



tira de linuxhispano.net



by danigm



gentoo



# “Introducción a la metadistribución Gentoo Linux”

por Héctor Fernández  
trorrr@gpul.org

# ¿Qué es gentoo linux?

Es una “distribución” Linux/BSD...



# ¿Qué es gentoo linux?

Es una “distribución” Linux/BSD...  
...sin un sistema real de paquetes

# ¿Qué es gentoo linux?

Es una “distribución” Linux/BSD...  
...sin un sistema real de paquetes

Se basa en un sistema de ports del tipo \*BSD

# ¿Qué es gentoo linux?

Es una “distribución” Linux/BSD...  
...sin un sistema real de paquetes

Se basa en un sistema de ports del tipo \*BSD

El código fuente se compila según:  
- arquitectura de la máquina  
*(11, las soportadas por gcc)*

# ¿Qué es gentoo linux?

Es una “distribución” Linux/BSD...  
...sin un sistema real de paquetes

Se basa en un sistema de ports del tipo \*BSD

El código fuente se compila según:

- arquitectura de la máquina  
*(11, las soportadas por gcc)*

Possible en otras distribuciones

# ¿Qué es gentoo linux?

Es una “distribución” Linux/BSD...  
...sin un sistema real de paquetes

Se basa en un sistema de ports del tipo \*BSD

El código fuente se compila según:

- arquitectura de la máquina  
*(11, las soportadas por gcc)*

Possible en otras distribuciones

- necesidades del usuario

# ¿Qué es gentoo linux?

Es una “distribución” Linux/BSD...  
...sin un sistema real de paquetes

Se basa en un sistema de ports del tipo \*BSD

El código fuente se compila según:

- arquitectura de la máquina  
*(11, las soportadas por gcc)*

Possible en otras distribuciones

- necesidades del usuario

No posible en otras distribuciones



# El sentido de gentoo

¿Qué sentido tiene si se puede compilar normal y pasarle las opciones al configure?

# El sentido de gentoo

¿Qué sentido tiene si se puede compilar normal y pasarle las opciones al configure?

- “algo” tiene que generar las dependencias

# El sentido de gentoo

¿Qué sentido tiene si se puede compilar normal y pasarle las opciones al configure?

- “algo” tiene que generar las dependencias
- palabras clave unificadas para las FLAGS

# Un poco de historia

Ligada a los avances de su fundador, *Daniel Robbins* (Nuevo México)

# Un poco de historia

Ligada a los avances de su fundador, *Daniel Robbins* (Nuevo México)

- su primera linux es una Debian en un pentium 166

# Un poco de historia

Ligada a los avances de su fundador, *Daniel Robbins* (Nuevo México)

- su primera linux es una Debian en un pentium 166
- se pasó a Stampede

# Un poco de historia

Ligada a los avances de su fundador, *Daniel Robbins* (Nuevo México)

- su primera linux es una Debian en un pentium 166
- se pasó a Stampede
- toma conciencia de que linux no es simplemente código, sino una comunidad de gente

# Un poco de historia

- se convierte en desarrollador de Stampede creando un formato de paquetes

# Un poco de historia

- se convierte en desarrollador de Stampede creando un formato de paquetes
- voces discordantes

# Un poco de historia

- se convierte en desarrollador de Stampede creando un formato de paquetes
- voces discordantes
- deja Stampede...

# Un poco de historia

- se convierte en desarrollador de Stampede creando un formato de paquetes
- voces discordantes
- deja Stampede...
- ...y comienza su propia distribución

# Un poco de historia

- se convierte en desarrollador de Stampede creando un formato de paquetes
- voces discordantes
- deja Stampede...
- ...y comienza su propia distribución
  - control

# Un poco de historia

- se convierte en desarrollador de Stampede creando un formato de paquetes
- voces discordantes
- deja Stampede...
- ...y comienza su propia distribución
  - control
  - rendimiento (CPU)

# Un poco de historia

- se convierte en desarrollador de Stampede creando un formato de paquetes
- voces discordantes
- deja Stampede...
- ...y comienza su propia distribución
  - control
  - rendimiento (CPU)
  - “demonios” (RAM)

# Un poco de historia

PROBLEMA

1 persona

# Un poco de historia

## PROBLEMA

1 persona => tiempo limitado

# Un poco de historia

## PROBLEMA

1 persona => tiempo limitado => *scripts*

# Un poco de historia

## PROBLEMA

1 persona => tiempo limitado => *scripts*

## Sistema ebuild

(scripts en bash para crear e instalar los binarios a partir de código fuente y parches)

# Un poco de historia

## PROBLEMA

1 persona => tiempo limitado => *scripts*

## Sistema ebuild

(scripts en bash para crear e instalar los binarios a partir de código fuente y parches)

Los primeros: gcc, glibc, binutils...

# Un poco de historia

## PROBLEMA

1 persona => tiempo limitado => *scripts*

## Sistema ebuild

(scripts en bash para crear e instalar los binarios a partir de código fuente y parches)

Los primeros: gcc, glibc, binutils...

Unos meses más tarde alcanzó la decena de desarrolladores y se bautizó como *Enoch*

# Un poco de historia

- Colabora en la solución de un bug en glibc ya solucionado internamente por RedHat

# Un poco de historia

- Colabora en la solución de un bug en glibc ya solucionado internamente por RedHat
- Enoch se mejora, se interesan más desarrolladores y el proyecto se renombra a *Gentoo Linux*

# Un poco de historia

- Colabora en la solución de un bug en glibc ya solucionado internamente por RedHat
- Enoch se mejora, se interesan más desarrolladores y el proyecto se renombra a *Gentoo Linux*
- Compra un Dual Celeron 366, que provoca un fallo sin solución, haciendo que la máquina se cuelgue. Esto le impedirá seguir con el desarrollo durante un tiempo.

# Un poco de historia

- Se pasa a FreeBSD, y se da cuenta de que se ajusta bastante a la idea que tenía para “su distribución”

# Un poco de historia

- Se pasa a FreeBSD, y se da cuenta de que se ajusta bastante a la idea que tenía para “su distribución”
- Unos meses más tarde vuelve a retomar el proyecto con una máquina nueva aplicando lo aprendido de FreeBSD, principalmente el sistema de Ports, al que llamará Portage y estará escrito en Python

# Un poco de historia

- Se pasa a FreeBSD, y se da cuenta de que se ajusta bastante a la idea que tenía para “su distribución”
- Unos meses más tarde vuelve a retomar el proyecto con una máquina nueva aplicando lo aprendido de FreeBSD, principalmente el sistema de Ports, al que llamará Portage y estará escrito en Python
- Versión 1.0 de Gentoo liberada en Marzo de 2002

# ¿Distribución?

Scripts

# ¿Distribución?

Scripts + código fuente

# ¿Distribución?

Scripts + código fuente + parches

# ¿Distribución?

Scripts + código fuente + parches

NO es una distribución, es “algo” que sirve para crear distribuciones

# ¿Distribución?

Scripts + código fuente + parches

NO es una distribución, es “algo” que sirve para crear distribuciones

***GENTOO LINUX ES UNA METADISTRIBUCIÓN***



# ¿Por qué ese nombre?



Gentoo (*Pygoscelis papua*) es el pingüino más rápido bajo el agua, alcanzando velocidades de 36 km/h

gentoo

# Un poco más de historia

- En 1999 se crea la Gentoo Foundation
  - temas financieros (sponsors, donaciones...)

# Un poco más de historia

- En 1999 se crea la Gentoo Foundation
  - temas financieros (sponsors, donaciones...)
  - protección jurídica (protección de las licencias sobre el código fuente, marcas registradas, artwork...)

# Un poco más de historia

- En 1999 se crea la Gentoo Foundation
  - temas financieros (sponsors, donaciones...)
  - protección jurídica (protección de las licencias sobre el código fuente, marcas registradas, artwork...)
  - votaciones

# Un poco más de historia

- En 1999 se crea la Gentoo Foundation
  - temas financieros (sponsors, donaciones...)
  - protección jurídica (protección de las licencias sobre el código fuente, marcas registradas, artwork...)
  - votaciones
  - mantenimiento del Contrato Social

# El Contrato Social

- **Gentoo** deja elegir entre varias opciones

# El Contrato Social

- **Gentoo** deja elegir entre varias opciones
- **Gentoo** libera todo lo desarrollado

# El Contrato Social

- **Gentoo** deja elegir entre varias opciones
- **Gentoo** libera todo lo desarrollado
- **Gentoo** sirve a la comunidad y está hecho por la comunidad

# El Contrato Social

- **Gentoo** deja elegir entre varias opciones
- **Gentoo** libera todo lo desarrollado
- **Gentoo** sirve a la comunidad y está hecho por la comunidad
- **Gentoo** es independiente de organizaciones y empresas

# Actualmente

- Daniel Robbins fue contratado en Abril de 2004 por Microsoft para "ayudar a comprender los proyectos de código abierto y colectivos (community)"

# Actualmente

- Daniel Robbins fue contratado en Abril de 2004 por Microsoft para "ayudar a comprender los proyectos de código abierto y colectivos (community)"
- En Enero de 2006 abandona el cargo

# Actualmente

- Daniel Robbins fue contratado en Abril de 2004 por Microsoft para "ayudar a comprender los proyectos de código abierto y colectivos (community)"
- En Enero de 2006 abandona el cargo
- Trabaja como CTO en una empresa de Nuevo Mexico

# Instalación

- Ha cambiado mucho en la última versión

# Instalación

- Ha cambiado mucho en la última versión
- Tradicionalmente no hay instalación, existe una livecd con todo lo necesario para “instalar manualmente”

# Instalación

- Ha cambiado mucho en la última versión
- Tradicionalmente no hay instalación, existe una livecd con todo lo necesario para “instalar manualmente”
- El proceso es muy parecido a crear una nueva livecd (salvo que debemos arrancarla desde disco duro, no volcarla a un soporte extraíble)

# Instalación

## 1. Particionado

# Instalación

1. Particionado
2. Copiar o compilar el sistema base  
(sin núcleo)

# Instalación

1. Particionado
2. Copiar o compilar el sistema base (sin núcleo)
3. Generar una copia de Portage

# Instalación

1. Particionado
2. Copiar o compilar el sistema base (sin núcleo)
3. Generar una copia de Portage
4. Entrar en el nuevo sistema con chroot

# Instalación

1. Particionado
2. Copiar o compilar el sistema base (sin núcleo)
3. Generar una copia de Portage
4. Entrar en el nuevo sistema con chroot
5. Compilar el núcleo a medida (forma rápida con genkernel)

# Instalación

1. Particionado
2. Copiar o compilar el sistema base (sin núcleo)
3. Generar una copia de Portage
4. Entrar en el nuevo sistema con chroot
5. Compilar el núcleo a medida (forma rápida con genkernel)
6. Configuración general del sistema (red, contraseña, syslogger, idioma...)

# Instalación

1. Particionado
2. Copiar o compilar el sistema base (sin núcleo)
3. Generar una copia de Portage
4. Entrar en el nuevo sistema con chroot
5. Compilar el núcleo a medida (forma rápida con genkernel)
6. Configuración general del sistema (red, contraseña, syslogger, idioma...)
7. Configuración del cargador de arranque

# Instalación

- Seguidamente instalaríamos los servicios y las aplicaciones a través de Portage

# Instalación

- Seguidamente instalaríamos los servicios y las aplicaciones a través de Portage
- Desde la versión 2006.0 existe un instalador gráfico (versión GTK y versión Dialog) que genera un script que más tarde realizará todos los pasos de instalación

# Instalación

- Seguidamente instalaríamos los servicios y las aplicaciones a través de Portage
- Desde la versión 2006.0 existe un instalador gráfico (versión GTK y versión Dialog) que genera un script que más tarde realizará todos los pasos de instalación
- (Opinión personal) ¡Así no se aprende!

# Instalación

- Seguidamente instalaríamos los servicios y las aplicaciones a través de Portage
- Desde la versión 2006.0 existe un instalador gráfico (versión GTK y versión Dialog) que genera un script que más tarde realizará todos los pasos de instalación
- (Opinión personal) ¡Así no se aprende!
- Si actualizamos desde Internet siempre estaremos a la última

# CONTRAS

- Complicado y lento de instalar

# CONTRAS

- Complicado y lento de instalar
- Tiempo de instalación de ebuilds

# CONTRAS

- Complicado y lento de instalar
- Tiempo de instalación de ebuilds
- Consumo de CPU al instalar ebuilds

# CONTRAS

- Complicado y lento de instalar
- Tiempo de instalación de ebuilds
- Consumo de CPU al instalar ebuilds
- Aplicaciones desfasadas rápidamente

# CONTRAS

- Complicado y lento de instalar
- Tiempo de instalación de ebuilds
- Consumo de CPU al instalar ebuilds
- Aplicaciones desfasadas rápidamente
- Configuraciones manuales

# CONTRAS

- Complicado y lento de instalar
- Tiempo de instalación de ebuilds
- Consumo de CPU al instalar ebuilds
- Aplicaciones desfasadas rápidamente
- Configuraciones manuales
- Cuidado al cambiar de versión de gcc

# CONTRAS

- Complicado y lento de instalar
- Tiempo de instalación de ebuilds
- Consumo de CPU al instalar ebuilds
- Aplicaciones desfasadas rápidamente
- Configuraciones manuales
- Cuidado al cambiar de versión de gcc
- Demasiados ebuilds y cada uno con varias versiones

# CONTRAS

- Complicado y lento de instalar
- Tiempo de instalación de ebuilds
- Consumo de CPU al instalar ebuilds
- Aplicaciones desfasadas rápidamente
- Configuraciones manuales
- Cuidado al cambiar de versión de gcc
- Demasiados ebuilds y cada uno con varias versiones
- Por esto último y por Portage estar hecho en Python (lenguaje interpretado), cada vez es más lento

# PROS

- Control absoluto del sistema  
("si algo está ahí, es que lo has puesto tú")

# PROS

- Control absoluto del sistema  
("si algo está ahí, es que lo has puesto tú")
- Gran posibilidad de aprendizaje

# PROS

- Control absoluto del sistema  
 (“si algo está ahí, es que lo has puesto tú”)
- Gran posibilidad de aprendizaje
- Posibilidad de reparar el sistema en base a los conocimientos adquiridos en la instalación

# PROS

- Control absoluto del sistema (“si algo está ahí, es que lo has puesto tú”)
- Gran posibilidad de aprendizaje
- Posibilidad de reparar el sistema en base a los conocimientos adquiridos en la instalación
- Siempre actualizado

# PROS

- Control absoluto del sistema  
(“si algo está ahí, es que lo has puesto tú”)
- Gran posibilidad de aprendizaje
- Posibilidad de reparar el sistema en base a los conocimientos adquiridos en la instalación
- Siempre actualizado
- Menor consumo de CPU y recursos

# PROS

- Control absoluto del sistema (“si algo está ahí, es que lo has puesto tú”)
- Gran posibilidad de aprendizaje
- Posibilidad de reparar el sistema en base a los conocimientos adquiridos en la instalación
- Siempre actualizado
- Menor consumo de CPU y recursos
- Tiempos mejorados en un 10% de media

# PROS

- Uso del juego de instrucciones propio para el procesador en el que se instale

# PROS

- Uso del juego de instrucciones propio para el procesador en el que se instale
- Existen paquetes ya compilados para máquinas poco potentes o para ahorro de tiempo (Openoffice, Mozilla, MPlayer...)

# PROS

- Uso del juego de instrucciones propio para el procesador en el que se instale
- Existen paquetes ya compilados para máquinas poco potentes o para ahorro de tiempo (Openoffice, Mozilla, MPlayer...)
- Elección entre diferentes versiones de un “paquete” a instalar

# PROS

- Uso del juego de instrucciones propio para el procesador en el que se instale
- Existen paquetes ya compilados para máquinas poco potentes o para ahorro de tiempo (Openoffice, Mozilla, MPlayer...)
- Elección entre diferentes versiones de un “paquete” a instalar
- Permisividad de “paquetes” en pruebas (Masked) o inestables (Hard Masked) configurable individualmente

# Portage

- Es nuestro sistema de “paquetes”

# Portage

- Es nuestro sistema de “paquetes”
- Escrito en Python

# Portage

- Es nuestro sistema de “paquetes”
- Escrito en Python
- Más de 10000 ebuilds en bash (probablemente muchos más)

# Portage

- Es nuestro sistema de “paquetes”
- Escrito en Python
- Más de 10000 ebuilds en bash (probablemente muchos más)
- Versiones en MacOSX, Solaris y FreeBSD

# Portage

- Es nuestro sistema de “paquetes”
- Escrito en Python
- Más de 10000 ebuilds en bash (probablemente muchos más)
- Versiones en MacOSX, Solaris y FreeBSD



gentoo

# Portage

- Es nuestro sistema de “paquetes”
- Escrito en Python
- Más de 10000 ebuilds en bash (probablemente muchos más)
- Versiones en MacOSX, Solaris y FreeBSD
- Se utiliza a través de la herramienta **emerge**



gentoo

# Portage

emerge --help

emerge --sync

emerge --search audacity

emerge --searchdesc audio\ editor

emerge --fetchonly audacity

emerge --ask --verbose --tree audacity

emerge --unmerge audacity

emerge --update --ask world

emerge --update --deep --newuse world

emerge --usepkg --getbinpkg audacity

# Portage

Fichero `/etc/make.conf`  
Herramienta `etc-update`

# Ruegos y preguntas

# GRACIAS POR VENIR

Héctor Fernández López  
trorrr@gpul.org





tira de linuxhispano.net

by danigm



gentoo