Mantener el reloj del sistema

Peso	3
Tópico Cubierto	108.1 Mantener el reloj del sistema
Descripción	Los candidatos deberán ser capaces de mantener apropiadamente el reloj del sistema y sincronizar el reloj vía NTP
Temas	* Establecer el reloj del hardware al horario correcto UTC. * Configurar correctamente la zona horaria * Configuración básica de NTP * Conocimiento del uso del servicio pool.ntp.org
Ejemplos	* /usr/share/zoneinfo * /etc/timezone * /etc/localtime * /etc/ntp.conf * date * hwclock * ntpd * ntpdate * pool.ntp.org

Peso: Indica el valor de importancia que tiene este tópico en la certificacion.

Tópico Cubierto: Indica según el programa de certificacion LPI que tópico le corresponde a este tema.

Descripción: Un resumen de lo que se verá.

Temas: Un resumen de los conceptos primordiales que están cubiertos.

Ejemplos: Palabras claves que se tienen que tener en cuenta.

Introducción

En el siguiente tópico se introducirá acerca del uso del servicio NTP y sus variantes. Este servicio me permitirá establecer de manera correcta la hora del sistema.

Administrando la hora del equipo

En este capítulo aprenderemos a configurar la hora del sistema mediante el uso de servidores de tiempo NTP.

Reloj de tiempo bajo linux

Linux utiliza dos relojes:

Reloj de Software

Reloj de Hardware

Reloj de Software

Un equipo personal tiene un reloj de hardware alimentado por una batería. Esa batería asegura que el reloj continúe trabajando aún cuando la computadora se encuentre sin suministro eléctrico.

El reloj de hardware puede ser modificado (o definido) desde la pantalla de configuración de la BIOS o desde cualquier sistema operativo.

El kernel Linux mantiene la fecha y hora de manera independiente al reloj de hardware. Durante el inicio de un sistema Linux, el kernel configura su propio reloj de software accediendo a la fecha y hora mantenida por el reloj de hardware. Luego, ambos relojes trabajan independientemente.

Linux mantiene su propio reloj debido a que leer el reloj de hardware constantemente es lento y complicado. El reloj del kernel siempre muestra la hora universal, por lo que no necesita conocer cómo utilizar husos horarios.
La simplicidad de este modo de trabajar proporciona alta confiabilidad y facilita actualizar la información de la zona horaria. Cada proceso realiza las conversiones de zona horaria de manera independiente.
El reloj de hardware puede estar en formato de hora local u hora universal. Usualmente es mejor que el reloj de hardware mantenga la hora universal, porque de esta manera no será necesario modificar la hora del reloj cuando el horario de verano empiece o finalice.

Programas para controlar el reloj del sistema

Bajo linux existen dos herramientas principales que son implementadas para administrar el reloj hardware y el de software del sistema, estas herramientas son:

hwclock Controla el reloj de hardware

date Controla el reloj del sistema

Ajustes de tiempo y de las zonas horarias

Hay 2 formas estándar para ajustar el reloj de un equipo

localtime Por ubicación geográfica (Hora Local)

UTC El tiempo universal coordinado, o UTC, en español, también conocido como tiempo civil, es el tiempo de la zona horaria de referencia respecto a la cual se calculan todas las otras zonas del mundo

Ajuste de la hora en Linux

El procedimiento es relativamente simple:

Primero consiste en ajustar el reloj de Hardware a través de la configuración del BIOS

El segundo consiste en establecer la variable de entorno **TZ** a la zona horaria adecuada utilizando el siguiente comando

 # tzselect

Alternativa al paso 2 Utilice el programa **tzconfig** que establecerá un enlace simbólico de la siguiente forma:

 # ln -s /usr/share/zoneinfo/Mexico/General /etc/localtime

El tercero consiste en decirle a Linux que nuestro reloj de hardware y de sistema corre bajo **UTC**, para ello teclee lo siguiente:

 # hwclock --utc --hctosys

Comando hwclock

Hwclock es una herramienta que nos permite:

- Acceder al Reloj del Hardware
- Mostrar la hora actual
- Poner el Reloj del Hardware a una hora especificada
- Poner el Reloj del Hardware a la Hora del Sistema
- Poner el Tiempo del Sistema desde el Reloj del Hardware.

también puede ejecutar **hwclock** periódicamente para insertar o quitar tiempo del Reloj del Hardware para compensar desviaciones sistemáticas en las que el reloj gana o pierde tiempo consistentemente a una cierta velocidad si se deja solo.

Sintaxis:

 # hwclock [opciones]

Opciones	**Descripción**
--show	Lee el reloj del hardware y muestra la hora en la salida estándar.
--set	Pone el reloj del hardware a la hora dada por la opción –date
--hctosys	Pone el tiempo del sistema a partir del reloj del hardware. Ésta es una buena opción para poner en

uno de los guiones de arranque del sistema.

--systohc Pone el reloj del hardware a la hora del sistema actual.

--adjust Añade o sustrae tiempo del reloj del hardware para tener en cuenta el desvío sistemático desde la última vez que el reloj se puso o se ajustó. Vea la discusión al respecto más adelante.

--utc Indica que el reloj del hardware se mantiene en el tiempo universal coordinado (UTC). Es cosa suya si mantiene su reloj en hora local o UTC, pero nada en el reloj le dice qué es lo que ha escogido. Así que con esta opción es como le da esa información a hwclock.

Comando date

Este programa se utiliza para mostrar o establecer la hora del sistema

Sintaxis:

 # date [opciones] [+FORMATO]

Formato:

Opciones	**Descripción**
Una literal %\\	
%a	Localización del nombre abreviado del día de la semana (Sun,Sat)
%A	Localización del nombre completo del día de la semana (Sunday,Saturday)
%b	Localización del nombre abreviado del nombre del mes
%c	Localización de fecha y hora (Sat Nov 04 12:02:33 EST 1989)
%C	Siglo (año dividido por 100 y truncado a un entero) 00-99
%d	Dia del mes (01..31)
%D	Fecha (mm/dd/yy)
%e	Día del mes (1..31,), a diferencia de %d este suprime el espacio en blanco
%F	Lo mismo que %Y-%m-%d
%h	Lo mismo que %b
%H	Hora (00..23)
%I	Hora (01..12)
%j	Dia del año (001..366)
%k	Hora (0..23)
%l	Hora (1..12)
%m	Mes (01..12)
%M	Minutos (00..59)
%n	Una nueva línea
%N	Nanosegundos (000000000..999999999)
%p	Indicación de PM o AM en mayúsculas
%P	Indicación de pm o am en minúsculas
%r	Tiempo en 12 horas (hh:mm:ss AP M)
%R	Tiempo en 24 horas (hh:mm)
%s	Segundos desde `00:00:00 1970-01-01 UTC' (una extension de GNU)
%S	Segundos (00..60);
%t	Un tabulador horizontal
%T	Tiempo , 24 horas (hh:mm:ss)
%u	Dia de la semana (1..7) el dia 1 representa el lunes

%U Número de semana del año con el domingo como primer día de la semana (00..53)

%V Número de semana del año con el lunes como primer día de la semana (00..53)

%w Dia de la semana (0..6); 0 representa el Domingo

%W Número de la semana del año con Lunes como primer dia de la semana (00..53)

%x Representación de la fecha local (mm/dd/yy)

%X Representación de la hora local (%H:%M:%S)

%y Los últimos dos dígitos del año (00..99)

%Y Año (1970..)

En el siguiente ejemplo se muestra como poder transformar una hora que esta en otro formato

$ date

dom ene 22 18:45:27 MST 2012

$ date -d '2012-01-26 1730 UTC'

jue ene 26 10:30:00 MST 2012

$ date -d '2012-01-26 1730 ART'

jue ene 26 13:30:00 MST 2012

Ntp

Una razón importante para tener un sistema de reloj que me permita sincronizar el tiempo entre todos nuestros equipos para poder tener bien controlado cuando sucedido cada evento cuando se generan los logs. Otro ejemplo podría ser cuando varios equipos usan un recurso compartido con NFS por ejemplo. Para esto se usa —> Network Time Protocol (NTP) http://www.ntp.org.

Los servidores de NTP proporcionan servicios de sincronización al cliente que se conecte a ellos, y ellos mismos se sincronizan gracias a servidores superiores de tiempo. Las capas en este modelo se llaman strata, con el nivel superior , stratum 0, consiste en un hardware de hora dedicado, como pueden ser relojes atómicos o receptores por satélite. Lo servidores conectados a estas fuentes de hora stratum 0 se llaman servidores stratum 1. Los servidores que se sincronizan por los servidores stratum 1 son lo servidores stratum 2, y así con el resto. [2]

Los servidores NTP se pueden utilizar de dos formas. Una es ejecutando una utilidad cliente llamada ntpdate, que sincroniza el reloj del sistema una vez. La otra forma es ejecutar un servicio NTP que sincroniza automáticamente el reloj del sistema en cuanto su hora no sea la correcta. Hay una gran cantidad de sistemas que utilizan ambos métodos. Si el reloj del sistema y el reloj atómico difieren mucho, puede llevarle al sistema un tiempo el sincronizarse con un servidor de tiempo superior. Para solventar esto, se llama a la utilidad ntpdate y se sincroniza el reloj antes de iniciar el servicio NTP.

Comando ntpdate

Opciones

-b Con esta opción, la hora del sistema se establece en lugar de ser poco a poco ajustado, no importa cuán lejos de la hora local es.

-d Modo debug donde vemos mucha info pero la hora no se ajusta

-p n El número de muestra que va a capturar para setear la hora, (default 4)

-q Realizar una consulta a un servidor

-s La salida la envía al syslog y no la salida estándar.

-t n Timeout de respuesta

-u Especificar qué puerto udp va utilizar

-v Verbose mode

-B Toma más tiempo para sincronizar nuestro reloj.

 # ntpdate pool.ntp.org

2 Dec 07:47:09 ntpdate[5399]: step time server 200.11.116.1 offset 1.817247 sec

Como ven aca teniamos desfaso el reloj casi dos segundos. Aca esta utilidad se conecto al servidor pool.ntp.org y ajusto nuestra hora.

El ntpdate se puede agregar en el crontab para que actualice cada dos horas.

0 */2 * * * root /usr/sbin/ntpdate pool.ntp.org > /dev/null 2>&1

 # ntpdate -q ar.pool.ntp.org

server 200.3.168.192, stratum 2, offset -0.039172, delay 0.04465

server 200.11.116.1, stratum 2, offset -0.040442, delay 0.04062

server 200.59.8.234, stratum 3, offset -0.037901, delay 0.03978

12 Dec 23:02:40 ntpdate[6000]: adjust time server 200.11.116.1 offset -0.040442 sec

No obstante, necesitará instalar y mantener una entrada en crontab en cada uno de sus hosts, y además, dependiendo de la calidad del hardware, el reloj del sistema puede desincronizarse en ese intervalo de dos horas. Puede asegurarse de que el reloj del sistema esté sincronizado cada vez que piense que puede no estarlo, instalando y ejecutando un servidor NTP en su host. Esto mantendrá su host sincronizado y además le permitirá utilizarlo para sincronizar otros hosts de su red.

Ntpd

La configuración principal del servicio de NTP se encuentra en **/etc/ntp.conf** En /etc/sysconfig o en /etc/default se encuentran las opciones de inicio del servicio.

 # cat /etc/sysconfig/ntpd

Command line options for ntpd

OPTIONS="-g"

 # cat /etc/ntp.conf

Drift file. Put this in a directory which the daemon can write to.

No symbolic links allowed, either, since the daemon updates the file

by creating a temporary in the same directory and then rename()'ing

it to the file.

driftfile /var/lib/ntp/drift

statsdir /var/log/ntpstats/

statistics loopstats peerstats clockstats

filegen loopstats file loopstats type day enable

filegen peerstats file peerstats type day enable

filegen clockstats file clockstats type day enable

Use public servers from the pool.ntp.org project.

Please consider joining the pool (http://www.pool.ntp.org/join.html).

server 0.centos.pool.ntp.org

server 1.centos.pool.ntp.org

server 2.centos.pool.ntp.org

Con la lista de servers se está estableciendo los servidores stratum para nuestro cliente de NTP.

Explicación

La directiva **driftfile** proporciona al servidor un sitio donde guardar la información sobre la idiosincrasia del reloj del sistema local. Cuando pase el tiempo, utilizará esta información para calcular la hora con más precisión entre los intervalos de sincronización, dado que el demonio sabe cómo mantener la hora local correcta.

 # cat /var/lib/ntp/drift

4.787

Los informes estadísticos corresponden a la directiva statistics y se activarán cada uno como dice el archivo.

loopstats información de actualizaciones por el servidor local

peerstats registra información sobre todos los peers (tanto servidores superiores como clientes que utilizan nuestra reloj para sincronizar)

clockstats escribe información estadística del reloj local en el archivo del registro.

filegen le indica al demonio en que archivo queremos que se escriba esta información estadística y con que frecuencia se necesita cambiar el archivo (type day)

server le indica a ntpd que servidor superior debe utilizar para la sincronización.

restrict se utiliza para definir las clases de acceso. También se definen los mismos niveles de acceso para los clientes ipv4 y ipv6 utilizando parámetros 4 y -6

Ahora en esta otra parte define como los host pueden acceder a nuestro servidor ntp.

Permit time synchronization with our time source, but do not

permit the source to query or modify the service on this system.

restrict default kod nomodify notrap nopeer noquery

restrict -6 default kod nomodify notrap nopeer noquery

Permit all access over the loopback interface. This could

be tightened as well, but to do so would effect some of

the administrative functions.

restrict 127.0.0.1

restrict -6 ::1

Hosts on local network are less restricted.

#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

Palabra Clave	Definición
default	Define cuál será la acción a tomar para los hosts no definidos en restrict
kod	Se utiliza para ralentizar las consultas de clientes, enviando un paquete especial (kiss of death)
notrap	Rechaza paquetes de control

nopeer	Asegura que el servidor no utilizará un cliente conectado como servidor ntp
nomodify	Rechaza los intentos de actualización de hora en nuestro servidor ntp
noquery	Rechaza las peticiones de información y configuración, evita que el servidor sea consultado por estadísticas peer y otras.

El segundo ejemplo de conjunto de directivas restrict garantiza que las conexiones desde la máquina local puedan configurarse e interrogar al servidor NTP. No obstante, ninguna de estas evita que un cliente se sincronice con nuestro servidor NTP. (restrict 127.0.0.1)

Parámetros que podemos usar el comando ntpd

-c file le indicamos un archivo de configuración

-g Esta opción le permitirá ntpd iniciar en un sistema con un reloj que está más afuera porque el umbral de pánico (1000 por defecto , segundos)

-n Normalmente ntpd se ejecuta como un demonio, pero con esta opción deshabilita ese comportamiento

-q Se le comunica a ntpd que se finalice luego de sincronizar.

-N Arranca ntpd con el nivel de prioridad más alto posible.

ntpq

El comando ntpq es una utilidad para consultar un servidor ntp y determinar su performance.

Opciones

-c comando

-i modo interactivo

-n no resuelve nombres

-p consulta de peers o usar -c peers

La siguiente tabla muestra los valores que nos devolverá en sus columnas el comando ntpq:

Palabra Clave	Definición
remote	IP o nombre del servidor ntp a consultar, definido en ntp.conf
refid	IP o nombre del servidor actualmente en uso.
st	Número de stratum del servidor.
when	Número de segundos que pasaron desde la última consulta.
poll	Número de segundos entre las dos últimas consultas.
reach	Muestra la cantidad de veces que fue alcanzado el servidor. Cada consulta satisfactoria incrementa en uno el campo.
delay	Tiempo en milisegundos que tomó en responder el servidor a nuestra consulta.
offset	Muestra la diferencia de tiempo en milisegundos que hay entre el reloj del sistema y el proveedor de ntp.
jitter	Indica la diferencia de tiempo en milisegundos entre las últimas dos muestras.

Otros términos que hay que tener en cuenta:

Stepping y Slewing: El protocolo NTP inicia la sincronización del tiempo entre el tiempo del consumidor y el tiempo del proveedor aproximadamente una vez por minuto. Sin embargo, el intervalo aumenta gradualmente a una vez cada 17 minutos una vez que el tiempo

está estrechamente sincronizado entre el proveedor y el consumidor.
En esencia, los ajustes grandes ocurren con relativa rapidez, pero sólo los pequeños ajustes se realizan a través de un intervalo más largo.
Si la diferencia de tiempo entre el proveedor y el consumidor es pequeña (menos de 128 milisegundos), entonces NTP ajusta la hora en el consumidor poco a poco. Esto se llama rotación(slewing). Si, por el contrario, la diferencia de tiempo entre el proveedor y el consumidor es relativamente grande, entonces los ajustes se realizan con mayor rapidez en el consumidor. Esto se llama paso a paso(stepping).

Insane Time: Si la diferencia de tiempo entre el proveedor y el consumidor es más de 17 minutos, el demonio NTP (ntpd) considera el tiempo como “insane” y no ajusta el tiempo.

Drift: NTP mide y corrige los errores accidentales de frecuencia de reloj (llamado de deriva(drift)). Escribe el valor de la frecuencia actual en el archivo ntp.drift en el directorio **/var/lib/ntp/drift**. Si detiene y reinicia el demonio NTP, se inicializa la frecuencia de reloj a través del valor de este archivo. Esto evita que ntpd tenga tener que aprender de nuevo el error de frecuencia asociado con el reloj del sistema .

Jitter: Jitter es la diferencia de tiempo estimado entre el consumidor y el proveedor desde el último sondeo.
Ejemplo:

 # ntpq -c peers -n ar.pool.ntp.org
remote refid st t when poll reach delay offset jitter
==
+216.244.192.3 193.190.230.65 2 u 994 1024 377 7.538 -17.797 5.992
200.220.152.62 200.189.40.8 3 u 1945 1024 376 165.686 -134.38 8.285
+200.137.65.85 200.137.64.20 3 u 917 1024 377 269.163 -15.150 19.308
*200.189.40.8 200.20.186.76 2 u 173 1024 377 50.197 -18.401 19.196

Para hacer una consulta:
 # ntpq -4 -p localhost
remote refid st t when poll reach delay offset jitter
==

+mx30178.godns.n 200.69.222.92 5 u 29 64 377 33.565 -2.459 10.417
+ntp.copaco.com. 201.198.247.252 2 u 23 64 377 53.475 4.583 14.489
*a.st1.ntp.br .ONBR. 1 u 23 64 377 345.858 4.637 10.048

Muestra todos los peers que están conectados con ipv4 en formato numérico
 # ntpq -n -4 -p localhost
remote refid st t when poll reach delay offset jitter
==

+190.228.30.178 200.69.222.92 5 u 33 64 377 33.565 -2.459 10.417
+201.217.3.85 201.198.247.252 2 u 27 64 377 53.475 4.583 14.489
*200.160.7.186 .ONBR. 1 u 27 64 377 345.858 4.637 10.048

Comando ntptrace

Se utiliza para rastrear los servidores de ntp.

 # ntptrace ntp0.cornell.edu

cudns.cit.cornell.edu: stratum 2, offset -0.004214, synch distance 0.03455

dtc-truetime.ntp.aol.com: stratum 1, offset -0.005957, synch distance

0.00000, refid 'ACTS'

 # ntptrace ntp-2.mcs.anl.gov

mcs.anl.gov: stratum 2, offset -0.004515, synch distance 0.06354

clepsydra.dec.com: stratum 1, offset 0.002045, \

synch distance 0.00107, refid 'GPS

Cliente NTP

Para poder configurar nuestro servicio ntp como cliente debemos tener al menos estas líneas (en /etc/ntp.conf):

```
driftfile /var/lib/ntp/drift

# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery

# Permit all access over the loopback interface.  This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict -6 ::1
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.fedora.pool.ntp.org iburst
server 1.fedora.pool.ntp.org iburst
server 2.fedora.pool.ntp.org iburst
server 3.fedora.pool.ntp.org iburst
```

Server NTP

Para poder hacer que nuestro equipo sea usado como servidor NTP deberíamos tener que retocar ciertos parámetros.

```
restrict 192.168.122.0 mask 255.255.255.0 notrap nomodify

peer 0.rhel.pool1.ntp.org
peer 1.rhel.pool1.ntp.org
peer 2.rhel.pool1.ntp.org
peer 3.rhel.pool1.ntp.org
```

Nota: Algunas distribuciones usan de manera predeterminada **chrony** que es un software alternativa a **ntpd**, pensado para computadores

que no están prendidas las 24 hs.

Bibliografía

Libros:

LPI Linux Certification in a Nutshell, Third Edition, June 2010 LPIC-1: Linux Professional Institute Certification Study Guide: (Exams 101 and 102), 2nd Edition, February 2009

Páginas:

[1] Mantenimiento Reloj
[2]Reloj Del Sistema
[3]Configurar Reloj Servidor
[4] Guia 1
[5] Guia 2
[6] Guia 3
[7] Guia 4
[8] chrony

LABORATORIO COMPILACIÓN

Kernel

Kernel Monolítico

Un núcleo monolítico es un tipo de núcleo o kernel de un sistema operativo. Como ejemplo de sistema operativo de núcleo monolítico están UNIX, Linux y FreeBSD.[3]

Estos sistemas tienen un núcleo grande y complejo, que engloba todos los servicios del sistema. Está programado de forma no modular, y tiene un rendimiento mayor que un micronúcleo. Sin embargo, cualquier cambio a realizar en cualquier servicio requiere la recopilación del núcleo y el reinicio del sistema para aplicar los nuevos cambios.

Hay diversas ramificaciones de este diseño, que se han ido amoldando a nuevas necesidades. Podemos citar el sistema de módulos ejecutables en tiempo de ejecución, que le brinda al modelo de núcleo monolítico algunas de las ventajas de un micronúcleo. Dichos módulos pueden ser compilados, modificados, cargados y descargados en tiempo de ejecución, de manera similar a los servicios de un micronúcleo, pero con la diferencia de que se ejecutan en el espacio de memoria del núcleo mismo (anillo 0). De esta forma, un bloqueo del módulo, es probable que bloquee todo el núcleo. Además, el módulo pasa a formar un todo con el núcleo, usando la API del mismo, y no se emplea un sistema de mensajes como en los micronúcleos. Este es el esquema usado por, entre otros, Linux, FreeBSD y varios derivados de UNIX. Cabe resaltar que el paso constante de mensajes entre los servicios del micronúcleo, es en parte responsable del pobre rendimiento de los micronúcleos.

Un sistema operativo con núcleo monolítico concentra todas las funcionalidades posibles (planificación, sistema de archivos, redes, controladores de dispositivos, gestión de memoria, etc) dentro de un gran programa. El mismo puede tener un tamaño considerable, y deberá ser recopilado por completo al añadir una nueva funcionalidad. Todos los componentes funcionales del núcleo tienen acceso a todas sus estructuras de datos internas y a sus rutinas. Un error en una rutina puede propagarse a todo el núcleo.

La alternativa es tener una estructura de micronúcleo, donde las partes funcionales están divididas en unidades separadas con mecanismos de comunicación estrictos entre ellos.

Sistemas operativos con núcleos monolíticos

Entre los sistemas operativos que cuentan con núcleos monolíticos se encuentran:

```
Núcleos tipo Unix
    Linux
    Syllable
    Unix
        BSD (FreeBSD,NetBSD,OpenBSD)
        Solaris
Núcleos tipo DOS
    DR-DOS
    MS-DOS
    Familia Microsoft Windows 9x (95, 98, 98SE, Me)
Núcleos del Mac OS hasta Mac OS 8.6
OpenVMS
XTS-400
```

ARQUITECTURA

Diagrama del núcleo

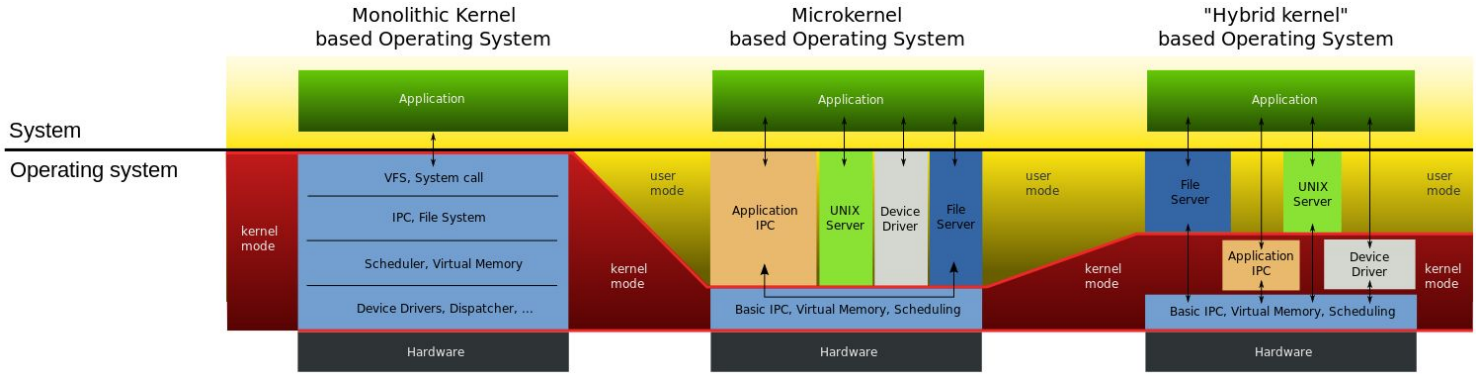


Imagen: By Golftheman - http://en.wikipedia.org/wiki/Image:OS-structure.svg, Public Domain, https://commons.wikimedia.org/w/index.php?curid=4397379

Actualmente Linux es un núcleo monolítico híbrido. Los controladores de dispositivos y las extensiones del núcleo normalmente se ejecutan en un espacio privilegiado conocido como anillo 0 (ring 0), con acceso irrestricto al hardware, aunque algunos se ejecutan en espacio de usuario. A diferencia de los núcleos monolíticos tradicionales, los controladores de dispositivos y las extensiones al núcleo se pueden cargar y descargar fácilmente como módulos, mientras el sistema continúa funcionando sin interrupciones. También, a diferencia de los núcleos monolíticos tradicionales, los controladores pueden ser prevolcados (detenidos momentáneamente por actividades más importantes) bajo ciertas condiciones. Esta habilidad fue agregada para gestionar correctamente interrupciones de hardware, y para mejorar el soporte de multiprocesamiento simétrico.

El hecho de que Linux no fuera desarrollado siguiendo el diseño de un micronúcleo (diseño que, en aquella época, era considerado el más apropiado para un núcleo por muchos teóricos informáticos) fue asunto de una famosa y acalorada discusión entre Linus Torvalds y Andrew S. Tanenbaum.[2]

Categorías del kernel

Prepatch o RC	Mainline	Stable	Longterm	Distribution
Tiene nuevas funcionalidades aun no probadas	Posee Nuevas funcionalidades	Tiene correcciones de errores provenientes de	Corrección de errores para las ramas más viejas	Cualquier versión que viene empaquetada en

		Mainline		una distribución de Linux y no ha sido compilada “a mano” a partir del código fuente de kernel.org

Compilando el Kernel

Primero antes que nada vamos a listar paquetes que necesitaremos para poder compilar nuestro Kernel.

Si quisiéramos el kernel que nos provee nuestra distribución

```
kernel.x86_64
kernel-headers.x86_64
kernel-devel.x86_64
[root@localclient ~]# yum install kernel kernel-headers
Setting up Install Process
Package kernel-headers-2.6.18-274.17.1.el5.x86_64 already installed
and latest version
Resolving Dependencies
--> Running transaction check
---> Package kernel.x86_64 0:2.6.18-274.17.1.el5 set to be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

```
================================================================================
================================================================
 Package                     Arch                        Version
Repository                    Size
================================================================================
================================================================
Installing:
 kernel                      x86_64                      2.6.18-
274.17.1.el5                      updates                        21 M

Transaction Summary
================================================================================
================================================================
Install       1 Package(s)
Upgrade       0 Package(s)

Total download size: 21 M
Is this ok [y/N]:
[root@localclient kernels]# yum install kernel-devel.x86_64
Dependencies Resolved

================================================================================
================================================================
 Package                         Arch                       Version
Repository                  Size
================================================================================
================================================================
Installing:
 kernel-devel                    x86_64                     2.6.18-
274.17.1.el5                      updates                    5.6 M

Transaction Summary
================================================================================
================================================================
Install       1 Package(s)
Upgrade       0 Package(s)

Total download size: 5.6 M
Is this ok [y/N]:
```

Si quisiéramos el kernel oficial de Kernel.org.

```
[root@localclient ~]# cd /usr/src/kernels/
[root@localclient kernels]# ls
[root@localclient kernels]# wget
http://www.kernel.org/pub/linux/kernel/v3.0/linux-3.2.2.tar.bz2
--2012-01-30 02:56:09--
http://www.kernel.org/pub/linux/kernel/v3.0/linux-3.2.2.tar.bz2
Resolviendo www.kernel.org... 149.20.4.69
Connecting to www.kernel.org|149.20.4.69|:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 78124665 (75M) [application/x-bzip2]
Saving to: `linux-3.2.2.tar.bz2'

11% [========>
] 8.656.467    272K/s  eta 2m 48s
```

Ahora vamos a listar el directorio donde se encuentra el kernel .

```
[root@localclient ~]# ls -l /usr/src/kernels/
total 8544
drwxr-xr-x 19 root root    4096 ene 30 03:05 2.6.18-274.17.1.el5-
x86_64
drwxrwxr-x  3 root root    4096 ene 30 03:06 linux-3.2.2
[root@localclient ~]#
```

Tenemos el kernel del paquete de fedora y el kernel oficinal de kernel.org

Menú Compilación

Para poder empezar a compilar nuestro kernel debemos ir al directorio donde se encuentra este.

```
[root@localclient kernels]# pwd
/usr/src/kernels
[root@localclient kernels]# cd 2.6.18-274.17.1.el5-x86_64/
[root@localclient 2.6.18-274.17.1.el5-x86_64]# ls
arch     fs       kabi_whitelist  mm              net       sound
block    include  kernel          Module.kabi     samples   symsets-
2.6.18-274.17.1.el5.tar.gz
crypto   init     lib             Module.markers  scripts   usr
drivers  ipc      Makefile        Module.symvers  security
[root@localclient 2.6.18-274.17.1.el5-x86_64]# ls -la .config
-rw-r--r-- 1 root root 67614 ene 10 19:57 .config
[root@localclient 2.6.18-274.17.1.el5-x86_64]#
```

El archivo .config es el que contiene la data de todo lo que va a contener nuestro kernel, tanto lo que está totalmente compilado o lo que va a estar como módulo.

Luego los comandos esenciales para poder empezar a tocar de una manera amigable las muchas opciones que hay son los siguientes:

Tener en cuenta que el archivo .config me sirve para luego copiarlo a otra pc que quiera compilar con las mismas opciones, o sino una vez compilado el kernel podemos generar un paquete rpm e instalar el kernel en otro equipo sin tener que compilar devuelta. También si bajamos algún kernel y queremos usar las opciones que teníamos antes podemos copiarlo en la carpeta donde bajamos el kernel.

make config

Probemos algunos:

```
[root@localclient 2.6.18-274.17.1.el5-x86_64]# make config
  HOSTLD  scripts/kconfig/conf
scripts/kconfig/conf arch/x86_64/Kconfig
*
* Linux Kernel Configuration
*
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (EXPERIMENTAL)
[Y/n/?] y
*
* General setup
*
Local version - append to kernel release (LOCALVERSION) [] y
Automatically append version information to the version string
(LOCALVERSION_AUTO) [N/y/?]
```

Así nos va a preguntar por cada opción existente.

make menuconfig

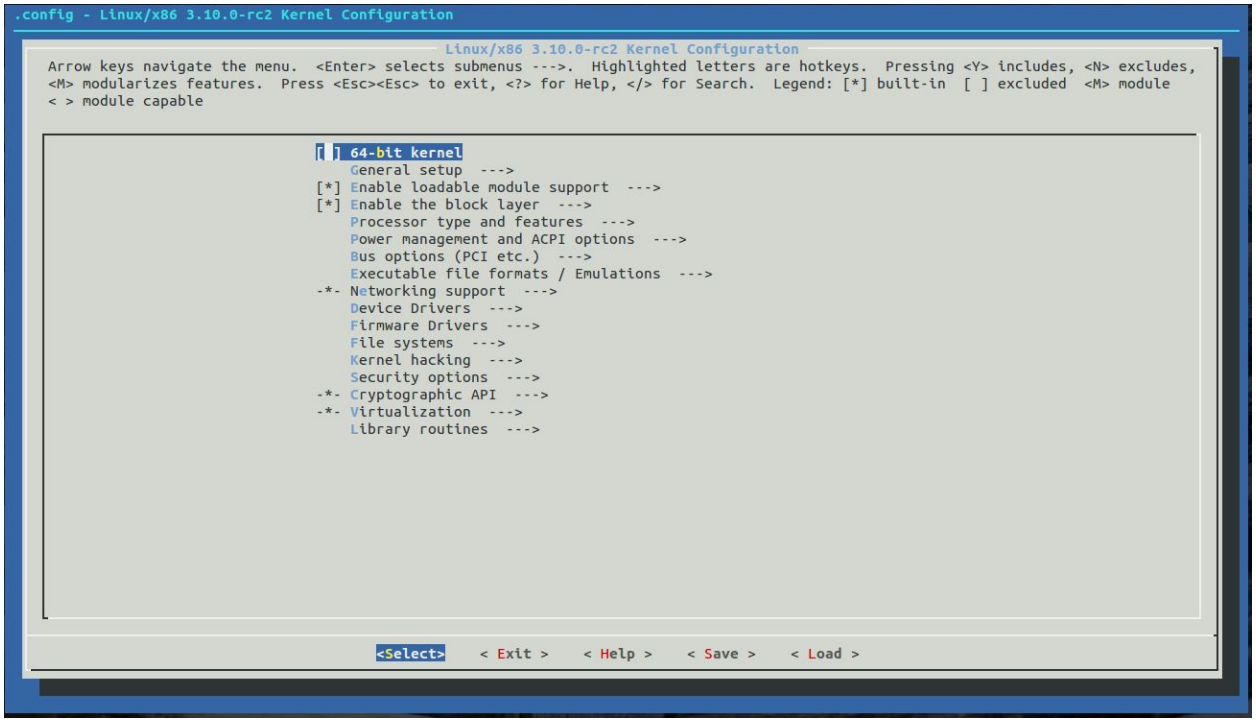
Otra forma más amigable con menuconfig:

```
[root@localclient 2.6.18-274.17.1.el5-x86_64]# make menuconfig
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
En el fichero incluído de scripts/kconfig/lxdialog/checklist.c:24:
scripts/kconfig/lxdialog/dialog.h:31:20: error: curses.h: No existe el
fichero o el directorio
```

Si tenemos ese error es porque no tenemos el paquete ncurses que es el que se encarga de armar en nuestro equipo el menú.

```
[root@localclient 2.6.18-274.17.1.el5-x86_64]# yum install ncurses-
devel.x86_64
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
================================================================
 Package                          Arch                          Version
Repository                  Size
================================================================================
================================================================
Installing:
 ncurses-devel                    x86_64                        5.5-
24.20060715                    base                         1.7 M

Transaction Summary
================================================================================
================================================================
Install       1 Package(s)
Upgrade       0 Package(s)

Total download size: 1.7 M
Is this ok [y/N]:
<code>
Ahora si...
<code>
```

```
[root@localclient 2.6.18-274.17.1.el5-x86_64]# make menuconfig
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/lxdialog.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTCC  scripts/kconfig/lxdialog/msgbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTLD  scripts/kconfig/lxdialog/lxdialog
scripts/kconfig/mconf arch/x86_64/Kconfig
```



make gconfig

Ahora la siguiente imagen muestra la utilización de gconfig

make bzImage

Una vez terminado el proceso anterior, previo a este podríamos haber ejecutado un make clean para que elimine todo lo que tenga de una compilación previa.

Este paso nos va a crear el archivo del kernel.

bzImage es una versión comprimida del kernel, similar a vmlinuz. Este fue creado para evitar ciertas limitaciones técnicas que tenia la creación de un kernel zimage.

zImage ya no se usa más es similar al formato vmlinuz. Anteriormente cuando se compilaba se usaba ese nombre, como el kernel fue creciendo en tamaño este último para los sistemas modernos quedo inadecuado dado que está limitado en tamaño en 512KB.

vmlinux es una versión descomprimida del kernel de linux, generalmente es usada como paso intermedio y no es copiada en /boot. Este kernel no esta preparado para arrancar dado que algunas opciones no están habilitadas.

vmlinuz es una variante de vmlinux en donde este se encuentra comprimido y con diversas herramientas que habilitan su uso para arrancar el sistema agregando otras características más. Las distribuciones de linux proveen generalmente un kernel vmlinuz como nombre del kernel precompilado de sus paquetes.

make bzImage

El siguiente cuadro muestra la forma de que se creara el kernel. En las versiones 2.6 en adelante se puede ejecutar **make** sin parámetros, creará el bzImage de manera predeterminada.

```
[root@oc6127656113 linux-3.2.2]# make bzImage
  HOSTLD  scripts/kconfig/conf
scripts/kconfig/conf --silentoldconfig Kconfig
  CHK     include/linux/version.h
  UPD     include/linux/version.h
  CHK     include/generated/utsrelease.h
  UPD     include/generated/utsrelease.h
  CC      kernel/bounds.s

  CC      arch/x86/boot/regs.o
  CC      arch/x86/boot/string.o
  CC      arch/x86/boot/tty.o
  CC      arch/x86/boot/video.o
  CC      arch/x86/boot/video-mode.o
  CC      arch/x86/boot/version.o
  CC      arch/x86/boot/video-vga.o
  CC      arch/x86/boot/video-vesa.o
  CC      arch/x86/boot/video-bios.o
  LD      arch/x86/boot/setup.elf
  OBJCOPY arch/x86/boot/setup.bin
  OBJCOPY arch/x86/boot/vmlinux.bin
  HOSTCC  arch/x86/boot/tools/build
  BUILD   arch/x86/boot/bzImage
Setup is 16876 bytes (padded to 16896 bytes).
System is 3937 kB
CRC 815c13c5
Kernel: arch/x86/boot/bzImage is ready  (#1)
[root@oc6127656113 linux-3.2.2]#
```

La imagen del kernel fue generada.

```
[root@oc6127656113 linux-3.2.2]# ls -l arch/x86/boot/bzImage
-rw-r--r-- 1 root root 4047632 ene 30 06:37 arch/x86/boot/bzImage
[root@oc6127656113 linux-3.2.2]# ls -l arch/x86_64/boot/bzImage
lrwxrwxrwx 1 root root 22 ene 30 06:37 arch/x86_64/boot/bzImage ->
../../x86/boot/bzImage
[root@oc6127656113 linux-3.2.2]#
```

make modules

Ahora el próximo paso es crear los módulos del kernel y esto va a tardar más que lo anterior. En la versión 2.6 en adelante se puede ejecutar make modules_install directamente.

```
[root@oc6127656113 linux-3.2.2]# make modules
  CHK     include/linux/version.h
  CHK     include/generated/utsrelease.h
  CALL    scripts/checksyscalls.sh
  AS [M]  arch/x86/crypto/ghash-clmulni-intel_asm.o
  CC [M]  arch/x86/crypto/ghash-clmulni-intel_glue.o
  AS [M]  arch/x86/crypto/salsa20-x86_64-asm_64.o
  CC [M]  arch/x86/crypto/salsa20_glue.o
IHEX    firmware/mts_edge.fw
  H16TOFW firmware/edgeport/boot.fw
  H16TOFW firmware/edgeport/boot2.fw
  H16TOFW firmware/edgeport/down.fw
  H16TOFW firmware/edgeport/down2.fw
  IHEX    firmware/edgeport/down3.bin
  IHEX2FW firmware/whiteheat_loader.fw
  IHEX2FW firmware/whiteheat.fw
  IHEX2FW firmware/keyspan_pda/keyspan_pda.fw
  IHEX2FW firmware/keyspan_pda/xircom_pgs.fw
  IHEX    firmware/cpia2/stv0672_vp4.bin
  IHEX    firmware/yam/1200.bin
  IHEX    firmware/yam/9600.bin
[root@oc6127656113 linux-3.2.2]#
```

make modules_install

Para poder instalar los módulos y generar los archivos correspondientes faltaria este paso.

```
[root@oc6127656113 linux-3.2.2]# make modules_install
  INSTALL arch/x86/crypto/crc32c-intel.ko
  INSTALL arch/x86/crypto/ghash-clmulni-intel.ko
  INSTALL arch/x86/crypto/salsa20-x86_64.ko
  INSTALL arch/x86/crypto/twofish-x86_64.ko
  INSTALL arch/x86/kernel/microcode.ko

  INSTALL /lib/firmware/edgeport/boot.fw
  INSTALL /lib/firmware/edgeport/boot2.fw
  INSTALL /lib/firmware/edgeport/down.fw
  INSTALL /lib/firmware/edgeport/down2.fw
  INSTALL /lib/firmware/edgeport/down3.bin
  INSTALL /lib/firmware/whiteheat_loader.fw
  INSTALL /lib/firmware/whiteheat.fw
  INSTALL /lib/firmware/keyspan_pda/keyspan_pda.fw
  INSTALL /lib/firmware/keyspan_pda/xircom_pgs.fw
  INSTALL /lib/firmware/cpia2/stv0672_vp4.bin
  INSTALL /lib/firmware/yam/1200.bin
  INSTALL /lib/firmware/yam/9600.bin
  DEPMOD  3.2.2
[root@oc6127656113 linux-3.2.2]#
```

PREPARAR ARRANQUE

Manualmente

Para lograrlo de forma manual deberíamos hacer lo siguiente.

```
cp arch/x86/boot/bzImage /boot/bzImage-3.2.2
cp System.map /boot/System.map-3.2.2
ln -s /boot/System.map-3.2.2 /boot/System.map
cat /boot/System.map > /boot/vmlinuz-3.2.2
ln -s /boot/vmlinuz /boot/vmlinuz-3.2.2
```

Luego tendríamos que mirar alguna entrada vieja del grub y editarla para que siga el mismo patrón que el kernel nuevo.

```
title Open Client Fedora 64 3.10 (Daily) (3.2.2)
        kernel /vmlinuz-3.2.2 ro root=/dev/mapper/vg_restaurador-
RootVol rd_LUKS_UUID=luks-11afdf03-a117-4c12-8139-faba5bc00604
rd_LVM_LV=vg_restaurador/RootVol rd_LVM_LV=vg_restaurador/SwapVol
rd_NO_MD rd_NO_DM LANG=es_AR.UTF-8 SYSFONT=latarcyrheb-sun16
KEYTABLE=la-latin1 pcie_aspm=force rhgb quiet selinux=0
        initrd /initramfs-3.2.2.img
```

Luego generamos la imagen de RAM disk.

```
mkinitrd /boot/initrd-3.2.2.img 3.2.2
```

make install

De esta forma nos instala el kernel y editar el grub dejando todo preparado.

```
[root@oc6127656113 linux-3.2.2]# make install
sh /usr/src/kernels/linux-3.2.2/arch/x86/boot/install.sh 3.2.2
arch/x86/boot/bzImage \
        System.map "/boot"
[root@oc6127656113 linux-3.2.2]#
[root@oc6127656113 linux-3.2.2]# ls -l /boot/System.map-3.2.2
/boot/vmlinuz-3.2.2
-rw-r--r-- 1 root root 2458659 ene 30 20:45 /boot/System.map-3.2.2
-rw-r--r-- 1 root root 4047632 ene 30 20:45 /boot/vmlinuz-3.2.2
[root@oc6127656113 linux-3.2.2]#
```

El grub ya posee la entrada nueva.

```
[root@oc6127656113 ~]# cat /boot/grub/grub.conf |grep 3.2
title Open Client Fedora 64 3.10 (Daily) (3.2.2)
        kernel /vmlinuz-3.2.2 ro root=/dev/mapper/vg_restaurador-
RootVol rd_LUKS_UUID=luks-11afdf03-a117-4c12-8139-faba5bc00604
rd_LVM_LV=vg_restaurador/RootVol rd_LVM_LV=vg_restaurador/SwapVol
rd_NO_MD rd_NO_DM LANG=es_AR.UTF-8 SYSFONT=latarcyrheb-sun16
KEYTABLE=la-latin1 pcie_aspm=force rhgb quiet selinux=0
        initrd /initramfs-3.2.2.img
[root@oc6127656113 ~]#
```

Podríamos ver que el script hace lo siguiente.

```
[root@oc6127656113 ~]# cat /usr/src/kernels/linux-
3.2.2/arch/x86/boot/install.sh
#!/bin/sh
#
# This file is subject to the terms and conditions of the GNU General
Public
# License.  See the file "COPYING" in the main directory of this
archive
# for more details.
#
# Copyright (C) 1995 by Linus Torvalds
#
# Adapted from code in arch/i386/boot/Makefile by H. Peter Anvin
#
# "make install" script for i386 architecture
#
# Arguments:
#   $1 - kernel version
#   $2 - kernel image file
#   $3 - kernel map file
#   $4 - default install path (blank if root directory)
#

verify () {
        if [ ! -f "$1" ]; then
                echo ""
1>&2
                echo " *** Missing file: $1"
1>&2
                echo ' *** You need to run "make" before "make
install".' 1>&2
                echo ""
1>&2
                exit 1
        fi
}

# Make sure the files actually exist
verify "$2"
verify "$3"
```

```
if [ -x /sbin/lilo ]; then
        /sbin/lilo
elif [ -x /etc/lilo/install ]; then
        /etc/lilo/install
else
        sync
        echo "Cannot find LILO."
fi
[root@oc6127656113 ~]#
```

Herramientas de RAM disk

Estas herramientas lo que me van a permitir es tener un set módulos y herramientas críticas esenciales para el arranque de la máquina que van a ser utilizadas en el inicio de arranque del kernel previo al montaje del kernel en el sistema.

El kernel es cargado en memoria como si estuviera en un disco, cargando los módulos y herramientas desde la RAM disk para luego poder montar el sistema raíz.

El comando es **mkinitrd** :

Podria usarse de una manera basica: **mkinitrd /boot/initrd-3.2.2.img 3.2.2**

INSTALANDO DESDE LAS FUENTES

En este laboratorio vamos a ver como instalar fluxbox desde la fuente.

Instalando Fluxbox

Lo primero que vamos hacer es bajar fluxbox desde la página oficinal.

–> http://sourceforge.net/projects/fluxbox/files/fluxbox/1.3.2/fluxbox-1.3.2.tar.gz

```
[root@localclient tmp]# wget
http://sourceforge.net/projects/fluxbox/files/fluxbox/1.3.2/fluxbox-
1.3.2.tar.gz
--2012-01-30 00:54:21--
http://sourceforge.net/projects/fluxbox/files/fluxbox/1.3.2/fluxbox-
1.3.2.tar.gz
Resolviendo sourceforge.net... 216.34.181.60
Connecting to sourceforge.net|216.34.181.60|:80... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Localización:
http://sourceforge.net/projects/fluxbox/files/fluxbox/1.3.2/fluxbox-
1.3.2.tar.gz/download [siguiendo]
--2012-01-30 00:54:22--
http://sourceforge.net/projects/fluxbox/files/fluxbox/1.3.2/fluxbox-
1.3.2.tar.gz/download
Connecting to sourceforge.net|216.34.181.60|:80... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Localización:
http://downloads.sourceforge.net/project/fluxbox/fluxbox/1.3.2/fluxbox
-1.3.2.tar.gz?r=&ts=1327895666&use_mirror=ufpr [siguiendo]
--2012-01-30 00:54:23--
http://downloads.sourceforge.net/project/fluxbox/fluxbox/1.3.2/fluxbox
-1.3.2.tar.gz?r=&ts=1327895666&use_mirror=ufpr
Resolviendo downloads.sourceforge.net... 216.34.181.59
Connecting to downloads.sourceforge.net|216.34.181.59|:80...
conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
```

```
Localización:
http://ufpr.dl.sourceforge.net/project/fluxbox/fluxbox/1.3.2/fluxbox-
1.3.2.tar.gz [siguiendo]
--2012-01-30 00:54:23--
http://ufpr.dl.sourceforge.net/project/fluxbox/fluxbox/1.3.2/fluxbox-
1.3.2.tar.gz
Resolviendo ufpr.dl.sourceforge.net... 200.236.31.2,
2801:82:80ff:8000::3
Connecting to ufpr.dl.sourceforge.net|200.236.31.2|:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1065005 (1,0M) [application/x-gzip]
Saving to: `fluxbox-1.3.2.tar.gz'

100%[=======================================================================
======================>] 1.065.005    184K/s   in 5,9s

2012-01-30 00:54:30 (177 KB/s) - `fluxbox-1.3.2.tar.gz' saved
[1065005/1065005]

[root@localclient tmp]#
```

Ahi lo bajamos.

Una vez que lo descomprimimos.

```
[root@localclient tmp]# cd fluxbox-1.3.2
[root@localclient fluxbox-1.3.2]# ls
3rd           AUTHORS      configure     data     INSTALL
Makefile.am  NEWS    src   version.h.in
acinclude.m4  ChangeLog    configure.in  depcomp  install-sh
Makefile.in  nls     TODO
aclocal.m4    config.h.in  COPYING       doc      ltmain.sh   missing
README  util
[root@localclient fluxbox-1.3.2]#
```

Siempre es una buena idea tratar de mirar toda la documentación posible para ver como se instala desde la fuente. Generalmente se encontrará un archivo **README** o un archivo **INSTALL** con las instrucciones.

```
[root@localclient fluxbox-1.3.2]# cat README
Fluxbox is a fork of the original Blackbox 0.61.1 sourcecode with
different goals.

Read NEWS to see whats new in this release.

For copyright information see COPYING

For more information go to:
  http://fluxbox.org/


Compile and Install:

  $ ./configure
  $ make
  and then as root
  # make install


Thanks:

  Blackbox team

  People at #fluxbox on the irc.freenode.net irc-network.

  skypher of openprojects for bugtesting and providing fluxbox with
    themes: Clean CleanColor Makro, Carbondioxide and MerleyKay.


  And all the people who sent bugfixes/patches and helped us making
  Fluxbox a better application.


[root@localclient fluxbox-1.3.2]#
```

Vamos a seguir los pasos que nos recomienda.

```
Compile and Install:

  $ ./configure
  $ make
  and then as root
  # make install
```

./configure

El script **configure** es el encargado de verificar si tenemos todo lo necesario para compilar. También al **configure** se le pasan parámetros como **--prefix** para indicarle donde se instalará el programa.

```
[root@localclient fluxbox-1.3.2]# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for gcc... no
checking for cc... no
checking for cl.exe... no
configure: error: in `/tmp/fluxbox-1.3.2':
configure: error: no acceptable C compiler found in $PATH
See `config.log' for more details.
[root@localclient fluxbox-1.3.2]#
```

En este caso muestra es que no tenemos un compilador que compile el código fuente que estamos intentando instalar.

 # yum groupinstall “Bibliotecas de desarrollo”

Si no, instalar herramienta por herramienta

```
[root@localclient fluxbox-1.3.2]# yum search gcc

gcc-c++.x86_64 : Soporte de C++ para GCC
<code>

Vamos a instalar gcc.\\
```

 # yum install gcc-c++.x86_64

 # yum grouplist “Desarrollo de software para X”

```
 mesa-libGLU                             x86_64                 6.5.1-
7.8.el5                                 base                  224 k
 mesa-libGLU-devel                       x86_64                 6.5.1-
7.8.el5                                 base                   91 k
 netpbm                                  i386                   10.35.58-
8.el5_7.3                               updates               834 k
 netpbm                                  x86_64                 10.35.58-
8.el5_7.3                               updates               836 k
 xorg-x11-proto-devel                    x86_64                 7.1-13.el5
base                      247 k
 xorg-x11-util-macros                    x86_64                 1.0.2-
4.fc6                                   base                    8.1 k
 zlib-devel                              x86_64                 1.2.3-
4.el5                                   base                  103 k
Updating for dependencies:
 freetype                                i386                   2.2.1-
28.el5_7.2                              updates               312 k
 freetype                                x86_64                 2.2.1-
28.el5_7.2                              updates               311 k
 libX11                                  i386                   1.0.3-
11.el5_7.1                              updates               797 k
 libX11                                  x86_64                 1.0.3-
11.el5_7.1                              updates               798 k
 libpng                                  i386                   2:1.2.10-
7.1.el5_7.5                             updates               241 k
 libpng                                  x86_64                 2:1.2.10-
7.1.el5_7.5                             updates               235 k

Transaction Summary
================================================================================
Install     145 Package(s)
Upgrade       6 Package(s)

Total download size: 37 M
Is this ok [y/N]:
```

Podríamos haber instalado el paquete necesario pero prefiero instalar esto en este caso dado que es para fines educativos.

```
[root@localclient fluxbox-1.3.2]# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
..
..
..
..
..
config.status: creating nls/zh_CN/Makefile
config.status: creating nls/zh_TW/Makefile
config.status: creating config.h
config.status: executing depfiles commands

        fluxbox version 1.3.2 configured successfully.

Using '/usr/local' for installation.
Using '$(prefix)/share/fluxbox/menu' for location menu file.
Using '$(prefix)/share/fluxbox/styles/bloe' by default style.
Using '$(prefix)/share/fluxbox/keys' for location keys file.
Using '$(prefix)/share/fluxbox/init' for location init file.
Using '$(prefix)/share/fluxbox/nls' for nls files.
Using 'g++' for C++ compiler.
Building with '-g -O2   -I/usr/include/freetype2  ' for C++ compiler
flags.
Building with '  -lX11  -lXft -lXrender -lfreetype -lX11 -lfontconfig
-lXrender -lXpm -lXext -lXrandr -lXrandr' for linker flags.

Now build fluxbox with 'make'

[root@localclient fluxbox-1.3.2]#
```

Como vemos ahi termino y nos dice cual es el siguiente paso.

Make

El comando make leerá un archivo llamado **Makefile**, el cual contendrá todo el listado de archivos a compilar generado por el configure.

```
[root@localclient fluxbox-1.3.2]# make
make  all-recursive
make[1]: se ingresa al directorio `/tmp/fluxbox-1.3.2'
Making all in doc
make[2]: se ingresa al directorio `/tmp/fluxbox-1.3.2/doc'
sed -e "s,@pkgdatadir@,/usr/local/share/fluxbox," fluxbox.1.in >
fluxbox.1
sed -e "s,@pkgdatadir@,/usr/local/share/fluxbox," fbrun.1.in > fbrun.1
sed -e "s,@pkgdatadir@,/usr/local/share/fluxbox," fbsetbg.1.in >
fbsetbg.1
sed -e "s,@pkgdatadir@,/usr/local/share/fluxbox," fbsetroot.1.in >
fbsetroot.1
sed -e "s,@pkgdatadir@,/usr/local/share/fluxbox," fluxbox-apps.5.in >
fluxbox-apps.5
sed -e "s,@pkgdatadir@,/usr/local/share/fluxbox," fluxbox-keys.5.in >
fluxbox-keys.5
sed -e "s,@pkgdatadir@,/usr/local/share/fluxbox," fluxbox-menu.5.in >
fluxbox-menu.5
sed -e "s,@pkgdatadir@,/usr/local/share/fluxbox," fluxbox-remote.1.in
> fluxbox-remote.1
sed -e "s,@pkgdatadir@,/usr/local/share/fluxbox," fluxbox-style.5.in >
fluxbox-style.5
```

```
La salida anterior esta recortada.\\
```

Los pasos desarrollados hasta ahora lo podríamos haber ejecutado sin necesidad de ser root.

Make install

Este último paso moverá todo lo compilado al directorio predeterminado o al que se haya definido con **--prefix** dependiendo de esto lo podremos ejecutar como usuario común o como root.

```
[root@localclient fluxbox-1.3.2]# make install
Making install in doc
make[1]: se ingresa al directorio `/tmp/fluxbox-1.3.2/doc'
make[2]: se ingresa al directorio `/tmp/fluxbox-1.3.2/doc'
make[2]: No se hace nada para `install-exec-am'.
test -z "/usr/local/share/man/man1" || /bin/mkdir -p
"/usr/local/share/man/man1"
 /usr/bin/install -c -m 644 fluxbox.1 fbrun.1 fbsetbg.1 fbsetroot.1
fluxbox-remote.1 startfluxbox.1 '/usr/local/share/man/man1'
test -z "/usr/local/share/man/man5" || /bin/mkdir -p
"/usr/local/share/man/man5"
 /usr/bin/install -c -m 644 fluxbox-apps.5 fluxbox-keys.5 fluxbox-
menu.5 fluxbox-style.5 '/usr/local/share/man/man5'
make[2]: se sale del directorio `/tmp/fluxbox-1.3.2/doc'
make[1]: se sale del directorio `/tmp/fluxbox-1.3.2/doc'
Making install in nls
..
..
Installing windowmenu file in /usr/local/share/fluxbox/windowmenu
/bin/sh /tmp/fluxbox-1.3.2/install-sh -d /usr/local/share/fluxbox/
/usr/bin/install -c -m 644 ./windowmenu
/usr/local/share/fluxbox/windowmenu
make[3]: se sale del directorio `/tmp/fluxbox-1.3.2/data'
make[2]: se sale del directorio `/tmp/fluxbox-1.3.2/data'
make[1]: se sale del directorio `/tmp/fluxbox-1.3.2/data'
make[1]: se ingresa al directorio `/tmp/fluxbox-1.3.2'
make[2]: se ingresa al directorio `/tmp/fluxbox-1.3.2'
make[2]: No se hace nada para `install-exec-am'.
make[2]: No se hace nada para `install-data-am'.
make[2]: se sale del directorio `/tmp/fluxbox-1.3.2'
make[1]: se sale del directorio `/tmp/fluxbox-1.3.2'
[root@localclient fluxbox-1.3.2]#
```

La salida fue recortada.

Hasta acá compilo todo correctamente pero nos falta instalar el Servidor X que fluxbox va a necesitar para poder iniciar.

Instalando X Server

Para instalar el X server vamos a instalar el paquete del server X11.

```
[root@localclient fluxbox-1.3.2]# yum install  xorg-x11-server-Xorg
Dependencies Resolved

================================================================================
================================================================
 Package                                Arch                    Version
Repository               Size
================================================================================
================================================================
Installing:
 xorg-x11-server-Xorg                   x86_64                  1.1.1-
48.76.el5_7.5                  updates                  3.4 M
Installing for dependencies:
 chkfontpath                            x86_64                  1.10.1-
1.1                           base                     15 k
 libFS                                  x86_64                  1.0.0-
3.1                            base                     30 k
 libdmx                                 x86_64                  1.0.2-
3.1                            base                     13 k
 libxkbfile                             x86_64                  1.0.3-
3.1                            base                     73 k
 ttmkfdir                               x86_64                  3.0.9-
23.el5                         base                     46 k
 xkeyboard-config                       noarch                  0.8-
9.el5_7.1                        updates                 321 k
 xorg-x11-drv-evdev                     x86_64
1:1.0.0.5-5.el5                    base                     10 k
 xorg-x11-drv-keyboard                  x86_64                  1.1.0-3
base                     14 k
 xorg-x11-drv-mouse                     x86_64                  1.1.1-
1.1                            base                     28 k
 xorg-x11-drv-vesa                      x86_64                  1.3.0-
8.3.el5                        base                     19 k
 xorg-x11-drv-void                      x86_64                  1.1.0-
3.1                            base                    7.6 k
 xorg-x11-font-utils                    x86_64                  1:7.1-3
base                     78 k
 xorg-x11-fonts-base                    noarch                  7.1-
2.1.el5                          base                   3.7 M
 xorg-x11-server-utils                  x86_64                  7.1-
5.el5_6.2                        base                   172 k
 xorg-x11-utils                         x86_64                  7.1-
2.fc6                            base                   123 k
 xorg-x11-xfs                           x86_64                  1:1.0.2-
5.el5_6.1                      base                    73 k
 xorg-x11-xkb-utils                     x86_64                  1.0.2-
2.1                            base                    183 k
```

```
Transaction Summary
================================================================================
================================================================
Install      18 Package(s)
Upgrade       0 Package(s)

Total download size: 8.3 M
Is this ok [y/N]:
```

 # yum install xorg-x11-xinit.x86_64

Generamos este archivo:

```
[root@localclient ~]# cat .xinitrc
exec xterm &
exec /usr/local/bin/fluxbox
[root@localclient ~]#
```

Para eso tendremos que instalar un xterm.

```
[root@localclient ~]# yum install xterm

Dependencies Resolved

================================================================================
================================================================
 Package                      Arch                          Version
Repository                   Size
================================================================================
================================================================
Installing:
 xterm                        x86_64                        215-
8.el5_4.1                         base                          410 k

Transaction Summary
================================================================================
================================================================
Install       1 Package(s)
Upgrade       0 Package(s)

Total download size: 410 k
Is this ok [y/N]:
```

Ahora si en la máquina tipeamos **xinit** iniciara fluxbox y abrirá una terminal **xterm**, dado que el **xinit** lee el archivo **xinitrc**.

Anexo: Versionado antiguo del kernel

Más allá de haber desarrollado su propio código y de integrar los cambios realizados por otros programas, Linus Torvalds continúa lanzando nuevas versiones del núcleo Linux. Estos son llamados núcleos “vanilla”, lo que significa que no han sido modificados por nadie. Muchos desarrolladores de distribuciones Linux modifican dicho núcleo en sus productos, principalmente para agregarle soporte a dispositivos o herramientas que no fueron oficialmente lanzadas como estables, mientras que algunas distribuciones, como Slackware, mantienen el núcleo vanilla.

Numeración

La versión del núcleo Linux actualmente consta de cuatro números. Por ejemplo, asumamos que el número de la versión está compuesta de esta forma: A.B.C[.D] (ej.: 2.2.1, 2.4.13 ó 2.6.12.3).

```
    El número A denota la versión del núcleo. Es el que cambia con
menor frecuencia y solo lo hace cuando se produce un gran cambio en el
código o en el concepto del núcleo. Históricamente sólo ha sido
modificado tres veces: en 1994 (versión 1.0), en 1996 (versión 2.0) y
en 2011 (versión 3.0).

    El número B denota la subversión del núcleo.

    Antes de la serie de Linux 2.6.x, los números pares indicaban la
versión "estable" lanzada. Por ejemplo una para uso de fabricación,
como el 1.2, 2.4 ó 2.6. Los números impares, en cambio, como la serie
2.5.x, son versiones de desarrollo, es decir que no son consideradas
de producción.

    Comenzando con la serie Linux 2.6.x, no hay gran diferencia entre
los números pares o impares con respecto a las nuevas herramientas
desarrolladas en la misma serie del núcleo. Linus Torvalds dictaminó
que este será el modelo en el futuro.

    El número C indica una revisión mayor en el núcleo. En la forma
anterior de versiones con tres números, esto fue cambiado cuando se
implementaron en el núcleo los parches de seguridad, bugfixes, nuevas
características o drivers. Con la nueva política, solo es cambiado
cuando se introducen nuevos drivers o características; cambios menores
se reflejan en el número D.

    El número D se produjo cuando un grave error, que requiere de un
arreglo inmediato, se encontró en el código NFS de la versión 2.6.8.
Sin embargo, no habían otros cambios como para lanzar una nueva
revisión (la cual hubiera sido 2.6.9). Entonces se lanzó la versión
2.6.8.1, con el error arreglado como único cambio. Con 2.6.11, esto
fue adoptado como la nueva política de versiones. Bug-fixes y parches
de seguridad son actualmente manejados por el cuarto número dejando
los cambios mayores para el número C.
```

También, algunas veces luego de las versiones puede haber algunas letras como “rc1” o “mm2”. El “rc” se refiere a release candidate e indica un lanzamiento no oficial. Otras letras usualmente (pero no siempre) hacen referencia a las iniciales de la persona. Esto indica una bifurcación en el desarrollo del núcleo realizado por esa persona, por ejemplo ck se refiere a Con Kolivas, ac a Alan Cox, mientras que mm se refiere a Andrew Morton.

El modelo de desarrollo para Linux 2.6 fue un cambio significativo desde el modelo de desarrollo de Linux 2.5. Previamente existía una rama estable (2.4) donde se habían producido cambios menores y seguros, y una rama inestable (2.5) donde estaban permitidos cambios mayores. Esto significó que los usuarios siempre tenían una versión 2.4 a prueba de fallos y con lo último en seguridad y casi libre de errores, aunque tuvieran que esperar por las características de la rama 2.5. La rama 2.5 fue eventualmente declarada estable y renombrada como 2.6. Pero en vez de abrir una rama 2.7 inestable, los desarrolladores de núcleos eligieron continuar agregando los cambios en la rama “estable” 2.6. De esta forma no había que seguir manteniendo una rama vieja pero estable y se podía hacer que las nuevas características estuvieran rápidamente disponibles y se pudieran realizar más test con el último código.

Sin embargo, el modelo de desarrollo del nuevo 2.6 también significó que no había una rama estable para aquellos que esperaban seguridad y bug fixes sin necesitar las últimas características. Los arreglos solo estaban en la última versión, así que si un usuario quería una versión con todos los bug fixed conocidos también tendría las últimas características, las cuales no habían sido bien testeadas. Una solución parcial para esto fue la versión ya mencionada de cuatro números (y en 2.6.x.y), la cual significaba lanzamientos puntuales creados por el equipo estable (Greg Kroah-Hartman, Chris Wright, y quizás otros). El equipo estable solo lanzaba actualizaciones para el núcleo más reciente, sin embargo esto no solucionó el problema del faltante de una serie estable de núcleo. Distribuidores de Linux, como Red Hat y Debian, mantienen los núcleos que salen con sus lanzamientos, de forma que una solución para algunas personas es seguir el núcleo de una distribución.

Como respuesta a la falta de un núcleo estable y de gente que coordinara la colección de corrección de errores, en diciembre de 2005 Adrian Bunk anunció que continuaría lanzando núcleos 2.6.16 aun cuando el equipo estable lanzara 2.6.17. Además pensó en incluir actualizaciones de controladores, haciendo que el mantenimiento de la serie 2.6.16 sea muy parecido a las viejas reglas de mantenimiento para las serie estables como 2.4. El núcleo 2.6.16 será reemplazado próximamente por el 2.6.27 como núcleo estable en mantenimiento durante varios años.

BIBLIOGRAFÍA

[1]Compilando fluxbox
[2]Kernel
[3]Kernel MOnolitico
[4]Kernel Fedora
[5]Device Drivers