Información de Versión

- Ultima actualización - 27/03/2013
- Versión - 0.2.1
- Contacto - tuxtor@shekalug.org

Control de cambios

VERSIÓN 0.2.1

- Actualización al año 2013
- Enfocado en x86_64
- Creando la primera versión en markdown
- Removida la etapa de castellanizacion porque el tutorial murio con la Gentoo wiki antigua

Preliminares

Esta es la actualización de una de las guias que hicieron famoso al El Abismo de Tux, la guia de instalación rapida de Gentoo que nacio en el año 2008 cuando realizaba mis estudios de pregrado,

Lo que la mayoría de personas se pregunta, es ¿porque elaborar un nuevo manual cuando todo esta dicho en el handbook?. Por una o varias razones esta guiá no pretende ser ni mejor ni ser reemplazo del handbook, de hecho esta basado en el mismo. Esta guiá no esta diseñada para aquellos que ya tengan el valor de empezar con instalaciones avanzadas, va dirigida a aquellas personas a las cuales el handbook les parece largo y tedioso. Ademas que no tienen los suficientes conocimientos técnicos pero de una u otra manera necesitan Gentoo instalado. Como un extra incluyo algunos links utiles que me han servido a lo largo de mi experiencia (poca a decir verdad) con Gentoo. Este manual esta pensado para arquitecturas x86_64 pero las instrucciónes estan escritas de manera general.

Paso 1 - Consiguiendo todo lo necesario

Para instalar Gentoo necesitamos un CD con el cual podamos iniciar un sistema Linux en memoria, tradicionalmente se suele utilizar el CD de instalación minimal oficial del proyecto, aunque tambien es posible instalar Gentoo desde SystemRescueCD o incluso desde CDs Live de otras distribuciones Linux este manual asume que la instalación es hecha con el CD minimal que podemos descargar en los mirrors de gentoo.

Después de arrancar el CD minimal tendremos un sistema operativo básico corriendo sobre memoria RAM, al inicio se nos mostrara un prompt con privilegios de superusuario, similar a este:

```
livecd root#
```

A partir de aca dejaremos el mouse por un lado. Se debe resaltar que una instalación normal se realiza via web, si la red no tiene un servidor DHCP (como el que se encuentra en la mayoría de routers y modems ADSL caseros) el comando net-setup nos presenta un asistente básico para configurar nuestra conexión a internet. Para verificar que nuestra dirección ip sea la correcta, podemos ejecutar el comando ifconfig.

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.0.4  netmask 255.255.255.0  broadcast 192.168.0.255
    inet6 fe80::a00:27ff:fe99:28c8  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:99:28:c8  txqueuelen 1000  (Ethernet)
    RX packets 129  bytes 18919 (18.4 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 61  bytes 9150 (8.9 KiB)
```

```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

    lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 0  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

En la salidad de ifconfig podemos observar que nuestra interfaz de red se denomina enp0s3 y su ip es la 192.168.0.4, estos datos pueden ser utiles más adelante

Paso 2 - Preparando nuestro disco duro

Es momento de preparar nuestro disco duro, para esto la herramienta predilecta es fdisk, sin embargo cuando no tenemos conocimientos de las características de lo que estamos haciendo podemos utilizar cfdisk que seria la versión amigable.

```
livecd ~ # cfdisk /dev/sda
```

El parametro /dev/sda es opcional si solo contamos con un unico disco duro y dependiendo si este disco es IDE, nuestro disco duro tambien puede estar representado por /dev/hda. El esquema de particiones es una elección personal y recomiendo uno similar a este:

- /dev/sda1 - /boot - 100MB - ext2
- /dev/sda2 - /swap - 125% de la memoria RAM si tenemos más de 2GB, el doble de la memoria ram si tenemos menos - swap
- /dev/sda3 - / - Minimo 3000 MB, idealmente 2000 MB - ext4
- /dev/sda4 - /home - Resto de espacio libre - ext4

Aplicado en un disco duro, tendra mas o menos la apariencia mostrada en la figura 1

Si utilizamos cfdisk basta con que seleccionemos *Free space* cada vez que necesitemos crear una nueva partición, luego seleccionemos con el teclado la opción [New] y cfdisk nos preguntara el tamaño y algunas opciones especiales, entre las cuales no debe faltar Bootable en la partición /boot. Luego de que todo este listo seleccionamos [Write].

Dependiendo si escogemos o no el esquema solo nos resta dar formato a cada una de las particiones con mkfs para ext2/ext4 y mkreiserfs si vamos a usar reiser. Además de esto activamos nuestra particion swap.

```
# mkfs.ext2 /dev/sda1
# mkswap /dev/sda2
# mkfs.ext4 /dev/sda3
# mkfs.ext4 /dev/sda4
# swapon /dev/sda2
```

Paso 3 - Configuramos el sistema

Hechas nuestras particiones nos queda montarlas para poder acceder a ellas, regularmente el punto de montaje sera /mnt/gentoo, pero puede ser cualquier otro directorio dentro de /mnt siempre y cuando exista. Tenemos que montar todo en el siguiente orden, /, /boot, /home por consiguiente debemos crear los directorios boot y home **luego** de crear /

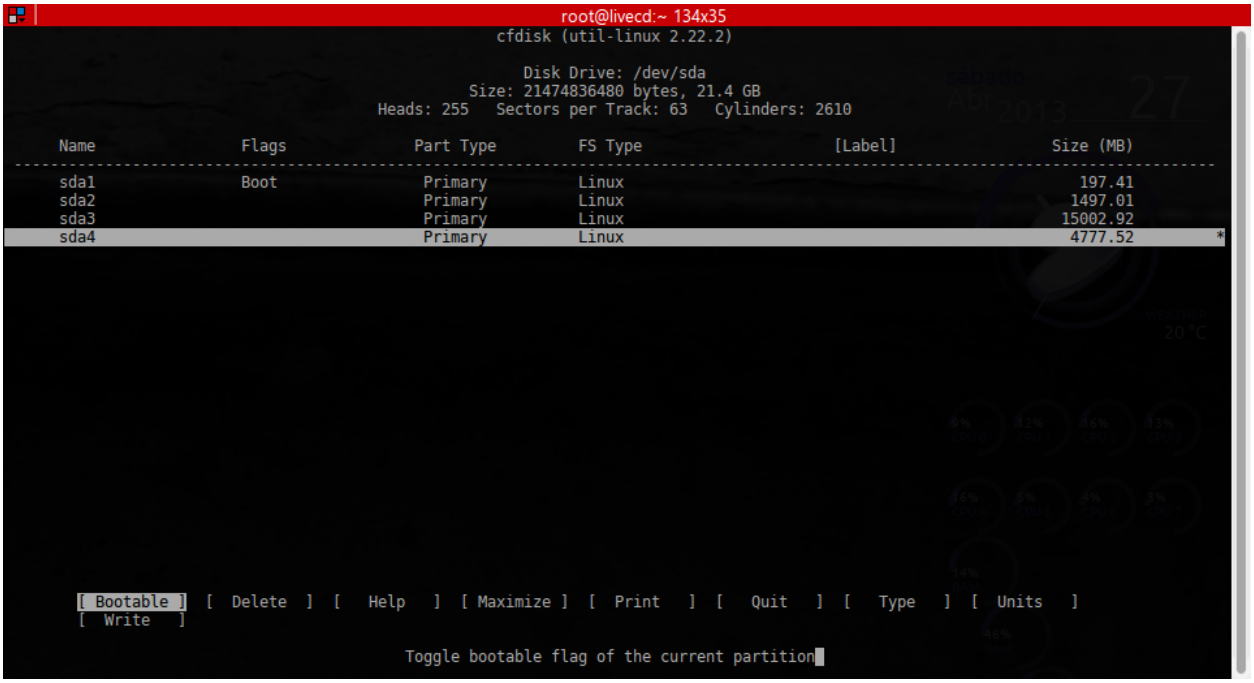

Figure 1: cfdisk

```
# mount /dev/sda3 /mnt/gentoo
# mkdir /mnt/gentoo/boot
# mkdir /mnt/gentoo/home
# mount /dev/sda4 /mnt/gentoo/home
# mount /dev/sda1 /mnt/gentoo/boot
```

En este punto es conveniente configurar la fecha y hora de nuestro sistema mediante el comando date, el formato es el siguiente date MMDDhhmmYYYY.

```
# date 080800002013
```

Paso 4 - Descargando el Stage

Stage 1,2 o 3, sera nuestro sistema base desde el cual se ira construyendo nuestro sistema a antojo. La regla es a mayor numero de stage menor optimización, aunque el único oficialmente soportado por Gentoo hasta la fecha es el stage 3.

Recordando que nuestros discos duros están ya montados sobre /mnt/gentoo procedemos a descargar y descomprimir el stage, estos stages de igual forma se encuentran en los mirrors de gentoo, recordando que los stages se encuentran en releases/amd64/current-stage3 independientemente del mirror que seleccionemos.

Suponiendo que queramos un mirror de Sudamerica, utilizamos wget para descargarlo, no se debe confundir con ia64 que es una arquitectura diferente creada por Intel para servidores, **notese que wget en este caso es solo un ejemplo ya que actualmente los Stages se generan automaticamente y se actualizan periodicamente**.

```
# cd /mnt/gentoo
# wget http://gentoo.c3sl.ufpr.br/releases/amd64/autobuilds/current-iso/20130425/stage3-amd64-20130
```

Luego de que hemos descargado el stage procedemos a descomprimirlo con el comando tar, las opciones significan

- x – Extract

- v – Verbose
- j – Sistema de archivos bz2
- p – Preservar sistema de archivos y symlinks
- f – Utilizar el sistema de archivos predeterminado

```
# tar xvjpf stage3-amd64-20130425.tar.bz2
```

Paso 5 - Configurando las opciones de compilación

Ahora que hemos descomprimido el stage, es momento de editar las opciónes de optimización, para esto utilizaremos el editor de textos nano que viene instalado en el CD Minimal

```
# nano -w /mnt/gentoo/etc/portage/make.conf
```

En este momento lo más importante es editar los parametro CFLAGS y CXXFLAGS, se puede ver un listado completo en el manual on-line de GCC, sin embargo un camino rapido es verificar el apartado safe CFLAGS en la antigua Gentoo Wiki y tambien en la nueva y oficial Gentoo Wiki donde muchos usuarios han contribuido con las configuraciones de CFLAGS que funcionan sin introducir inestabilidad en el sistema. Para saber que tipo de procesador tenemos podemos utilizar el archivo cpuinfo, en este caso es un intel core i7-2670QM (saldra la misma información para todos los cores disponibles).

```
# cat /proc/cpuinfo
processor   : 0
vendor_id   : GenuineIntel
cpu family  : 6
model       : 42
model name  : Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz
stepping    : 7
cpu MHz     : 2211.612
cache size  : 6144 KB
fpu     : yes
fpu_exception   : yes
cpuid level : 5
wp      : yes
flags       : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx f
bogomips    : 4423.22
clflush size    : 64
cache_alignment : 64
address sizes   : 36 bits physical, 48 bits virtual
power management:
```

Para este procesador i7 nuestro archivo make.conf deberia quedar asi:

CFLAGS=“-march=corei7 -O2 -pipe”

CXXFLAGS=“${CFLAGS}”

Además de las CFLAGS debemos agregar el parametro makeopts para compilar utilizando los diferentes cores/cpus fisicos que tengamos en nuestro sistema, esto lo hacemos cambiando el parametro MAKEOPTS, por regla general el valor de MAKEOPTS debe ser numero de cores fisicos+1, por ejemplo para un procesador con 4 cores, el valor seria:

MAKEOPTS=“-j5”

Presionamos control+O para guardar y control+X para salir, ahora todo ha quedado configurado.

Paso 6 - Seleccionando nuestros orígenes de descarga

Gentoo descargara el código fuente desde internet, para lo cual disponibiliza algunos mirrors diseñados para esto, aunque este paso es opcional es bueno que escojamos desde donde Gentoo debe descargar los paquetes, principalmente si nuestro pais tiene mirros disponibles. Mediante mirror select podemos elegir el(los) mirror(s) desde donde Gentoo descargara, y mediante una tubería >> vaciarlo en nuestro archivo make.conf del paso anterior.

```
# mirrorselect -i -o >> /mnt/gentoo/etc/portage/make.conf
```

Luego de seleccionar los mirrors con barra espaciadora y guardar los cambios en OK, tambien podemos elegir mirrors para la actualización del administrador de paquetes.

```
# mirrorselect -i -r -o >> /mnt/gentoo/etc/portage/make.conf
```

Hasta este punto, hemos llegado estamos a un paso de entrar en nuestro nuevo sistema Gentoo, como consejo adicional deberíamos de copiar la información de nuestro servidor DNS para proseguir con la instalación

```
# cp -L /etc/resolv.conf /mnt/gentoo/etc/resolv.conf
```

Paso 7 - Llego la hora del chrooting

La instrucción chroot nos permitirá ejecuta comandos o shells interactivos con un directorio raíz especial. En este caso nuestro nuevo sistema que actualmente esta en /mnt/gentoo, ademas de esto debemos de montar el sistema de archivos proc (que representa algunos procesos del sistema y su información) y dev (que representa los dispositivos de nuestro sistema), esto ayudara posteriormente cuando configuremos el kernel linux a utilizar.

```
# mount -t proc none /mnt/gentoo/proc
# mount --rbind /sys /mnt/gentoo/sys
# mount --rbind /dev /mnt/gentoo/dev
```

Y hacemos un chroot utilizando bash como interprete de comandos en el nuevo entorno

```
# chroot /mnt/gentoo /bin/bash
```

Y dentro del chroot, actualizamos nuestras variables de entorno

```
# source /etc/profile
```

Podemos poner la leyenda (chroot) en nuestro interprete de comandos para facilidad, aunque puede ser cualquier otra leyenda. # export PS1="(chroot) $PS1"

Paso 8 - Sincronizando portage

Dentro de nuestro nuevo sistema necesitamos portage, el corazón y alma de Gentoo. Son una serie de scripts llamados ebuilds que contienen las dependencias, opciones e instrucciones de compilación de los paquetes oficialmente soportados por Gentoo, para esto creamos el directorio /usr/portage y lo actualizamos mediante emerge-webrsync.

```
# mkdir /usr/portage
# emerge-webrsync
```

Paso 8 - Escogiendo un perfil adecuado

Dado que Gentoo es un sistema operativo multi proposito, disponibiliza varios conjuntos de configuraciones predeterminadas conocidas como perfiles, podemos ver los perfiles disponibles utilizando eselect.

```
# eselect profile list
[1]   default/linux/amd64/13.0 *
[2]   default/linux/amd64/13.0/selinux
[3]   default/linux/amd64/13.0/desktop
[4]   default/linux/amd64/13.0/desktop/gnome
[5]   default/linux/amd64/13.0/desktop/kde
[6]   default/linux/amd64/13.0/developer
[7]   default/linux/amd64/13.0/no-multilib
[8]   default/linux/amd64/13.0/x32
[9]   hardened/linux/amd64
[10]  hardened/linux/amd64/selinux
[11]  hardened/linux/amd64/no-multilib
[12]  hardened/linux/amd64/no-multilib/selinux
[13]  hardened/linux/uclibc/amd64
```

En este listado aparecen dos tipos de perfiles normales y no-multilib, la diferencia es que los perfiles no-multilib no dan soporte a ejecutar aplicaciones de 32 bits en entornos de 64 bits por defecto. Asi pues podemos seleccionar un perfil utilizando eselect, en este ejemplo seleccionamos un perfil de escritorio normal para Gnome.

```
# eselect profile set 4
```

Ademas de esto hay que configurar nuestra zona horaria, las disponibles están dentro de /usr/share/zoneinfo, un ls dentro de esta carpeta puede aclararnos el panorama, las que apliquen a nuestro sistema van dentro de /etc/localtime, asi que procedemos a copiarlas:

```
# cp /usr/share/zoneinfo/America/Sao_Paulo /etc/localtime
```

Y además de esto lo establecemos en nuestro archivo /etc/timezone

```
# echo "America/Sao_Paulo" > /etc/timezone
```

Paso 9 - La variable USE

Una de las verdaderas ventajas de Gentoo además de la optimización en la compilación, es el control granular de las caracteristicas de nuestro sistema, esto mediante las USE flags. Las USE flags indicaran que caracteristicas seran incluidas en los paquetes que utilicemos y para definirlas globalmente, se utiliza la variable USE dentro del archivo make.conf, asi que nuevamente lo editamos (esta vez el comando cambia por el chroot).

```
# nano /etc/portage/make.conf
```

Definir las USE flags es un tema complicado y dependera de que necesitemos para cada paquete, como regla general para cada paquete nuevo que instalemos deberiamos revisar la Gentoo wiki si existe alguna recomendación, de momento dejaremos una configuración basica.

```
USE="gtk gnome -qt4 -kde dvd alsa cdr bindist mmx sse sse2"
```

Grabamos el archivo y estamos listos para continuar

Paso 10 – El kernel Linux

Es tal ves el paso más fundamental y a la ves difícil, Gentoo en este caso nos proporciona dos caminos, la vía tradicional y mediante la herramienta genkernel que genera un kernel similar al del live cd que evaluá nuestros dispositivos y decide que módulos cargar o no, y una configuración manual donde nosotros seremos responsables de compilar manualmente el kernel, para lo cual hay más detalles en la guia Gentoo del Kernel. Para este tutorial utilizaremos la via del genkernel por ser principiantes.

Primero instalamos el codigo fuente mediante portage, este codigo fuente se instala con el paquete gentoo-sources (más información de portage en la guia Gentoo de Portage)

```
# emerge gentoo-sources
```

Y procedemos a configurarlo mediante genkernel

```
# emerge genkernel
# genkernel all
```

Este paso llevara un tiempo, asi que mientras tanto pueden leer mis ultimos post en El Abismo de Tux :D

Paso 12 – Configurando fstab

Fstab, es un fichero para definir puntos de montaje (recordemos que en UNIX todo cuelga de una única raíz /) que serán cargados al arrancar el sistema. Cada linea define una partición y lleva mas o menos esta estructura.

- Partición
- Punto de montaje
- Tipo de sistema de archivos
- Opciones especiales
- Determina si se debe volcar o no la partición (si no sabe que es esto solo dejen 0)
- Orden en que fsck debe comprobar en caso de que el sistema no se apague correctamente (si tampoco sabes que es deja todos 0 y boot 1)

```
# nano -w /etc/fstab
```

Retomando nuestro esquema de particiones inicial debería de verse como el ejemplo mostrado en la figura 2

Por ultimo guardamos el archivo

Paso 10 – Mas configuraciones

RED

Establecemos el nombre que tendrá nuestro equipo en la red:

```
# nano -w /etc/conf.d/hostname
```

Reemplazar “localhost” por cualquier otro nombre.

Si nuestra pc fuera parte de un dominio

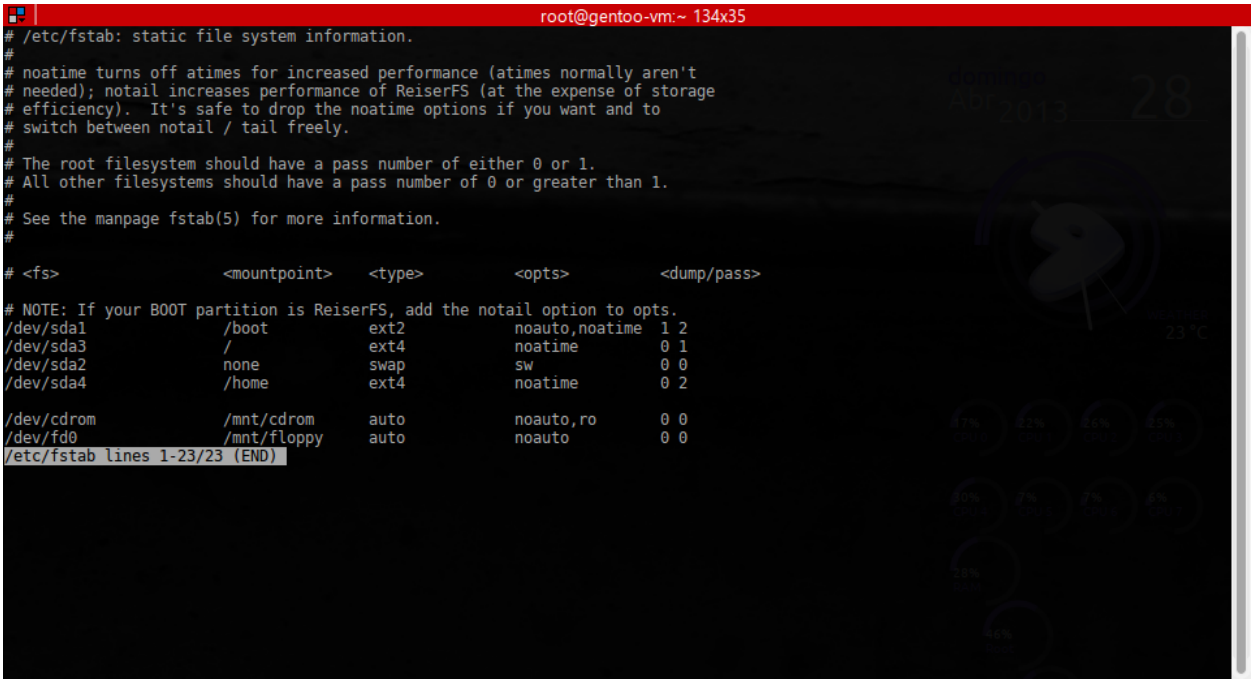

Figure 2: Ejemplo fstab

```
# nano -w /etc/conf.d/net
```

Y etablecemos el dominio: dns_domain_lo=“dominio”.

Habilitar el demonio dhcp para determinadas interfaces

```
# nano -w /etc/conf.d/net
```

Con estas linea bastara, notese que el nombre enp0s3 fue el nombre que nos dio ifconfig al inicio (conmunmente es eth0)

config_enp0s3=(“dhcp”)

Luego tomamos como base net.lo y hacemos un link simbolico para la nueva interfaz, ya configurada

```
# cd /etc/init.d/
# ln -s net.lo net.enp0s3
```

Y lo agregamos al arranque del sistema con rc-update

```
# rc-update add net.enp0s3 default
```

Modificamos el archivo /etc/rc.conf que son las configuraciones globales iniciales, el archivo esta bastante bien comentado queda a nuestro criterio que cambios hacerle.

```
# nano /etc/rc.conf
```

SUPER USUARIO

Debemos de cambiar la clave de el usuario root (superusuario) ya que de reiniciar sin este cambio no podremos administrar el sistema con privilegios.

```
# passwd
```

KEYMAPS

Establecemos nuestra variante de teclado

```
# nano -w /etc/conf.d/keymaps
```

Agregamos las siguientes líneas si nuestro teclado es en español: > KEYMAP="es" > SET_WINDOWKEYS="yes"

CONFIGURACIÓN DEL RELOJ

El reloj se configura de la siguiente manera:

```
# nano -w /etc/conf.d/hwclock
```

Al igual que el anterior esta bastante bien comentado, algunas lineas importantes son:

clock="UTC" clock_systohc="YES"

Si el sistema fuera dual-boot con windows se debe cambiar de UTC a local, de lo contrario ambos relojes estaran siempre mal configurados, si queremos que Gentoo actualice el reloj del sistema al apagarlo dejamos clock_systohc.

CONFIGURACIÓN DE LOCALIZACIONES

En esta parte le decimos a Gentoo que sistemas soportara en el archivo /etc/locale.gen.

```
# nano -w /etc/locale.gen
```

Por ejemplo para dar soporte a ingles, español de Guatemala y portugues de Brasil, el contenido deberia ser

en_US ISO-8859-1

en_US.UTF-8 UTF-8

pt_BR.UTF-8 UTF-8

pt_BR ISO-8859-1

es_ES.UTF-8 UTF-8

es_ES ISO-8859-1

es_ES@euro ISO-8859-15

es_GT.UTF-8 UTF-8

es_GT ISO-8859-1

Actualizamos todo mediante locale-gen

```
# locale-gen
```

Por ultimo establecemos una localización por defecto en el archivo /etc/env.d/02locale

LANG="es_ES.UTF-8"

LANGUAGE="es_ES.UTF-8"

LC_COLLATE="C"

Y refrescamos variables de entorno y el entorno como tal

```
# env-update && source /etc/profile
```

Si tienen duda de este proceso pueden consultar la guia de localización de Gentoo

Paso 11 – Instalando herramientas del sistema

REGISTRO DE SISTEMA (LOG)

En mi caso utilizo syslog

```
# emerge syslog-ng
# rc-update add syslog-ng default
```

SERVICIO CRON

Para ejecutar tareas programadas en determinado tiempo (en mi caso vixie-cron)

```
# emerge vixie-cron
# rc-update add vixie-cron default
```

Indexado de ficheros (para búsquedas rápidas con herramienta locate)

```
# emerge mlocate
```

Cliente dhcp (para obtener los datos de la red de manera automática

```
# emerge net-misc/dhcpcd
```

Paso 12 – Configurando GRUB

En este punto tenemos dos opciones usar grub2 o grub original, si tienen un sistema UEFI deben consultar la wiki de Gentoo para Grub2 (en ingles) si no, grub original funciona bien, además de esto tambien pueden utilizar lilo, que queda fuera de este tutorial.

Ok, manos a la obra e instalamos grub

```
# emerge grub
```

Genkernel ejecuto un paso importante que fue copiar las imagenes de nuestro kernel y el archivo initramfs hacia /boot, para lo cual debemos verificar cual es el nombre exacto de los archivos que nos servira al configurar el grub.

```
# ls /boot/kernel* /boot/initramfs*
/boot/initramfs-genkernel-x86_64-3.7.10-gentoo-r1  /boot/kernel-genkernel-x86_64-3.7.10-gentoo-r1
```

La creación del menu de arranque se hace mediante el archivo grub.conf entonces procedemos a editarlo, este archivo es muy personalizable, asi que solo definiremos un ejemplo de lo necesario para que nuestro sistema arranque

```
# nano -w /boot/grub/grub.conf
```

Que en nuestro caso tendria el siguiente contenido, notese que los parametros kernel e initrd tienen los valores que vimos anteriormente

```
default 0
timeout 30
title Mi primer gentoo
root (hd0,0)
kernel /boot/kernel-genkernel-x86_64-3.7.10-gentoo-r1 real_root=/dev/sda3
initrd /boot/initramfs-genkernel-x86_64-3.7.10-gentoo-r1
```

Por ultimo pero no menos importante instalamos el gestor en el disco duro-

```
# grep -v rootfs /proc/mounts > /etc/mtab
# grub-install --no-floppy /dev/sda
```

Ya casi, por el momento reiniciamos el sistema, y si todo va bien, nos vemos en un par de minutos # exit # cd # umount -l /mnt/gentoo/dev{/shm,/pts,} # umount -l /mnt/gentoo{/boot,/proc,} # reboot

Ya de vuelta nos encontramos con gentoo funcionando.

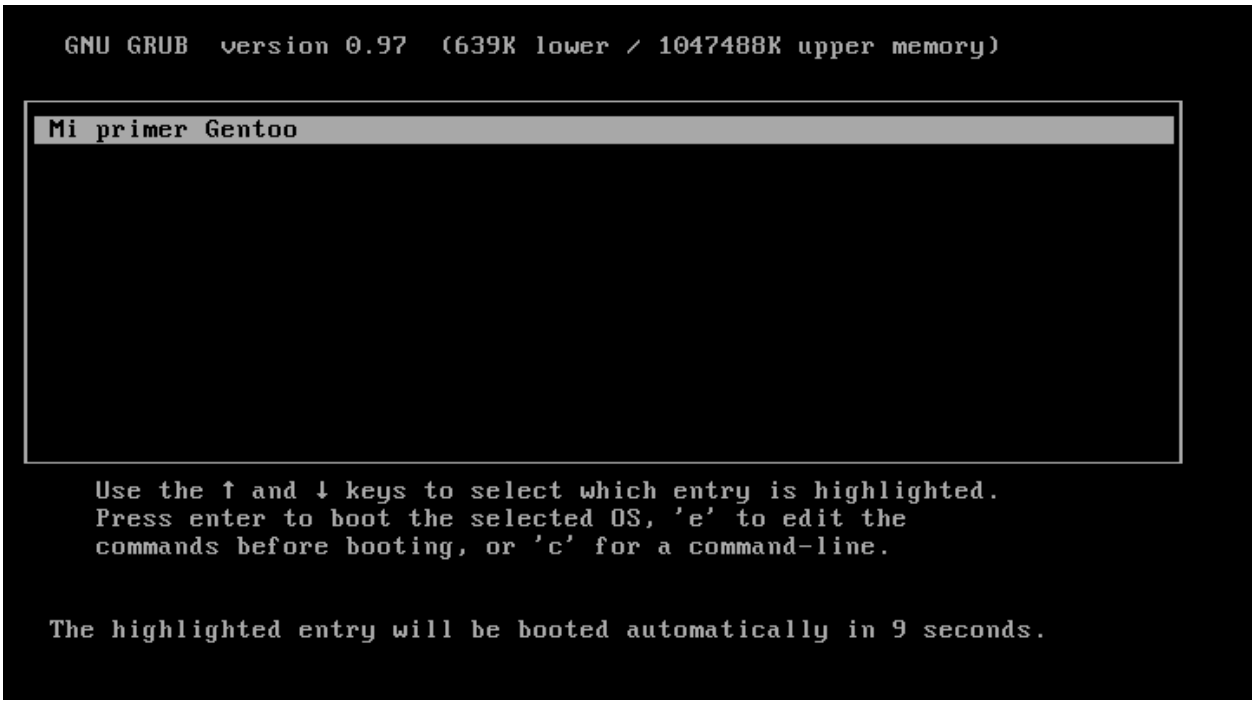

Figure 3: Grub

Iniciamos sesión como root y faltaría un ultimo paso, crear nuestro usuario de día a día, root solo es necesario para administrar el sistema ya que el tiene el poder de editar cualquier archivo es peligroso utilizarlo más allá de administración, por defecto lo agregamos a los grupos de usuario que recomienda el handbook de gentoo y establecemos que el interprete de comandos a utilizar sera bash. Noten que aca creamos un usuario **tuxtor** que deben ajustar a sus necesidades, notese que si las locales estan bien configuradas, nuestro sistema estara en español.

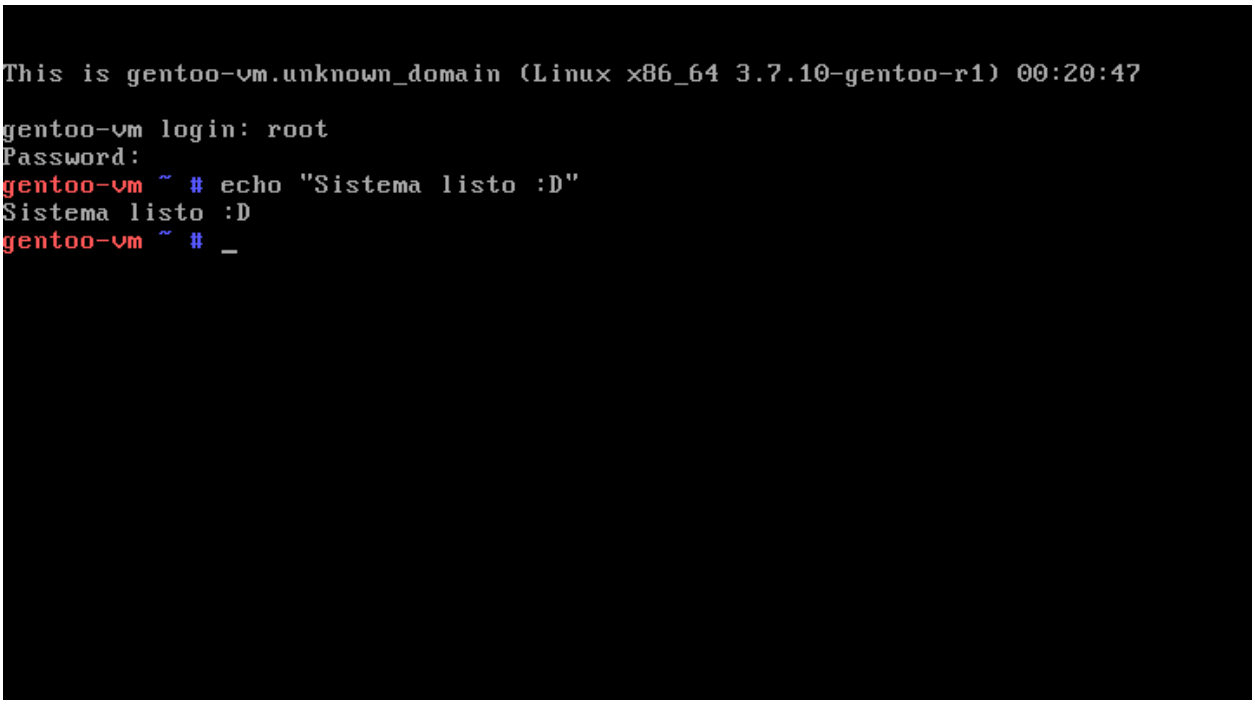

Figure 4: Gentoo Listo

```
# useradd -m -G users,wheel,audio,cdrom,usb,video -s /bin/bash tuxtor
# passwd tuxtor
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
```

Por ultimo limpiamos espacio en disco borrando el stage que descargamos.

```
# rm /stage3-*.tar.bz2*
```

Y si ya llegaste hasta aqui Felicidades!!! ya tenemos gentoo instalado.

Desde aquí sera cuestión de nosotros instalar paquetes y personalizar el sistema, algunos links utiles:

- Portage(en ingles): http://www.gentoo.org/doc/en/handbook/handbook-x86.xml?part=2&chap=1
- ALSA (sonido): http://www.gentoo.org/doc/es/alsa-guide.xml
- Gnome (entorno de escritorio): http://www.gentoo.org/doc/es/gnome-config.xml
- Kde(en ingles): http://www.gentoo.org/proj/en/desktop/kde/

Despues de esto un Gentoo bien configurado puede tener una apariencia similar a la figura 5

Si este manual les ha sido de ayuda no duden en contactarme ya sea mediante mi email o por IRC en el canal #gentoo-es

Figure 5: Gentoo Configurado