartir las clases prácticas en los laboratorios.

La siguiente es la lista recopilada de programas que los profesores consideran necesarios para sus clases prácticas. Todos estos programas serán incluidos en la nueva distribución como parte de los requerimientos funcionales de la misma.

Paquete	Uso	Materia
NetSnmpp	Todo el semestre	Gestión de Redes
Wireshark	Todo el semestre	Gestión de Redes
Traceroute	Todo el semestre	Gestión de Redes
Fping	Todo el semestre	Gestión de Redes
Nmap	Todo el semestre	Gestión de Redes
Java	1 clase	Sistemas operativos
NS-2	Todo el semestre	WLAN
Wireshark	Todo el semestre	Hardware de Conectividad
Apache	Todo el semestre	Hardware de Conectividad
FTP	Todo el semestre	Hardware de Conectividad
DNS	Todo el semestre	Hardware de Conectividad
Minicom	Todo el semestre	Hardware de Conectividad
OpenSSL	Todo el semestre	Seguridad de Redes
Kerberos	Todo el semestre	Seguridad de Redes
IPSec	Todo el semestre	Seguridad de Redes
OpenLDAP	Todo el semestre	Aplicaciones Distribuidas
Heartbeat	Todo el semestre	Aplicaciones Distribuidas
J2ee	Todo el semestre	Aplicaciones Distribuidas

**TABLA 1.** PAQUETES DE SOFTWARE REQUERIDOS POR CADA MATERIA

No se indican paquetes para las clases de la materia de Redes e Intranets. Debido a que los contenidos de la materia y de las clases prácticas de laboratorio consisten en aprender a instalar y configurar herramientas y software en sistemas basados en Linux.

### III. DISEÑO DEL SISTEMA

**Capas del Sistema--** En la figura 2 se tiene como base al núcleo del sistema Linux. Esta capa se encarga de gestionar el manejo de hardware como son: gestión de memoria, manejo de periféricos, dispositivos de entrada/salida, etc. La segunda capa contempla las bibliotecas del sistema, que se utilizan como interfaz de comunicación entre la tercera capa (la que interactúa con el usuario) y la primera capa (núcleo del sistema).

En la tercera capa se contemplan todos los utilitarios del sistema y del usuario, además de contener los compiladores del sistema.



**FIGURA 2.** CAPAS DEL SISTEMA OPERATIVO

**Kernel a Implantar--** El kernel Linux a implantar dentro del sistema GNU/Linux se caracteriza por tener un arquitectura monolítica. Este tipo de arquitectura concentra funcionalidades básicas de un sistema operativo (gestión de memoria, controladores de hardware, gestión de archivos, sistema de archivos, redes, etc.) dentro de un solo programa llamado kernel. Por esto, para realizar cambios o añadir nuevas funcionalidades dentro del kernel es necesario recompilarlo, y volverlo a cargar. El kernel no tiene la flexibilidad de realizar cambios en caliente (Hot Swap). Al poseer esta estructura monolítica, las funciones y componentes del núcleo tienen acceso directo a sus rutinas y estructuras de datos internas. El kernel Linux provee la capacidad de que el sistema sea multitarea y multiusuario.

**Módulos del Kernel--** Los módulos del kernel permiten brindar funcionalidad modular al sistema operativo GNU/Linux. A diferencia del kernel, el cual es de estructura monolítica, el sistema operativo en conjunto brinda funcionalidad modular al usuario.

Esto implica que se va a tener la facultad de cargar y descargar secciones arbitrarias de código (módulos) cuando sea necesario hacerlo. El sistema continuará funcionando sin interrupciones al cargar un módulo.

Los módulos serán cargados en el desarrollo del LiveCD. Estos módulos son los correspondientes a: Aufs, Squashfs y Lzma.

**Compiladores y Librerías--** Los compiladores son programas que permiten traducir el código fuente de un programa a lenguaje de máquina. Todos los programas, incluidos los del sistema y los programas pedidos en el análisis de

requerimientos, deben ser compilados antes de ser instalados. Con base en el lenguaje de programación en el cual se distribuyen los programas, para el proyecto se implementa el paquete GCC, el cual incluye un compilador de lenguaje C y C++.

**Estructura de directorios--** Dentro del diseño del sistema GNU/Linux se sigue el estándar FHS (Filesystem Hierarchy Estándar). Esta norma define los directorios principales y sus contenidos en todo sistema operativo GNU/Linux. Cabe recalcar que existen programas que instalan sus propios directorios. Todos los directorios serán de lectura y escritura (rw) en el LiveCD. Físicamente ningún directorio se puede modificar debido a que el entregable al estar dentro de un CD es de sólo lectura.

**Paquetes de Software--** En el diseño se sigue la metodología LFS para construir un sistema GNU/Linux base. El orden de instalación y los paquetes a instalarse tienen como referencia la metodología LFS. Adicional a los paquetes necesarios se instalarán paquetes que complementen la funcionalidad del sistema. Un ejemplo de esto es un cliente DHCP. Para finalizar la construcción del sistema se instalarán los paquetes pedidos en el análisis de requerimientos.

#### IV. CONSTRUCCIÓN DEL SISTEMA

Para construir el sistema se utiliza la metodología LFS en su versión 6.3 (Linux from Scratch). Esta metodología permite instalar y configurar un sistema Linux a partir de código fuente. La metodología divide a la construcción del sistema en 2 etapas. En primer lugar se creará un sistema Linux base (LFS). El sistema base contiene: kernel, compilador de C, y terminal de consola. Permite la ejecución de comandos como por ejemplo: comandos para manipular directorios y ficheros, para monitorizar el sistema y sus procesos, para manejo de red, etc. Sobre este sistema base corren todas las herramientas requeridas por los profesores.

En la segunda etapa de la metodología se añadirá el software complementario. Se entiende así por ejemplo: los paquetes requeridos para brindar la funcionalidad deseada y scripts de configuración de inicio del sistema. Esta segunda parte tiene como referencia la metodología BLFS

6.3 (Beyond Linux from Scratch). Se consigue un sistema funcional y a la medida de la mayoría de los requerimientos del laboratorio.

Finalmente se crea el LiveCD de la distribución construida. Para la creación del LiveCD se sigue la metodología descrita en el foro de la página web: [www.linuxlive.org](http://www.linuxlive.org). Esta página detalla los pasos a seguir para la creación del LiveCD a partir de una distribución ya instalada. Se hace uso de los scripts publicados por Linux-Live.

#### V. CONSTRUCCIÓN DEL SISTEMA BASE

Para la construcción del sistema base se llevan a cabo los siguientes procesos:



FIGURA 3. PROCESOS PARA CREAR EL SISTEMA BASE

##### 1.- Preparativos para la construcción

Los subprocesos involucrados en los preparativos para la construcción son:



FIGURA 4. SUBPROCESOS INVOLUCRADOS EN LOS PREPARATIVOS PARA LA CONSTRUCCIÓN

**Particionar el disco--** La partición albergará al nuevo sistema. El LiveCD no requiere instalación nativa. Se creará una partición de intercambio y otra partición para el sistema a crear. Para la partición del sistema LFS se va a utilizar un

espacio de 8GB. Este espacio incluye la partición de intercambio. LFS recomienda hasta 3GB.

**Creación el sistema de archivos--** Se debe formatear la nueva partición. Esto se hará con el sistema de archivos de tipo ext3. También se debe iniciar y activar la partición de intercambio, partición en la que se cargan imágenes de procesos que no están en memoria física.

**Montar la partición--** Se debe montar la partición. Con esto se consigue hacerla accesible. Para esto se debe escoger un punto de montaje. Esto significa un directorio en particular para montarlo.

**Creación de directorios para almacenar fuentes e instalaciones--** A lo largo de todo el desarrollo de la primera parte se utilizan dos directorios en especial. Un directorio es para la instalación de todas las herramientas temporales (*/mnt/lfs/tools*). El otro directorio sirve para guardar todas las fuentes del proyecto (*/mnt/lfs/sources*).

**Creación de un grupo y un usuario--** Por cuestiones de seguridad y para manejo de credenciales de seguridad y permisos se deben crear al menos un usuario y un grupo diferente al del *root*.

## 2.- Construcción del sistema temporal

En esta parte se instalan y configuran las herramientas que conforman un entorno de desarrollo. Este entorno será independiente de la distribución anfitriona. Sólo habrá dependencia del núcleo. Los subprocesos involucrados en la construcción del sistema temporal son:



**FIGURA 5.** SUBPROCESOS INVOLUCRADOS EN LA CONSTRUCCIÓN DEL SISTEMA TEMPORAL

**Instalación de Binutils--** Binutils es un paquete que tiene una colección de herramientas binarias. Se destacan *ld*, el enlazador de GNU y *as*, el ensamblador de GNU.

**Instalación de GCC--** El GCC es el paquete que contiene los compiladores de GNU. Se incluyen los compiladores de C y C++. Los compiladores mencionados son necesarios debido a que los siguientes paquetes serán instalados después de compilar sus códigos fuente.

**Instalación de Glibc--** Glibc contiene la librería principal de C. Esta librería proporciona rutinas para actividades como: búsqueda de directorios, creación, eliminación, lectura de ficheros, manejo de patrones, de operaciones aritméticas, entre otras.

**Ajustar las herramientas--** Es necesario realizar ajustes para asegurar que la búsqueda y el enlazado tengan lugar solamente dentro del directorio */tools*. Así mismo, se instala un *ld* ajustado que limita su búsqueda interna a */tools/lib*.

**Herramientas para ejecutar los bancos de pruebas--** Se deben ejecutar los bancos de pruebas que vienen con las herramientas. Esto sirve para asegurar que las herramientas están instaladas correctamente. El propósito de la instalación de Tcl, Expect y DejaGNU es poder ejecutar los bancos de pruebas de GCC y Binutils.

**Instalación y configuración de herramientas para brindar mayor funcionalidad--** Luego de instalar y probar GCC y Glibc, es necesario instalar paquetes adicionales para aumentar la funcionalidad del sistema temporal. Estos paquetes proporcionan todas las herramientas necesarias para construir el sistema básico. El orden de instalación y los paquetes a instalarse son especificados por LFS. Estas herramientas son *bash*, *bzip2*, *coreutils*, *diffutils*, *findutils*, *make*, *patch*, *perl*, *tar*, *sed*, *util-linux*, etc.

## 3.- Construcción del sistema base

En esta sección se comienza la construcción del sistema definitivo. El orden de instalación sigue el establecido por la metodología. Los subprocesos involucrados en la construcción del sistema base son:



**FIGURA 6. SUBPROCESOS INVOLUCRADOS EN LA CONSTRUCCIÓN DEL SISTEMA BASE**

### Preparativos para la construcción

Las tareas involucradas en cada subproceso son las siguientes:

**Preparar los sistemas de ficheros virtuales del kernel--** Se exportan ficheros para la comunicación desde y hacia el núcleo. Estos ficheros son virtuales. Esto quiere decir que no usan espacio en el disco. El contenido de estos ficheros reside en la memoria. Estos ficheros se montarán sobre: `/dev`, `/proc` y `/sys`.

**Poblar /dev--** Linux accede a los dispositivos del sistema (hardware) a través de los nodos de dispositivos. Estos nodos se encuentran en el directorio `/dev`. Cuando arranca el sistema operativo necesita la presencia de estos nodos. En particular necesita los nodos `console` y `null`. Estos deben ser creados para que el sistema los encuentre en su arranque.

**Ingreso al entrono chroot --** En este entorno se comienza la instalación del sistema final. En este entorno solamente se encuentran las herramientas temporales. Toda la construcción del sistema básico se realizará en este entorno. Esto se hace para usar las herramientas temporales creadas anteriormente.

**Creación de los directorios--** El árbol de directorios está basado en FHS (Filesystem Hierarchy Standard). Sólo se crearán los directorios necesitados por el sistema. Los directorios se crean por defecto con permisos 755. El propietario tiene permisos para leer, escribir y ejecutar; el grupo y otros usuarios sólo de leer y ejecutar. Otros como el directorio `/root` tienen permisos 1777, es decir el propietario, el grupo y todos los usuarios pueden leer, escribir y ejecutar. El 1 al inicio es para que un usuario no

pueda borrar o renombrar ficheros de otros usuarios.

**Creación de ficheros necesarios--** Linux mantiene una lista de ficheros montados en `/etc/mtab`. No se montará ningún sistema de ficheros. Se crea este fichero para las utilidades que esperan que se encuentre presente. Para que se reconozca el nombre "root" se debe incluir entradas en los ficheros `/etc/passwd` y `/etc/group`. `/etc/passwd` determina quién puede acceder al sistema y qué puede hacer dentro del sistema. `/etc/group` contiene atributos de grupos. Se deben también crear los ficheros de registro. Estos ficheros abarcan información de los usuarios que están y estaban en el sistema.

### Paquetes del sistema base

En esta sección se describen paquetes que forman parte del sistema definitivo.

Paquete	Utilidad
Man-pages	Contiene páginas de manual.
Glibc	Contiene la principal librería de C.
Binutils	Posee utilidades para trabajar con ficheros objeto. Contiene el enlazador, así como el ensamblador.
GCC	Contiene la colección de compiladores de GNU. Incluye los compiladores de C y C++.
BerkeleyDB	Contiene utilidades y funciones utilizadas para aplicaciones que trabajan con bases de datos
Sed	Contiene un programa para editar flujos de texto.
Coreutils (GNU Core Utilities)	Contiene herramientas para utilidades de ficheros (fileutils), utilidades de intérpretes de comandos (shellutils) y utilidades de proceso de textos (textutils).
M4	Contiene un procesador de macros
Bison	Es un generador de parsers (analizador sintáctico) de propósito general.
Ncurses	Es una biblioteca de programación.
Procps	Contiene programas para monitorizar procesos.
Libtools	Provee un guión de GNU para dar soporte generalizado para la compilación de librerías.
Perl	Este paquete contiene al lenguaje de programación PERL (Lenguaje Práctico de extracción e Informe).
Readline	Este paquete contiene librerías con funciones para edición en línea de comandos y capacidades de historial
Zlib	Este paquete contiene librerías con funciones para compresión y descompresión.
Autoconf	Produce shellscrips. Estos automatizan la configuración de paquetes desde código fuente.
Automake	Es una herramienta para generar automáticamente archivos makefile
Bash	Es el intérprete de comandos. Contiene el

	Bourne-Again Shell
Bzip2	Contiene programas para comprimir y descomprimir ficheros.
Diffutils	Es un paquete que tiene programas para encontrar diferencias entre archivos.
File	Contiene una utilidad para determinar tipos de ficheros
Findutils	Este paquete tiene programas que permiten encontrar ficheros
Flex	Contiene un programa que reconoce patrones de texto.
GRUB	Es el gestor de arranque del sistema. Este paquete contiene el Grand Unified Bootloader.
Gawk	contiene programas para manipular ficheros de texto
Gettext	Contiene utilidades para la internacionalización y localización.
Grep	Contiene programas para realizar búsquedas dentro de ficheros.
Groff	Contiene programas para procesar y formatear texto.
Gzip	Contiene programas para comprimir y descomprimir ficheros.
Inetutils	Contiene utilidades para trabajo en la red.
Kbd	Este paquete contiene ficheros para mapas de teclado y utilidades para el teclado.
Make	Contiene un programa para compilar paquetes.
Man-DB	Contiene programas para encontrar y visualizar páginas de manual.
Module-Init-Tools	Contiene programas para manejar módulos del kernel.
Patch	Contiene un programa que permite crear o modificar ficheros mediante la aplicación de un archivo parche.
Psmisc	Los programas contenidos en este paquete muestran información de procesos que se encuentran en ejecución
Shadow	El paquete shadow contiene programas para manejar contraseñas seguras.
Sysklogd	Sus programas registran mensajes del sistema.
Sysvinit	Este paquete contiene programas para controlar el arranque, ejecución y cierre del sistema.
Tar	Contiene un programa de archivado.
Texinfo	Este paquete contiene programas para leer, escribir y convertir páginas info.
Udev	Contiene programas para la creación dinámica de nodos de dispositivo.
Util-Linux	Contiene utilidades del sistema. Estas utilidades permiten manejar sistemas de ficheros, consolas, particiones, etc.
Vim	Contiene un editor de texto vía consola.

**TABLA 2. PAQUETES DE SOFTWARE DEL SISTEMA BASE**

### Liberación de espacio

Se eliminarán símbolos de depuración. Esos símbolos aparecen al compilar librerías y

programas. Los símbolos hacen más grandes a los programas y librerías. La diferencia en tamaño entre programas con y sin símbolos de depuración es generalmente de 5 a 2. La depuración se hace con la ayuda del comando strip.

### Hacer arrancable el sistema

En primer lugar se implanta un sistema de inicio basado en *sysvinit*. Este posee 7 niveles de ejecución de 0 a 6. Los niveles indican al sistema que hacer al arrancar. Por defecto es el nivel 3 (modo multiusuario con red). Los niveles se encuentran en */etc/rcd/rc#*.

*Udev* se encarga de la creación de dispositivos o nodos en el arranque del sistema. Estos se crean en el directorio */dev*. Los dispositivos se crean estáticamente si existen o no físicamente.

Se deben crear varios guiones o archivos para el arranque del sistema:

Archivo	Contenido
<i>/etc/sysconfig/console</i>	El guión representa la disposición de teclas del teclado.
<i>/etc/sysconfig/network.</i>	Aquí se especifica el nombre del host
<i>/etc/hosts</i>	En este se asigna una Ip a la computadora, un dominio y un alias.
<i>/etc/sysconfig/network/netw ork-devices/ifconfig.eth#</i>	Es el archivo que tiene toda la configuración IPv4
<i>/etc/resolv.conf</i>	En este archivo consta la o las direcciones IP del o los servidores DNS.
<i>/etc/fstab</i>	Es la tabla de sistemas de archivos. Es utilizado por la mayoría de programas del sistema Linux para conocer dónde deben montarse los sistemas de archivos.

**TABLA 3. ARCHIVOS PARA EL ARRANQUE DEL SISTEMA**

Posteriormente se configurará y compila el kernel del sistema desde su código fuente.

Para preparar la compilación del kernel se realizan los siguientes pasos:

```
make mkproper
```

Para limpiar el árbol del núcleo. Recomendación por parte de los desarrolladores del kernel

```
make menuconfig
```

Permite ingresar a un menú gráfico más amigable para la configuración de compilación del kernel.

```
make
```

Compila la imagen del núcleo, así como sus módulos.

```
make modules_install
```

Permite instalar los módulos creados en el paso anterior.

Luego de completar la compilación es necesario copiar los nuevos archivos a la partición del nuevo GNU/Linux. Esto se lo realiza con los siguientes comandos:

```
cp arch/i386/boot/bzImage /boot/kernel-2.6.22.5
cp System.map /boot/System.map-2.6.22.5
cp .config /boot/config-2.6.22.5
```

Donde *bzImage* es la nueva imagen del kernel compilado.

*System.map* es un archivo de símbolos para en kernel. Permite realizar la traslación de las funciones en el API del kernel, así como las direcciones de las estructuras de datos del kernel para el kernel en ejecución.

Por último, el archivo *.config* contiene la configuración del núcleo creado.

Finalmente para que el nuevo sistema sea arrancable se debe contar con un gestor de arranque. Apegándose a la metodología LFS se toma el gestor GRUB.

Se inicia el intérprete de comandos de GRUB por medio del comando *grub*. Se debe crear un archivo que brinde la configuración del menú de arranque del GRUB. El archivo se ubica en el directorio */boot/grub/menu.ls*.

Luego de desmontar los sistemas de archivos virtuales (*/mnt/lfs/dev/pts*, */mnt/lfs/dev/shm*, */mnt/lfs/dev*, */mnt/lfs/dev/proc*, */mnt/lfs/dev/sys*) se reinicia el sistema

## VI. INSTALACIÓN DE HERRAMIENTAS REQUERIDAS Y COMPLEMENTARIAS DEL SISTEMA

En esta sección se describirán todos los procesos a seguir después de que el sistema básico arranque por primera vez. Esto corresponde a la segunda etapa de la metodología LFS. Estos paquetes no formaron parte del sistema Linux básico creado. A continuación se pueden observar los procesos involucrados



**FIGURA7.** SUBPROCESOS INVOLUCRADOS EN LA INSTALACIÓN DE HERRAMIENTAS REQUERIDAS Y COMPLEMENTARIAS

**Instalación de paquetes complementarios del sistema**— Estos paquetes instalan herramientas que complementan la funcionalidad del sistema como lo son: *aufs*, *pciutilities*, *usbutilities* que son herramientas instaladas brindan transparencia al usuario para el manejo de periféricos. Además existen otras herramientas básicas para el uso de la red e internet como lo son: *dhcpcd* y *wget*.

**Scripts de configuración de archivos de inicio**—Se crean los archivos de configuración de la siguiente tabla:

Archivo	Contenido
<i>/etc/profile</i>	En este archivo de configuración se especifica funciones de configuración para el manejo del PATH de trabajo para los usuarios del sistema operativo
<i>/etc/bashrc</i>	En este archivo se configura los colores del <i>prompt</i> del <i>bash</i>
<i>~/.bash_profile</i>	Contiene información de configuración del <i>bash</i> exclusiva para cada usuario.



**TABLA 4. SCRIPTS DE CONFIGURACIÓN DE ARCHIVOS DE INICIO**

**Instalación de paquetes pedidos en el análisis de requerimientos**— Se instalan todos los paquetes de la tabla 1 a excepción de NS-2.

## VII. CREACIÓN DEL LIVECD

Apegándose a las recomendaciones de la comunidad de BLFS se usa los scripts y recomendaciones publicadas en [www.linux-live.org](http://www.linux-live.org). Los procesos involucrados son los siguientes:



**FIGURA 8. PROCESOS INVOLUCRADOS EN LA CREACIÓN DEL LIVECD**

**Reemplazo del kernel**— Es necesario tener (como prerequisite) construidos los paquetes de sistemas de archivos *aufs* (*another union file system*) y *squashfs*. Ambos paquetes deben estar implementados como módulos del kernel y no como parte del kernel. Adicionalmente el paquete *squashfs* debe estar parchado con el módulo de soporte para LZMA (*Lemple-Ziv-Markov Algorithm*) que es un algoritmo de compresión de datos.

Se encontró que es necesario cambiar el kernel del sistema operativo de su versión 2.6.22.5 a la 2.6.27.27. Esto se debe a que el kernel 2.6.22.5 no soporta *squashfs* + *lzma* cuando se lo implementa como un módulo del kernel.

**Configuración del kernel**— Luego de reemplazar el Kernel se lo debe compilar para que soporte las funciones y módulos necesarios para crear el LiveCD. La metodología indica que las siguientes funciones deben ser compiladas directamente en el kernel:

```

ext2 (CONFIG_EXT2_FS=y)
tmpfs (CONFIG_TMPFS=y)
ramdisk (CONFIG_BLK_DEV_RAM=y)
initrd (CONFIG_BLK_DEV_INITRD=y)
  
```

Los siguientes módulos deben ser implementados fuera del kernel:

```

loop.ko
isofs.ko
aufs.ko
squashfs.ko
sqlzma.ko
unlzma.ko
  
```

**Liberación de espacio**— Se procede a liberar espacio de disco por medio de la ejecución del comando *compressdoc*. Esto hace que la distribución LiveCD sea lo suficientemente compacta para que quepa en un CD.

**Construcción de datos**— La construcción de los datos se la hace antes de la creación de la imagen .ISO del LiveCD. Se procede a ejecutar el script de construcción y compresión de los directorios que conformarán el LiveCD.

**Creación del ISO**— Un script crea el archivo *.iso del sistema*

## VIII. REFERENCIAS BIBLIOGRÁFICAS

LIBROS
SILBERSCHATZ, Abraham; GALVIN, Peter; "Sistemas Operativos"; Quinta Edición; Pearson Educación; 1999.
STALLINGS, William; "Sistemas Operativos"; Segunda Edición; Prentice Hall; 1997.
CARRETERO, Jesús; GARCÍA, Felix; PÉREZ, Fernando; DE MIGUEL, Pedro, "Sistemas Operativos Una Visión Aplicada"; McGraw Hill; 2001
BEEKMANS, Gerard; "Linux from Scratch"; Versión 6.3; Agosto 2007.
BEEKMANS, Gerard; BLFS EQUIPO DE DESARROLLO; "Beyond Linux from Scratch"; Versión 6.3; Agosto 2008.
KROAH-HARTMAN, Greg; "Linux Kernel in a Nutshell"; 1st. Edition; O'Reilly; USA, Diciembre 2006.
GARCÍA, Luis Martin; "¿Cómo compilar el Kernel Linux paso

<i>a paso?"; Versión 0.3; España, Marzo 2006.</i>
MOTAB, Islam; " <i>LiveCD Crash Course</i> "; Aka Phaeronix; Noviembre 2006
<b>INTERNET</b>
ITIL. <a href="http://itil.osiatis.es">http://itil.osiatis.es</a>
TAMAÑO DE MUESTRA. <a href="http://www.feedbacknetworks.com/cas/experiencia/solpreguntar-calculador.htm">http://www.feedbacknetworks.com/cas/experiencia/solpreguntar-calculador.htm</a>
HEARTBEAT. <a href="http://www.linux-ha.org/">http://www.linux-ha.org/</a>
SLAX LIVECD. <a href="http://www.linux-live.org/">http://www.linux-live.org/</a>
LIVECD <a href="http://www.mirphakblog.com.ar/?q=node/38">http://www.mirphakblog.com.ar/?q=node/38</a>
SQUASHFS. <a href="http://www.squashfs-lzma.org/sqlzma.txt">http://www.squashfs-lzma.org/sqlzma.txt</a>
AUFS. <a href="http://aufs.cvs.sourceforge.net/viewvc/aufs/aufs/">http://aufs.cvs.sourceforge.net/viewvc/aufs/aufs/</a>
LINUX KERNEL. <a href="http://www.kernel.org/">http://www.kernel.org/</a>
NET_SNMP. <a href="http://www.net-snmp.org/docs/man/">http://www.net-snmp.org/docs/man/</a>
FPING. <a href="http://fping.sourceforge.net/">http://fping.sourceforge.net/</a>
IPSEC. <a href="http://www.ipsec-howto.org/ipsec-howto.pdf">http://www.ipsec-howto.org/ipsec-howto.pdf</a>
<b>TESIS</b>
Barragán, Santiago; " <i>Análisis Diseño y Prototipo de un servicio LDAP para la Polired</i> "; 2007.

## IX. BIOGRAFÍAS

### **Cristian O. Giler Egúez**



Nació en Quevedo el 21 de octubre del 1981. Realizó sus estudios secundarios en el Colegio Nacional "Nicolás Infante Díaz" y los finalizó en el año 1999. En el año 2000 ingresó a la Escuela Politécnica Nacional; actualmente se encuentra culminando su carrera en Ingeniería Electrónica y Redes de Información.

### **Alexander W. Santamaría Salvador**



Nació en Quito el 4 de Octubre del 1985. Realizó sus estudios secundarios en el Colegio Municipal "Sebastián de Benalcázar" y los finalizó en el año 2004. En el mismo año ingresó a la Escuela Politécnica Nacional; actualmente se encuentra culminando su carrera en Ingeniería Electrónica y Redes de Información.

### **Dr. Enrique Mafla Gallegos**



Ing. en Meteorología y MSc en Geografía, Instituto de Meteorología, Odesa, Ucrania. MSc en Informática, Escuela Politécnica Nacional, Quito, Ecuador. PhD y MSc Ciencias de la Computación, Purdue University, W. Lafayette, IN, E.U. Actualmente trabaja como profesor en las facultades de Ingeniería en Electrónica y Redes de Información y de Ingeniería de sistemas de la EPN.