



Guia Mayix Stage4

**Guia detallada para nuevos usuarios de Gentoo GNU/Linux!
Por medio de mayix-stage4
Guatemala Septiembre 2004**

Una pequeña introducción

¿Como nace mayix liveCD y mayix-stage4?

Corrían los primeros días del mes de octubre del año 2003 y como cualquier día, me levante a compilar una nueva versión del kernel, pero como era de esperarse para iniciar esta tarea tenia que tener una conexión rápida de Internet, debido a que por vía telefónica se iba una eternidad para bajar el source de 28 mbs mas todos los parches que fueran necesarios, y así como yo hay muchas personas en todas partes que no cuentan con un enlace dedicado de Internet y lo teníamos que realizar por medio de teléfono! :-), luego de algo de investigación encontré una distro llamada knoppix, la cual funcionaba como un disto live y también se podía instalar completamente, pero como era de esperarse no se trataba de mi distribución favorita, entonces me di cuenta del poder que tenia encerrado el concepto live, y que si podía encerrar todo el poder de Gentoo en este mismo formato podría llevarlo a cualquier persona que estuviera dispuesta a aprender y experimentar, como muchos de nosotros linuxeros fui recopilando material sobre el funcionamiento interno de una de estas distribuciones, lo cual no fue muy fácil debido al gran espacio vacío que hay sobre el tema, y debido a eso me día a la tarea de llenarlo muy modestamente con el material que conseguía y las experiencias que tenia, en una modesta pagina Web, llamada codenamevivo, que era el nombre con el que había bautizado el proyecto de documentación en un principio, pero bueno fueron pasando los días y meses y me llenaba de material sobre este tema, tratando de portarlo todo a gentoo linux, y es ahí cuando nace el nombre de Mayix, de 2 palabras mayas

MAY.- Descendiente

IX.- Menudo; Pequeño

Que uniéndolas a Mayix representa al mas pequeño descendiente de Gentoo Linux!, que nace por la necesidad e inquietud de incorporar a mas y mas personas que de otro modo no lograrían obtener todo lo necesario para la instalación de un gentoo linux completo, al finalizar este primer lanzamiento ya contaba con un enlace dedicado de Internet en mi lab (casa☺) y quise compartir todo esto con el mundo!!

Paulo Cesar Alvarado

www.mayix.net

linux user: 323835

Antes de iniciar con el proceso de instalacion

Requerimientos basicos

- 128 mbs en ram
- 2.2 Gbs en disco duro
- CDrom
- Computador capas de iniciar desde Cdrom
- Conocimientos basicos conceptos linux
- Mayix liveCd! (disponible desde www.mayix.net)
- Muchas ganas de leer y aprender...!!

- **Instalación:**

En los siguientes capítulos veremos instalar Mayix Stage4 a partir del livecd que has o vas a bajar, desde ya veras en esta guía alguna similitud con la guía que se encuentra en los foros en español de Gentoo Linux!, pues para apegarme fielmente a la instalacion basica de gentoo la tome como marco principal de referencia.

Desde ya hace algun tiempo he estado buscando una manera rapida facil y siempre confiable de instalar mi poderoso gentoo en todas las maquinas a las cuales tengo acceso, pero siempre me he encontrado con muchos problemas como:

1. carencia a una buena velocidad de internet o carencia total de acceso
2. pobre desempeño de los CPUs que tengo a mano (no siempre es como en casa ☺)
3. poca disponibilidad de tiempo para compilación
4. poca confianza en "otros instaladores"

en fin son a veces innumerables los contratiempos que se presentan al querer instalar mi poderoso gentoo!. Es por esas necesidades y muchas mas que puse manos a la obra para que todos pudieran tambien disfrutar de gentoo en todos lados.

Mayix Stage4

Que es un stage? Y porque stage4?

Como este manual esta dirigido a esas personas totalmente nuevas en el mundo linux voy a dar una pequeña explicación del significado de un stage

Gentoo linux puede ser instalado en base a 3 tipos de etapas "oficiales", esto significa que dependiendo de cual fase o stage elijas asi sera la cantidad de paquetes que seran compilados por ti mismo.

Stage1

La primera fase o stage de gentoo nos provee un total control sobre la optimizacion que le daremos a nuestro linux, esta fase esta diseñada para aquellos usuarios experimentados que siempre buscan llegar a los extremos con linux, para continuar desde esta fase se necesita acceso total a internet (acceso rapido de preferencia)

Stage 1 + pros y - contras

- + **permite al usuario un mayor control sobre las optimizaciones de los programas base, sobre los cuales se compilara todo el arbol de aplicaciones que contendra nuestro linux maduro**
- + **perfecto para esos usuarios que verdaderamente saben lo que estan haciendo**
- + **permite aprender mucho mas sobre el funcionamiento interno de Gentoo linux**
- **toma mucho mas tiempo la instalacion de un sistema completo**

- **si no sabes como aplicar optimizaciones a tu sistema creo que iniciar desde este punto es una perdida de tiempo**
- **no es recomendado para las personas que carecen de internet**

Stage2

La segunda fase o stage2 de la instalacion, es un sistema con optimizaciones y peculiaridades de linux ya aplicadas de "fabrica" evitandonos asi cometer errores fatales en la contruccion del arbol de aplicaciones, en esta fase necesitaremos compilar a "mano" tambien algunos programas pero evitando el rieso de cometer errores mayores

Stage 2 + pros y - contras

- + **no necesitamos saber que optimizaciones hacer o como hacerlas**
- + **mucho mas rapido que iniciar de stage 1**
- + **todavía se pueden aplicar algunas personalizaciones u optimizaciones pero ya sin tantos riesgos**

- **no se pueden optimizar las bases tanto como en el stage 1**
- **esta no es la forma mas rapida de instalar gentoo linux**

- **se tienen que aceptar muchas de las optimizaciones hechas al stage1 desde "fabrica" aunque no nos guste.**
- **no es recomendable para las personas que carecen de internet**

Stage3

La tercera fase o stage3 de la instalacion, eligiendo esta fase obtendremos la instalacion mas rapida de gentoo linux!, pero tambien tendremos un sistema totalmente construido con la optimizaciones que ha sido elegidos por los desarrolladores de gentoo (que como ellos dicen son las mejores, para estabilidad y desempeño)

Stage 3 **+** pros y **-** contras

- + la forma mas rapida de instalar gentoo Linux!**
- + puede ser utilizada para una instalacion sin acceso a internet**
- no se puede modificar ninguna de las optimizaciones del sistema base que ya esta constuido en el**
- no puedes saber como se utilizo el stage 1 y 2**
- cuando se instala nos presta soporte solo para modo consola a menos que se utilize el disco 2 con los paquetes precompilados que este contiene, no recomendable para novatos.**



Mayix Stage4

en esta fase de instalacion, se a intentado utilizar todos los pro del stage3 agregando varias ventajas, como un sistema facil de configurar, y sin necesidad de tener una coneccion de internet tan exigente como en las otras fases, ademas de ser esta la fase mas rapida de obtener, un sistema TOTAL!

Mayix Stage 4 **+** pros y **-** contras

- + la forma super mas rapida de instalar gentoo Linux!**
- + puede ser utilizada para una instalacion sin acceso a internet**
- + obtendremos en la instalacion un sistema consola y un sistema grafico (gnome)**
- + se han agregado muchas mas herramientas que las del listado oficial del stage3**
- + no necesitamos de paquetería extra para instalarlo todo**

■ como en el stage3 no se puede modificar ninguna de las optimizaciones del sistema base que ya esta constuido en el

■ no puedes saber como se utilizo el stage 1 2 y 3

Recuerda que siempre que aparezca el simbolo # y un comando lo que se debera de poner es unicamente el comando, el simbolo # se refiere al icono de identificación de una shell de usuario.

- **Como obtener Mayix Stage4 LiveCD!:**

La distribución Mayix LiveCD! Con el stage4 (ocupa cerca los 700mb) y se puede obtener via nuestra web, o directamente en <ftp://gentoo-es.org/mayix-livecd>

- **Arranque desde el CD:**

Introducimos el cd en la unidad lectora y encendemos el ordenador. Acontinuación deberemos indicarle a la bios que arranque desde cd y no desde el disco duro(HD). Para acceder a la BIOS deberemos de apretar la letra de "Supr" que se encuentra alado del intro, justo cuando nos aparezca las letras de informacion de nuestro hardware, antes de ver el arranque de nuestro sistema operativo. Una vez hecho, al llegar al arranque nos leera el cd, nos aparecera un dibujo de gentoo y en la zona inferior nos aparecera "boot:", donde deberemos de ponerle opciones especiales que queremos que cargue con el arranque(apretar la letra F2 para ver las opciones), normalmente no nos hara falta poner nada, por lo que pulsaremos intro para cargar el kernel. Una vez cargado el kernel nos aparecera una shell donde comenzara nuestra instalación.

- **Conexion a Internet y configuraciones basicas:**

En Mayix Stage4 actual, la deteccion de la tarjeta de red se realiza, por lo que no es necesario cargar nuestro modulo de tarjeta de red, pero si este no se encontrara deberemos de hacer para ello lo siguiente:

buscar el modulo de nuestra tarjeta de red en internet y cargarlo:

Code:

```
# modprobe nombre_modulo
```

Puedes usar el comando `modprobe -l | grep net` para listar todos los modulos disponibles para tarjetas de red. Si disponemos de cable o adsl monopuesto deberemos de introducir lo siguiente:

Code:

```
# dhcpcd eth0
```

Si por lo contrario disponemos de adsl multipuesto, deberemos de introducir:

Code:

```
# net-setup eth0
```

Es importante asegurarnos que tenemos conexion antes de seguir con la instalación, para ello mandaremos unos paquetes icmp a alguna web

importante que estemos seguros de que se encuentre operativa:

Code:

```
# ping -c 5 www.google.com
```

Si nos responde, tendremos correctamente configurada la conexión, si no es así, es que no hemos sido capaces de ello. Mirar entonces los manuales de route y de ifconfig. A continuación deberemos de asegurarnos que la hora introducida es la correcta:

Code:

```
# date
```

Si la hora introducida no es la correcta tendremos que poner la que rectificarlo, la sintaxis es date MMDDhhmmyy, siendo los 2 primeros números el mes, los otros dos el día, los otros dos la hora y los otros dos los minutos y por último el año, si quisieramos poner 5 de enero a las 14:31 del 2004 sería:

Code:

```
# date 0105143104
```

Para más información:

Code:

```
# date --help
```

- **Particionar :**

Para particionar el disco duro usaremos el comando fdisk, este comando crea destrulle particiones pero no es capaz de reasignar particiones. Si tenemos una partición que se encuentra ocupando todo el espacio de nuestro disco duro tenemos que reparticionarla con otro programa como pudiera ser Partition Magic o borrar la partición que tenemos y crearla nueva con menos espacio asignado. Notar que los dispositivos en linux se encuentran en el directorio /dev/ y tienen una terminología muy concreta. Los dispositivos ide se conocen como hdX ej: /dev/hda. La X es el modo en que se encuentra, es decir, si veis una placa base tiene el ide0 o ide master y el ide1 o ide esclavo, dentro de cada ide este a su vez tiene un ide master y un ide esclavo, por lo que el máximo de dispositivos ide que puedes conectar a la placa con 4:

```
hda -> ide0 cable master(maestro)
```

```
hdb -> ide0 cable slave(esclavo)
```

```
hdc -> ide1 cable master
```

```
hdd -> ide1 cable slave
```

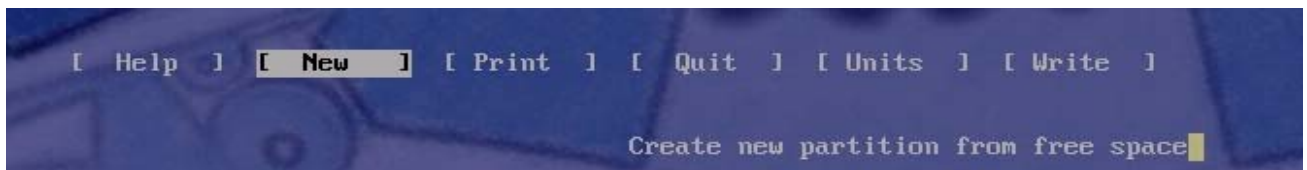
Vamos ahora a aprender como particionar también con cfdisk, de una manera mucho más fácil y visual.

Suponiendo que el disco que deseamos particionar será el primario master el comando a utilizar se debería de ver así:

```
# cfdisk /dev/hda
```



al iniciar el programa encontraremos en la parte superior la identificación de nuestro dispositivo, en la parte media el control de particiones, y en la parte inferior las opciones de control de cfdisk asi como se ve aquí



dentro de estos controles nos podremos mover con los cursores izquierda y derecha, los cursores de arriba y abajo se utilizan para movernos dentro de la parte media osea los controles de particiones, para crear una nueva partición utilizamos con los cursores derecha e izquierda la opcion new, despues nos preguntara el cfdisk si nuestra partion sera [primaria] [logica], esto queda a nuestro gusto, pero debemos tener en cuenta que solo podemos tener 4 primarias, luego de escoger el tipo de partición que se creara debemos de decirle a cfdisk cuanto espacio de nuestro disco duro va a ocupar la particon que estamos creando, por ultimo nos preguntara la posición en que deseamos tener nuestra partición al final o al principio del espacio disponible, seguido de esto se creara una nueva entrada en el control de particiones que se vera mas o menos asi:



en la imagen vemos que se han creado ya 2 particiones primarias utilizando el mismo metodo, y para empezar por primera vez en gentoo linux te recomiendo que unicamente crees 3 particiones, luego veremos como las diferenciaremos, por ultimo cuando ya tengamos nuestras particiones creadas tendremos que designarles el tipo de particiones que seran moviendonos con los cursores de arriba y abajo

seleccionando una por una las particiones creadas, cuando seleccionemos la primero utilizaremos los cursores derecha e izquierda para seleccionar la opcion [type] ahí se desplegara un listado de particiones posibles que puede crear cfdisk en este caso solo nos ocuparemos de la 82 y 83 veamos este cuadro:

dispositivo	punto de montaje	espacio	tipo
sda1 o hda1	boot	50 mb	82
sda2 o hda2	swap	variable	81
sda3 o hda3	root	el resto	82

por que la swap variable?

Desde hace mucho tiempo se utiliza una partion de intercambio de procesos, como una memoria extra pero en el disco duro, que historicamente se ha creado con un tamaño de el doble de la memoria fisica, 128 de ram = 256 mb swap, pero debido a muchos cambios que se han realizado al kernel y de ya no encontrarse computadores con tan poca memoria se ha descontinuado su uso.

Y listo ya tenemos nuestras particiones hechas con cfdisk!

- **Formatear particiones:**

Vamos a darles formato a las particiones. Existen actualmente 3 formatos principalmente, el ext3 que es el antiguo ext2 con journalist(si se apaga el ordenador no perdemos la información ni se producen daños), Reiserfs y XFS. En mi caso por pruebas la que mejor resultado me ha dado es la reiserfs, aunque ahi gente que opina que ext3 es mejor que reiserfs. XFS sera seguramente el mejor sistema de ficheros(formato) que ahi pero esta ahun un poco verde, por lo que no lo recomiendo.

Los pasos a seguir a continuación seran los siguientes:

1. Activamos el dma de nuestro hd por si no lo tubieramos, para aumentar la velocidad de nuestro discoduro:

Code:

```
# hdparm -d 1 /dev/hda
```

2. Damos formato a la particion swap:

Code:

```
# mkswap /dev/hda2
```

3. Acontinuación escogeremos nuestro tipo de ficheros, no voy a explicar la XFS por motivos citados anteriormente:

Code:

```
# mkfs.ext3 /dev/hda3 -> si queremos formatear la particion linux como ext3.
```

Code:

```
# mkfs.reiserfs /dev/hda3 -> si queremos formatear la particion como reiserfs.
```

- **Montar las particiones:**

Acontinuacion montaremos las particiones para poder ser usadas, esto es un concepto algo raro para los usuarios que no usen linux/unix. Montar una particion seria algo asi como activar una partición para poder ser usada. En el arranque de linux escribiremos en /etc/fstab las particiones que queremos que se arranque(tambien podemos decirle que no arranquen) con un conjunto de características como pueda ser acceso de solo lectura etc etc Esto lo veremos mas adelante pero es interesante ir sabiendolo ya.

Vamos a montar la particion swap:

Code:

```
# swapon /dev/hda2
```

Ahora montaremos la particion linux donde tendremos nuestra gentoo, tenemos que crear un directorio donde queremos montar la particion:

Code:

```
# mkdir /mnt/mayix -> creamos el directorio "mayix"
```

Ahora dependiendo del tipo de formato haremos:

Code:

```
# mount -t reiserfs /dev/hda3 /mnt/mayix ->montamos la nueva  
particion si la elejimos reiserfs
```

o

Code:

```
# mount -t ext3 /dev/hda3 /mnt/mayix ->montamos la nueva  
particion si la elejimos ext3
```

luego de esto crearemos un nuevo directorio llamado boot

Code:

```
# mkdir /mnt/mayix/boot -> creamos el directorio "boot"
```

Ahora montaremos la partición de boot

```
# mount -t ext3 /dev/hda1 /mnt/mayix/boot ->montamos la nueva  
particion si la elejimos ext3
```

El comando mount solo puede ser usado por el usuario root y su sintaxis basica es la siguiente:

```
mount -t tipo /dev/dispositivo /directorio/de/montage
```

Los tipos mas usados son los siguientes:

reiserfs -> para reiserfs

ext3 -> para ext3

vfat -> para fat32

ntfs -> para ntfs

iso9660 -> para cdroms

Para mas información man mount.

- **Stage:**

Como esta guía se ha modificado para ser mayix stage4 el centro de atención nos centraremos únicamente a referirnos al uso de el, omitiendo el uso del stage 1 2 3, el stage4 lo podras encontrar en /mnt/cdrom/stage/mayix-stage4.tar.bz2

Acontinuación vamos a desempaquetar y descomprimir el stage, con la opción p para que mantenga los privilegios originales del stage en que fue empacado:

Code:

```
# cd /mnt/cdrom/stage
```

podemos utilizar de 2 formas el stage4 para descomprimirlo

la primera:

Code:

```
# cp mayix-stage4.tar.bz2 /mnt/mayix -> en esta forma copiamos al disco duro primero el stage puede acelerar su desempacado  
# cd /mnt/mayix -> entramos al directorio  
# tar -xvjpgf mayix-stage4.tar.bz2 -> iniciamos el desempaquetado
```

la segunda:

Code:

```
#cd /mnt/mayix  
# tar -xvjpgf /mnt/cdrom/stage/mayix-stage4.tar.bz2 -> de esta forma nos evitamos copiar el stage al HD y luego desempacarlo, pero se puede perder tiempo tambien dependiendo de la velocidad de nuestro cdrom
```

- **Preparandonos nuestra partición:**

Montamos la proc del livecd(que se cargo y se creo con el kernel del live-cd) en nuestra gentoo:

Code:

```
# mount -t proc /proc /mnt/mayix/proc
```

Copiamos los DNS que nos permitan resolver dominios de internet:

Code:

```
# cp /etc/resolv.conf /mnt/mayix/etc/resolv.conf
```

Cambiamos nuestra raíz del live-cd por lo nuestra particion donde tenemos la gentoo. Notar que es interesante que leamos el punto 9 antes de pasar a este punto, puesto que los programas a usar una vez hecho el chroot para edicion de texto pueden resultar complejos(nano y vi):

Code:

```
# chroot /mnt/mayix /bin/bash
```

Cargamos las configuraciones de nuestro usuario, de lo que sera nuestra futura gentoo:

Code:

```
# env-update  
# source /etc/profile -> cargamos la configuración del profile
```

Este apartado sobre portage y emerge lo veremos como una introducción al manejo de gentoo en su paquetería, aunque en la instalación de mayix-stage4 no lo utilizaremos.

- **Portage y emerge:**

El portage es el gestor de paquetes de gentoo(perdonarme los expertos, por ser tan poco específico). La herramienta usada para la gestión de este es el comando emerge.

El portage se encuentra en /usr/portage y su configuración la tenemos en el fichero /etc/make.conf que veremos en el siguiente punto.

El portage está ordenado por subcategorías, y dentro de cada categoría su correspondiente herramienta/programa.

Los ficheros descargados, que en su mayoría son fuentes se descargan en el fichero /usr/portage/distfiles.

Los comandos más usados del emerge son los siguientes(un paquete es un programa o agrupación de estos):

```
# emerge -s paquete -> busca en el portage "paquete"
# emerge paquete -> Instala el paquete
# emerge -p paquete -> Nos muestra las dependencias que necesitamos
instalar para instalar el paquete
# emerge -f paquete -> Se descarga el programa pero no lo compila
# emerge -u paquete -> Actualiza el paquete
# emerge -C paquete -> Desinstala el paquete
# emerge sync -> Actualiza la lista de paquetes del portage
# emerge world -> Reinstala todo el sistema(No lo uses jamás, por la vida de
tu CPU xD)
```

```
# emerge -u world -> Actualiza la distribución
# emerge -b paquete -> Compila el programa, lo instala y nos genera el binario
# emerge -B paquete -> Compila el programa y nos genera el binario, usado si
tenemos dos máquinas y una es muy potente y la otra no lo es
# emerge -k paquete -> Instala un paquete binario anteriormente compilado
```

Muchas de las preguntas acerca de gentoo, va relacionado con el tiempo que tarda en compilar un programa y si realmente puede ser usado para un servidor. Para intentar solventar esta cuestión gentoo permite que una vez compilado un paquete podamos crear el binario optimizado para nuestra plataforma. Gracias a ello podemos tener nuestros programas optimizados para nuestra plataforma, con lo que, si hubiera algún tipo de fallo y necesitarámos reinstalar un programa únicamente deberíamos de "cargar" el binario que creamos cuando compilamos la primera vez.

Code:

```
# emerge -b paquete
```

(Y si necesitamos reinstalarlo con poner lo siguiente nos instalará el binario que hemos creado)

Code:

```
# emerge -k paquete
```

En gentoo existen unos paquetes, que nos aparecen como [MASKED], esto puede ser por 2º motivos:

1º)

El paquete es muy nuevo y se encuentra como "inestable" puesto que no se le ha realizado las suficientes pruebas, esto pasara si no aceptamos la rama inestable de gentoo en el make.conf. Con poner el siguiente comando lo instalaremos:

Code:

```
# ACCEPT_KEYWORDS="~x86" emerge paquete
```

2º)

El programa tiene algun tipo de error que ahun no ha sido solucionado por sus programadores, como por ejemplo un error de seguridad.

Podemos verlo en /usr/portage/profiles/package.mask, si deseamos instalarlo de todas formas, unicamente con comentar en el fichero el paquete sobra. No lo recomiendo nunca.

Code:

```
# nano /usr/portage/profiles/package.mask
```

```
# emerge paquete
```

El emerge nos muestra la version de la aplicación que se dispone actualmente, pero si queremos instalar una versión anterior o posterior a la que nso ofrece, deberemos de ir al portage y emergerla. Un ejemplo seria el siguiente:

Code:

```
# emerge -s gentoo-source
```

```
Searching...
```

```
[ Results for search key : gentoo-source ]
```

```
[ Applications found : 1 ]
```

```
* sys-kernel/gentoo-sources
```

```
Latest version available: 2.4.26_pre6
```

```
Latest version installed: [ Not Installed ]
```

```
Size of downloaded files: 32,178 kB
```

```
Homepage: http://www.kernel.org/ http://www.gentoo.org/
```

```
Description: Full sources for the Gentoo Kernel.
```

```
License: GPL-2
```

```
# cd /usr/portage/sys-kernel/gentoo-sources/
```

```
# ls
```

```
ChangeLog gentoo-sources-2.4.22-r3.ebuild
```

```
gentoo-sources-2.4.26_pre5.ebuild
```

```
files gentoo-sources-2.4.22-r5.ebuild gentoosources-
```

```
2.4.26_pre6.ebuild
```

```
gentoo-sources-2.4.19-r13.ebuild gentoo-sources-2.4.22-r9.ebuild
```

```
Manifest
```

```
gentoo-sources-2.4.20-r16.ebuild gentoo-sources-2.4.25-r2.ebuild
```

```
metadata.xml
```

```
gentoo-sources-2.4.20-r7.ebuild gentoo-sources-2.4.26_pre3.ebuild
```

```
# emerge gentoo-sources-2.4.25-r2.ebuild
```

Podríamos haber emergido cualquiera de las versiones que aparecían con el `.ebuild`, en nuestro caso instalaríamos un kernel más antiguo que el que nos ofrecía el emerge. Esto es algo único de Gentoo, puesto que en la mayoría de distribuciones solo te ofrecen una versión del programa. Existe una aplicación que nos permite mostrar el tiempo que tardamos en compilar un programa. Realmente es un script del log del portage, es como para hacernos a la idea de lo que puede tardar una aplicación en compilar, el programa se llama `genlop`:

Code:

```
# emerge genlop
# genlop -t paquete
```

Para más información acerca de las posibilidades que tiene `genlop`: `genlop -help`. Una nota importante, mirando para un futuro, son las actualizaciones en Gentoo, una vez actualicemos Gentoo mediante las siguientes dos instrucciones:

Code:

```
# emerge sync
# emerge -u world
```

Puede ocurrir que hayamos actualizado paquetes que poseen uno o más ficheros de configuración. Gentoo, a diferencia de otras distribuciones, no nos va a escribir encima de esos ficheros, sino que usaremos una utilidad/script que nos permite el control de estas actualizaciones, esta utilidad/script es:

Code:

```
# etc-update
```

Una vez lo ejecutemos nos aparecerán todos los ficheros de configuración que pueden ser modificados precedidos de un número), y al final el siguiente código:

Code:

Please select a file to edit by entering the corresponding number.

(-1 to exit) (-3 to auto merge all remaining files)

(-5 to auto-merge AND not use 'mv -i'):

Lo que tendremos que hacer es introducir el número) (solo el número sin signo delante) del fichero a configurar, y entonces nos aparecerá un 2º menú de configuración, donde se nos permite hacer 3 cosas: hacer una copia de seguridad del actual y reescribir el fichero, o reescribir el fichero sin hacer copia de seguridad, o no modificar el fichero y dejar nuestro antiguo fichero de configuración.

- **Modificamos el fichero `make.conf`:**

El fichero `/etc/make.conf` es el fichero de configuración del portage, donde le pasamos los flags de la compilación, los USE o opciones/modulos con que queremos que nos instale el programa. Así como configuración de nuestro portage. El fichero `make.conf` comienza con un `USE="opcion1 opcion2 opcion 3..."`, esto es una de las cosas que hace que Gentoo sea especial, gracias a esto, cuando compilamos una aplicación cargamos lo que nos interesa únicamente, así como cargamos módulos que de forma predeterminada no están accesibles. Un ejemplo sería el soporte de sonido. El soporte tradicional OSS va a ser substituido por el ALSA, pero muchas de las aplicaciones compiladas(binarios) y lista para ser

instaladas, siguen sin tener soporte de ALSA pero si de OSS. Pues introduciendo en el USE "alsa -oss" nos compilara los programas con soporte alsa y no con soporte oss. Como hemos podido apreciar introduciendo el caracter "-" delante de la opcion, hacemos lo contrario, es decir, le decimos que no tenga esa opción. Esto es muy usado por ejemplo por los amante de gnome que no quieren tener librerias de kde puesto que no las van a usar y son de gran tamaño ponen la opcion "-kde".

Acontinuación voy a poner un conjunto de USE. No estaran todos pero si los mas usados:

->dvdr y cdr para poder grabar dvd y cdrom respectivamente

->vcd,dvd para reproducir peliculas para video cd o video dvd

->mmx, sse y 3dnow son para optimizacion para usar estas instrucciones, causaban bastantes fallos(Especialmente mmx)

->x,opengl,dga para el motor de las X(entorno grafico), si tienes tarjeta 3dfx inclullelo tambien

->xv,fbcon, directfb, gpm para consola(framebuffer, raton, peliculas y demas)

->oss o arts o alsa para el sonido dependiendo de tu "eleccion de sonido"

->gphoto2 por si tienes camara digital

->kde,qt y arts por si usas kde(si no lo vas a usar con un simple -kde desactivaras el soporte)

->gnome, gtk, bonobo, gbm gtkhtml por si usas gnome(si no lo vas a usar con un simple -gnome desactivaras el soporte)

->aalib,ncurses,sd1 y svg librerias graficas en consola(recomendado)

->gif,jpeg,png y tiff soporte formatos

->avi,mpeg,encode y quicktime librerias de video(si quieres ver videos o codificar formatos multimedia)

->java,flash,perl,python,slang,guile,libg++ librerias de programacion(permite el

soporte a ciertos lenguajes de programación)

-> pam, ssl, crypt seguridad (directamente ya tardas en ponerlas)

-> spell, truetype, pdflib, tetex para fuentes corrección ortográfica etc...

-> nls soporte para más de un idioma

-> samba soporte para samba (para windows)

-> tcpd, pam, ssl directivas de seguridad (obligado)

Existe una utilidad que facilita la elección de los USE, así como una pequeña explicación de la función de cada USE, esta utilidad es `ufed`:

Code:

```
# emerge ufed
# ufed
```

- **Continuando...:**

Actualizamos la lista de paquetes de nuestro portage, en nuestro caso como es la primera vez, la generamos nueva:

```
#emerge sync -> para esto si requiere conexión a internet, pero por el momento lo podemos obviar
```

Establecemos la hora de nuestra gentoo de la siguiente forma: `ln -sf /usr/share/zoneinfo/path/to/timezoneinfo /etc/localtime`, para Guatemala sería así:

Code:

```
# ln -sf /usr/share/zoneinfo/America/Guatemala /etc/localtime
```

Y por último introduciremos nuestro password de root, aquí poner importancia en que sea un password o clave que este compuesto tanto por mayúsculas como minúsculas, así como con números y con signos, notar que gentoo usa MD5/FreeBSD lo que es un gran avance frente al obsoleto DES. Un pequeño truco es escribir una frase que nos acordemos y cambiar las vocales por números, siendo a-> 4, e->3, i->1, o->0 u->w y la primera letra y última de una palabra, si no es un número o "una vocal cambiada" con el truco anteriormente citado, ponerla en mayúsculas y las demás letras en minúsculas.

Por ejemplo:

-> mi camion es grande

-> m1 c4m10n 3s gr4nd3 *No se puede dejar espacios en blanco

-> m1-c4m10n-3s-gr4nd3

-> M1-C4m10N-3S-Gr4nd3

Como ves de una frase tonta obtenemos una contraseña de cierta dificultad para ser descifrada, para más información recomendaría una buena guía de

seguridad. El comando para cambiar el password seria este:

Code:

```
# passwd
```

Si aparece un mensaje de este estilo BAD PASSWORD: it is based on a dictionary word, es que no usaste un passwd fuerte(seguro), aunque si que acepte que lo introduzcas, el programa te advierte de la mala idea de usar una contraseña de ese tipo.

- **Compilación de nuestro kernel:**

Gentoo dispone de tanto kernels 2.4.2X, como kernels 2.6.X, así como pequeñas kernels donde se han cogido al kernel vanilla(kernel básico sin ningún tipo de modificación) y se le ha añadido un conjunto de parches. En principio estos kernels funcionan mejor puesto que son iguales que al tradicional y aparte le han añadido varios parches que nos pueden ser útiles como el bootsplash. El problema que tienen es que los kernel gentoo suelen ir unas revisiones de versiones o una versión como mucho por detrás de los oficiales, es decir si el oficial es el 2.6.6 puede darse que el gentoo sea el 2.6.5.rc3

2.6 de gentoo -> sys-kernel/gentoo-dev-sources
2.4 de gentoo -> sys-kernel/vanilla-sources
2.6 básico -> sys-kernel/development-sources
2.4 básico -> sys-kernel/vanilla-sources

mayix-stage4 contienen un solo kernel en su base, que es el gentoo-sources de la serie 2.4, eso no quiere decir que luego de la instalación no puedas bajarte los de la serie 2.6, pero si necesitaras compilar este, para luego bajarte los 2.6

Gentoo tiene muchas herramientas para hacerte fácil el ingreso al mundo linux, y si te da algo de pánico compilar tu mismo el kernel puedes utilizar una herramienta que lo hace por ti.

Code:

```
# genkernel all
```

y te realizará todo los procesos que necesitas para tener un kernel funcional, en esta etapa de desarrollo de genkernel ya trabaja muy bien!.

Pero si eres de los que les gusta el riesgo y quieren compilar a "mano" su propio kernel accedemos al directorio donde se suele encontrar las fuentes del kernel([i]/usr/src/) en el cual tendremos o deberemos de crear un link o enlace dinámico("acceso directo" de los windows) de las sources del kernel que estemos usando, en nuestro caso, al ser recién instalado solo tendremos un directorio.

Code:

```
# ls /usr/src
```

```
linux-2.4.26-gentoo-xx
```

```
# ln -s /usr/src/linux-2.4.26-gentoo-xx /usr/src/linux *(si tuvieramos el enlace no haria falta este paso)
```

```
# cd /usr/src/linux
```

Como vemos en el caso de mayix ya tenemos el link suave hacia el source del kernel. A continuación elijeremos el mejor menu para configurar nuestro kernel. Existen 3º tipos, el mas simple es el config, el siguiente es menuconfig el cual usa las ncurses y es muy facil de usar(recomiendo este) y por ultimo el xconfig[i] que se usa en las X(motor de los entornos graficos de linux), al tener gentoo recién instalado no disponemos todabia de las X por lo que la mejor opcion seria el #menuconfig.

Code:

```
# make config  
0
```

Code:

```
# make menuconfig  
0
```

Code:

```
# make xconfig
```

Ahora viene la parte mas dificil o compleja del kernel que es la configuración de este. Voy a poner unas opciones basicas para un usuario medio, nada especifico, si alguien quiere algo mas especifico que se lea cualquier manual que ahi del kernel. No es prioridad en la instalación de Linux Gentoo la configuración de un kernel y acordaros que existe el genkernel para no tener que entrar aqui.

Pondre unicamente las opciones escogidas. Puede que existen opciones que no esten correctamente, espero mejorarlo mas adelante. Puede ocurrir que tengas opciones identicas ya que, esto evoluciona, el kernel usado para el ejemplo es gentoo-sources-2.4.26 de la serie 2.4.

```
Code maturity level options --->  
[*] Prompt for development and/or incomplete code/drivers"  
(You need this to enable some of the options below)  
...  
File systems --->  
<*> Reiserfs support  
(Only needed if you are using reiserfs)  
...  
<*> Ext3 journalling file system support  
(Only needed if you are using ext3)  
...  
[*] Virtual memory file system support (former shm fs)  
(Required for Gentoo Linux)  
...  
<*> JFS filesystem support  
(Only needed if you are using JFS)  
...  
[*] /proc file system support  
(Required for Gentoo Linux)  
[*] /dev file system support (EXPERIMENTAL)  
[*] Automatically mount at boot  
(Required for Gentoo Linux)  
[ ] /dev/pts file system for Unix98 PTYs  
(Uncheck this, it is not needed unless you use a 2.6 kernel)  
...  
<*> Second extended fs support  
(Only needed if you are using ext2)  
...  
<*> XFS filesystem support  
(Only needed if you are using XFS)
```

Una vez compilado deberemos de guardar nuestra configuración en la opción de: Save Configuration to an Alternate File, con el nombre que nosotros deseemos. Y a continuación compilaremos el kernel y sus módulos, así como copiaremos el kernel compilado en el directorio boot preparado para ser cargado.

Code:

```
# make bzImage
# make modules
# make modules_install
# cp ./arch/i386/boot/bzImage /boot
```

A continuación tendríamos que editar el fichero `/etc/modules.autoload` y poner los módulos necesarios que queremos que se carguen en el arranque del kernel. Notar que solo ahí que poner el nombre y no `modprobe` ni `insmod`:

Code:

```
# nano /etc/modules.autoload
```

Un ejemplo de fichero `modules.autoload` sería este:

Code:

```
# cat /etc/modules.autoload
usb-storage
nvidia
8139too
iptables_filter
iptables_mangle
iptables_nat
ip_conntrack_irc
ipt_limit
ipt_state
ip_nat_irc
#
```

Existe una utilidad que tiene los módulos ordenados por categoría, además de una explicación sobre cada uno de ellos que nos puede facilitar la tarea ciertamente. Esta utilidad es muy usada sobretodo por la gente que pasa de Debian a Gentoo, hacer mención de que la utilidad está aún un poco verde:

Code:

```
# emerge modconf
# modconf
```

Si esta herramienta no funcionara correctamente, puesto que en ocasiones ocurre, una buena opción para editar el fichero sería usar la opción `-l` del `modprobe` para listar todos los posibles módulos que podemos cargar:

Code:

```
# modprobe -l
```

Ahora deberiamos de cargar nuestro sistema de arranque. Existen principalmente **dos**, LiLO y **GRUB**, en mi caso siempre he usado GRUB por lo que usaremos este, puesto que me parece mas facil de entender.

Code:

```
# grub
```

Y por fin ya hemos terminado, ves como no es tanto como parece ;).

Cuando ejecutemos este comando aparecera un pequeño prompt en el cual trabajaremos la instalacion de este loader

MBR = master boot record

```
grub> root (hd0,0) (direccion de tu particion boot)
```

```
grub> setup (hd0) (donde va a ser instalado el grub , aquí esta para el MBR)
```

instalando el grub no en el MBR

```
grub> root (hd0,0) (tu particion boot)
```

```
grub> setup (hd0,4) (donde el boot va a ser instalado ; ejemplo: en el /dev/hda5)
```

```
grub> quit
```

luego buscamos en /etc/boot/grub/
el archivo de configuración grub.conf

si utilizaste el genkernel se podria ver algo asi

```
default 0
timeout 30
splashimage=(hd0,0)/boot/grub/splash.xpm.gz

title=My example Gentoo Linux
# Si usaste genkernel la configuracion se podria ver algo asi:
title=My example Gentoo Linux (genkernel)
root (hd0,0)
kernel (hd0,0)/boot/kernel-KernelVersion root=/dev/hda3
initrd (hd0,0)/boot/initrd-KernelVersion
# Para las personas que necesita un boot tambien con windows
title=Windows XP
root (hd0,5)
chainloader (hd0,5)+1
```

Si compilaste tu kernel a "mano" el grub.conf se podria ver algo asi:

```
default 0
timeout 30
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
```

```

title=My example Gentoo Linux
root (hd0,0)
kernel (hd0,0)/boot/bzImage root=/dev/hda3

# Para las personas que necesita un boot tambien con windows
title=Windows XP
root (hd0,5)
chainloader (hd0,5)+1

```

Esta seccion tambien la utilizaremos para conocimientos basicos del funcionamiento de gentoo linux!

- **Runlevels en Gentoo:**

Los runlevels en gentoo siguen otro sistema distinto al acostumbrado con los linux tradicionales. Los runlevels en gentoo se dividen en:

- > boot #Programas que se cargan en el arranque seguido del kernel ej: alsa
- > default #Programas que se cargan una vez cargado el kernel y toda la distribucion, justo antes de proporcionarte una prompt de login. ej: apache
- > nonetwork #Programas que se cargan antes de la conexion a inet
- > single #Desconozco su uso Ò.Ó!

Se encuentran en /etc/runlevels. Los programas se cargan usando la siguiente sintaxis: rc-update add programa/demonio runlevel siendo runlevel cualquiera de los 4 expuestos anteriormente y programa la implementación de programa que se encuentra en /etc/init.d/. Por ejemplo para el apache2:

Code:

```

# rc-update add apache2 default
* apache2 added to runlevel default
* Caching service dependencies...
* rc-update complete.
#

```

Para eliminarlos pondremos unicamente: rc-update dell programa/demonio, sin especificar el runlevel , puesto que el rc-updateya lo busca en los 4 posibles, un ejemplo con el apache2 seria:

Code:

```

# rc-update del apache2
* apache2 removed from the following runlevels: default
* Caching service dependencies...
* rc-update complete.
#

```

Al cargarlos deben de tener una implementación en en /etc/init.d/. Si queremos cargar un fichero con un script de shell que no este implementado en el /etc/init.d deberemos de usar los ficheros local.start y local.stop que se encuentran en /etc/conf.d/

Para ver todos los posibles programas(Servicios) que tenemos arrancados, asi como los que disponemos pero no los tenemos asignados a ningun runlevel:

Code:

```

# rc-status -a

```

Como habeis visto, cuando instalais un programa que necesita de los runlevels,

para la ejecución de un demonio/servicio/etc se genera su pertinente implementación en el directorio /etc/init.d/ pero además existen unos pequeños scripts localizados en /etc/conf.d que deberían de ser usados, para por ejemplo el uso de iptables. A mi gusto esta es la zona donde gentoo peca más de falta de "edad"(junto con el emerge).

Recordar que muchos de los programas que cargamos en los runlevels son demonios/servicios. Estos servicios se cargan automáticamente al haber sido añadidos con el rc-update al runlevel. Si queremos pararlos, reiniciarlos o volverlos a arrancar una vez parados, un ejemplo sería con el ssh, apache, proftpd etc. se usa los siguientes comandos:

```
# /etc/init.d/servicio stop ->Detiene la ejecución
```

```
# /etc/init.d/servicio start ->Arranca de nuevo el demonio/servicio, si ya está arrancado dará error
```

```
# /etc/init.d/servicio restart ->Rearranca el servicio, es equivalente a hacer un stop y un start
```

Un ejemplo de lo anteriormente citado sería este:

Code:

```
# /etc/init.d/apache2 stop
* Stopping apache2... [ ok ]
DarkTemplar root # /etc/init.d/apache2 start
* Starting apache2... [ ok ]
DarkTemplar root # /etc/init.d/apache2 restart
* Stopping apache2... [ ok ]
* Starting apache2... [ ok ]
#
```

Los servicios que necesitamos agregar en mayix-stage4 a los run levels

```
#rc-update add vixie-cron default
#rc-update add metalog default
#rc-update add xdm default
#rc-update add numlock default
```

- **Creación de usuarios:**

La creación de usuarios debe de ser obligatorio aunque solo usemos el ordenador una persona. El usuario ROOT solo debe de usarse cuando tengamos la necesidad de configurar/administrar nuestro sistema, puede parecer algo engorroso, pero os dais cuenta que siendo usuarios normales si cometéis algún tipo de error, si no soys root, es difícil que dañéis el sistema, además que no siendo root evitáis problemas como pueden ser: troyanos, gusanos, etc etc

Para la creación de ficheros podemos hacerlo a "mano" usando los comandos de:

Code:

```
# adduser nombre_usuario  
# passwd nombre_usuario  
# mkdir /home/nombre_usuario
```

A continuación debemos de comprobar si el usuario esta creado, y verificar que fue correctamente. Para ello editaremos el fichero /etc/passwd y verificaremos que el usuario sigue una estructura asi(se encontrara en la ultima linea): nombre_usuario:x:1000:100::/home/nombre_usuario:/bin/bash (el valor 1000 es el identificador del usuario, puede ser 1000 o superior).

Code:

```
# cat /etc/passwd | grep usuario  
usuario:x:1000:100::,/home/usuario:/bin/bash  
#
```

Ahora deberemos de agregar el usuario a los grupos que nos interese añadir, es importante, remarcar que ahí un grupo en gentoo(y en otras distribuciones de linux/unix muy seguras) que tienen el grupo wheel el cual es necesario tener a nuestro usuario si queremos que sea capaz de usar el comando su - para tener acceso como root, sin tener que habrir una consola nueva. Los grupos que seria logico añadir al usuario serian: wheel, audio, cdrom, games, cdrw, usb. Para añadir usuarios seria: grupo:usuario1,usuario2,usuario3.... Un ejemplo:

Code:

```
cat /etc/group | grep wheel  
wheel::10:root,usuario  
#
```

Gentoo implementa un script para la edición facil y rapida de usuarios. Se llama superadduser. Yo os lo recomiendo puesto que os ahorrara tiempo. el script se encuentra en /usr/sbin/superadduser. Vamos a emergerlo:

Code:

```
# superadduser  
Login name for new user []: nombre  
User ID ('UID') [ defaults to next available ]: (enter) *(pulsar  
intro, dejamos la opcion predeterminada o default)  
Initial group [ users ]: (enter)  
Additional groups (comma separated) []:  
wheel,audio,cdrom,games,cdrw,usb  
Home directory [ /home/nombre ] (enter)  
Shell [ /bin/bash ] (enter)  
Expiry date (YYYY-MM-DD) []: (enter)  
New account will be created as follows:  
-----  
Login name.....: nombre  
UID.....: [ Next available ]  
Initial group....: users  
Additional groups: wheel,audio,cdrom,games,cdrw,usb  
Home directory...: /home/nombre  
Shell.....: /bin/bash  
Expiry date.....: [ Never ]
```

This is it... if you want to bail out, hit Control-C. Otherwise, press ENTER to go ahead and make the account.

(enter)

Creating new account...

Changing the user information for nombre

Enter the new value, or press ENTER for the default

Full Name []: (enter)

Room Number []: (enter)

Work Phone []: (enter)

Home Phone []: (enter)

Other []: (enter)

New UNIX password: (password)

Retype new UNIX password: (password)

passwd: password updated successfully

Account setup complete.

#

Como ves es bastante facil la creación de usuarios con este script.

- **Configuración de ficheros basicos de gentoo:**

Ahora modificaremos el fichero /etc/fstab donde le indicamos a ciertos dispositivos su punto de montaje asi como un conjunto de propiedades. El ejemplo basico de configuración de este fichero es: /dev/dispositivo /punto/de/montaje tipo opciones 0 0. Los tipos mas usado son los mencionados en el principio de la instalación de gentoo: reiserfs, ext3, iso9660, ntfsm vfat ... En el caso de la particion raiz deberemos de poner la opcion noatime, si por lo contrario es una particion de tipo ext3 deberemos de poner defaults.

Las opciones mas usadas son las siguientes:

->sw #Usado para particiones swap

->default #Esta es un conjunto de opciones: rw, suid, dev, exec, auto, nouser, y async

->auto #Se monta en el arranque

->noauto #No se monta en el arranque, muy usado en los cdrom, ya que puede darse que no tengamos un cdrom introducido

->ro #Solo lectura

->rw #Lectura y escritura

->exec #Permite la ejecución de binarios

->noexec #No permite la ejecución de binarios

->noatime #No actualizar los tiempos de acceso a nodo-í en este sistema de ficheros

->users #Permite que los usuarios puedan montar la partición
Un ejemplo de fichero /etc/fstab seria este:

Code:

```
# nano /etc/fstab
#Dispositivo #Punto de montaje #tipo #opciones
/dev/hda1 / reiserfs noatime 0 0
```

#Ejemplo de particion Swap:
/dev/hda2 none swap sw 0 0

#Ejemplo de particion en otro HD:
/dev/hdb1 /ftp reiserfs noatime,users 0 0

#Ejemplo de particion en windows con ntfs:
/dev/hda1 /mnt/win ntfs ro,auto 0 0

#Ejemplo de particion en windows con vfat:
/dev/hda1 /mnt/win vfat users,auto,rw 0
0

#Montage de un cdrom:
/dev/cdroms/cdrom0 /mnt/cdrom iso9660
noauto,ro,users 0 0

#Sistema, necesario para los virtual filesystem:
proc /proc proc defaults 0 0
tmpfs /dev/shm tmpfs defaults 0 0
#

Nota: al poner la opcion users, como usuario normal puedes montar la partición pero sin hacer objetivo ni a tipo ni a dispositivo, por ejemplo, en el caso del cdrom, para montarlo como usuario normal tendremos que introducir lo siguiente:

Code:

```
# mount /mnt/cdrom *Si el punto de montaje /mnt/cdrom esta en el
fstab y tiene la opcion de user/s entonces si podremos montar
Pero no podemos indicar ni opciones ni dispositivo:
```

Code:

```
# mount -t iso9660 /dev/cdroms/cdrom0 /mnt/cdrom *solo el
usuario root puede hacer eso
```

Para mas información man mount y man fstab.

Acontinuación usaremos un programa que transforma un texto en otra cadena

texto mediante caracteres ascii y introduciremos el resultado en /etc/motd para que cada vez que nos logueemos aparezca dicho mensaje:

- **Configuraciones del entorno grafico:**

Que tal la nueva gentoo?

Una vez tengamos ya cargada la nueva gentoo, lo que normalmente se suele hacer es ponerse a instalar los programas que necesitemos mediante el emerge. Asi como la configuración de dispositivos como tarjetas de sonido, aceleración grafica, internet, etc y sobretodo nuestro entorno grafico tan querido.

Comencemos con la instalación del motor de nuestro entorno grafico, las Xfree(apartir de ahora las X:

Code:

```
emerge xfree -> esto ya esta hecho en mayix-stage4
```

Despues de un cierto tiempo, tendremos listo nuestro motor para ser configurado y usado. El archivo de configuracion de las X se llama /etc/X11/XF86Config, existen un gran numero de utilidades para poder configurarlo, entre ellas destaco una herramienta facil de usar y configurar:

Code:

```
# xf86config
```

Como vereis con este menu las preguntas son intuitivas y faciles de contestar. Vamos a ver un ejemplo para ver como lo deberia de usar:

Code:

```
# xf86config
```

pero este script es algo tedioso y engorroso para los nuevos en el sistema es por eso que:

Entra mayix-stage4 otra vez al rescate....

Un truco bastante bueno para tener el XF86config hecho de forma automatica y con la mejor resolucion lilsta para utilizar es

Agregar a los runlevels el siguiente servicio que ya viene incluido en mayix-stage4

```
#rc-update add mkxf86config default
```

luego eso eso como ya habiamos agregado el gdm en nuestro run level, se podra iniciar el sistema grafico automáticamente en cuanto reiniciemos por primera vez nuestro sistema.

Luego de eso es recomendable que quites el mkxf86config del runlevel default

```
# rc-update del mkxf86config default
```

y listo ya esta todo con este archivos

para para finalizar modificamos en archivo /etc/rc.conf

descomentariando la seccion
XSESSION="gnome"

Y cambiando la seccion
DISPLAYMANAGER="xdm" por "gdm"

- **Reiniciar:**

Ahora saldremos de nuestra flamante y nueva gentoo y reiniciaremos para cargar con nuestro nuevo kernel. Acordaros de volver a poner en la bios el arranque desde disco duro y de retirar el cd live.

Code:

```
# exit  
# umount /mnt/mayix/proc  
# umount /mnt/mayix/boot  
# umount /mnt/mayix  
# reboot
```



Esta seccion es extra para cuando ya tenemos acceso a internet o no tanta prisa por tener nuestro sistema corriendo

Lo primero que debemos de hacer es un
emerge sync

o podemos bajar un snapshot del portage posicionados en

```
#cd /usr/  
#wget gentoo.osuosl.org/snapshots/portage-20040919.tar.bz2 -> el archivo  
portage-20040919.tar.bz2 puede y debe variar segun la fecha en que se realice esta  
actualizacon  
# pwd -> esto no dara la seguridad de que estamos en el directorio correcto  
/usr/  
# tar -xvjpgf portage-20040919.tar.bz2  
Listo !!
```

- **Sonido mediante ALSA:**

Vamos a configurar nuestra tarjeta de sonido mediante ALSA. Lo primero que deberemos de hacer es saber exactamente cual es nuestra tarjeta de sonido:

Code:

```
# cat /proc/pci | grep audio
```

Una vez sepamos cual es, buscaremos información acerca de nuestro modulo en <http://www.alsa-project.org/alsa-doc/> , y en la opción de Choose manufacturer for more details buscar el modulo de la tarjeta de sonido. Una vez localizado deberemos de apretar sobre la opción de Driver & Docs. Ahi podremos leer toda la información de nuestro modulo, asi como tendremos un pequeño manual de configuración basico.

Editaremos nuestro make.conf y añadiremos(si no lo hicimos anteriormente) la opción de alsa y de esd(si eres usuario de sound blaster no lo pongais).

Ademas de añadir una variable de entorno, donde le indicamos cual es el modulo de tarjeta de sonido que tenemos(omitimos el snd-card- de delante del nombre del modulo):

Code:

```
# nano /etc/make.conf  
USE="... alsa esd"
```

...

```
ALSA_CARDS="nodulo" #ej: ALSA_CARDS="intel8x0"
```

```
#
```

Acontinuación editaremos nuestro kernel, si no lo hemos hecho, para el soporte de alsa, tal y como indicamos en el capitulo del kernel.

Ahora deberemos de emerge los siguientes paquetes:

Code:

```
# emerge alsa-lib  
# emerge alsa-utils  
# emerge alsa-tools
```

```
# emerge alsamixergui
# emerge alsa-oss
```

Siguiendo la información de la web anteriormente citada, deberemos de copiar exactamente igual como esta el fichero .asoundrc en la home de cada usuario, o crear un fichero general para todos en /etc/asound.conf:

Code:

```
# nano /etc/asound.conf
#Lo creamos tal y como aparece en la web
```

...

```
#
```

A continuación editaremos el fichero /etc/modules.d/alsatal como indica la web:

Code:

```
# nano /etc/modules.d/alsa
#Lo creamos tal y como aparece en la web
```

...

```
#
```

Existen una alternativa a la edición del fichero /etc/modules.d/alsa y es usar la herramienta alsacnf:

Code:

```
# alsacnf
```

Por ultimo introduciremos el alsa en los runlevels y cargaremos el demonio:

Code:

```
# rc-update add alsasound boot
# /etc/init.d/alsasound start
```

Para gestionar el volumen de los altavoces, así como activar el 5.1 o variantes del estilo, usaremos la herramienta alsamixergui :

Code:

```
$ alsamixergui
```

Ahora solo falta con decirle a los programas que usen de plugin de sonido, ALSA o OSS(tenemos emulación del oss mediante ALSA).

Existen un problema originado por el Full duplex, es decir, oír dos sonidos procedentes de 2 programas distinto a la vez. Esto ocurre en la mayoría de tarjetas de sonido(en las sound blaster no ocurre). Unas veces son porque no aceptan y otras veces es porque el modulo no esta suficientemente bien programado, ya que, el full duplex no es mediante hardware, sino mediante software. Para conseguir este efecto, usaremos el esound:

Code:

```
# emerge esound
# rc-update add esound default
# /etc/init.d/esound/start
```

En este caso deberemos de cambiar el plugin de sonido a usar por esound en todos los programas, y con ello dispondremos de full duplex. En ocasiones tendremos problemas ya que, en algunos programas no existe el soporte esound.

Bibliografía

- www.gentoo.org
- <http://forums.gentoo.org>
- www.gentoo-es.org
- www.knoppix.com
- www.distrowatch.com
- www.vidalinux.com
- www.vivalinux.com.ar
- www.slashdot.com
- www.osnews.com
- www.kernel.org
- packages.gentoo.org
- www.gentoo-la.org

Agradecimientos

Son muchas las personas a las cuales debo de agradecer por el apoyo brindado a este proyecto:

Pero principalmente quiero agradecer a mi hermano Jorge Luis Alvarado †, por haberme inculcado el valor del aprendizaje y la sagrada costumbre de estar siempre actualizado y de bajar de lo nuevo lo mejor, a ir siempre un paso adelante, haberme contagiado del amor por los computadores.

A carlitos de vidalinux por el hosting y mucho conocimientos sobre redes y autoconfiguración que hemos compartido

A Guillermo abate de gentoo-la por el apoyo incondicional, en este proyecto tanto intelectual como moral

A Bass y fernando de gentoo-es por el hosting para nuestra imagen iso que tanto nos costo encontrar

Y otras personas que anónimamente siempre dejan sus conocimientos plasmados dentro los foros de gentoo.org

A lugusac por haberme encaminado en este maravilloso mundo de linux

Al final y pero el mas importante:

A mis padres por aguantarme tantas noches de desvelo y tantos fines de semana sin salir de mi laboratorio para poder hacer realidad este proyecto.

Gracias a todos
Paulo Cesar Alvarado (techno)