

MINIMANUAL EMERGE

Emerge es la herramienta de instalación en Gentoo Linux. Emerge se encarga de leer el “índice” de programas disponibles desde el directorio */usr/portage* y de instalar el programa si éste está incluido en ese “árbol” de aplicaciones, instalando también las dependencias que dicho programa necesite de forma automática. Su uso es bastante sencillo:

Instalación de paquetes:

La forma de instalar un paquete es:

```
emerge packagename
```

Si el programa tiene dependencias emerge las instalará.

Para instalar un paquete y actualizar los paquetes que dependan de él se utiliza la flag -u

```
emerge -u package
```

Para actualizar un paquete, los paquetes que de él dependen y los que dependen de las dependencias Ufff !!!! se usa la flag -deep, -D abreviado

```
emerge -uD package
```

Desinstalar.

Para desinstalar un programa se usa -C mayúscula.

```
emerge -C package
```

También se puede usar la sintaxis

```
emerge unmerge paquete
```

Buscando paquetes nuevos para instalar.

Para buscar paquetes para ser instalados se usa:

```
emerge -s keyword
```

Otra forma de búsqueda, usando su propia descripción:

```
emerge -S keyword
```

Para actualizar los paquetes de nuestro árbol portage se usa la flag sync, que se conecta a un servidor oficial y se descarga el árbol entero, con lo que así descargamos actualizaciones e incluso programas nuevos que antes no teníamos. Esta operación sobrescribe el árbol anterior y lleva un tiempo para su actualización completa.

```
emerge sync
```

Otras opciones.

Pretend.

Prácticamente todos los paquetes están ligados a dependencias que hay que satisfacer para que el programa funcione correctamente. Estas dependencias se averiguan con la flag -pretend ó -p en forma abreviada.

```
emerge -p package
```

Esta opción no instala el paquete, sino que solo nos da información de las dependencias que dicha aplicación instalará también para poder funcionar.

Solo descarga de fuentes.

En el caso de que solo nos interese descargar los archivos sin la necesidad de la compilación se usa la flag `--fetchonly` ó `-f` en su forma abreviada. También descargará sus dependencias.

```
emerge -f package
```

Los paquetes son descargados en `/usr/portage/distfiles`.

Si se quieren descargar todos los programas fuente instalados en un sistema se puede usar:

```
emerge -eD world
```

Actualización completa del sistema.

En cualquier momento se puede actualizar nuestro sistema a las versiones más actuales disponibles. Esto se consigue con las flags `--update --deep`, ó `-uD` en su forma abreviada:

```
emerge -uD system
```

Para actualizar también los programas instalados se usa la orden:

```
emerge -uD world
```

USE flags

Para ver las flags de la variable USE en el `/etc/make.conf` se usa la flag `--verbose` ó `-v` en su forma abreviada. Es útil ya que muchos programas instalan partes especiales a cada entorno. Dichas partes especiales nos pueden interesar y por tanto activar en el `/etc/make.conf`:

```
emerge -pv package
```

Paquetes Masked.

Algunos paquetes están en fase de depuración o testeo ya que no han sido lo suficientemente probados y algunos tienen bugs o fallos de compilación, pero no se consideran inestables para el sistema como mucha gente cree, por lo que emerge lo sabe y nos avisa de ello.

La instalación se puede hacer de 2 maneras.

1- Se puede especificar antes del emerge que el paquete a instalar será mask.

```
ACCEPT_KEYWORDS="~x86" emerge package
```

Si no se quiere estar tecleando eso cada vez se puede especificar como variable en el `/etc/make.conf` aunque no es una buena idea:

2- Se puede especificar a mano los paquetes que queremos instalar, de forma que emerge sabrá en todo momento los paquetes que son mask de una forma mucho más controlada y por lo tanto más eficiente a la hora de desinstalar o actualizar. Esto se consigue alterando los archivos:

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

```
/etc/portage/package.keywords
```

En el primer archivo hay paquetes que están considerados masked.. Todo programa que se encuentre en este archivo NO se podrá instalar, por lo que deberemos borrarlo o comentar la línea que lo nombra.

El siguiente archivo es crucial, ya que en él se especifican los archivos mask que queremos instalar y que no estén en el `/usr/portage/profiles/package.mask` ya que éste último archivo manda sobre el `package.keywords`.

Para instalar una version mask de gimp, por ejemplo, haríamos lo siguiente:

1- Mirar si el paquete está dentro del package,mask

```
grep gimp /usr/portage/profiles/package.mask
```

Si está nos aparecerá el nombre:

```
>=media-gfx/gimp-1.3
```

Editamos el archivo y comentamos la línea:

```
# >=media-gfx/gimp-1.3
```

Aconsejo no borrar nada de los archivos de configuración, así podremos volver atrás en caso de error. Mejor agregar al principio de la línea un “#”.

2- En caso de que no esté nombrado en el archivo package.mask o continúe sin dejarnos instalarlo, deberemos editar el archivo */etc/portage/package.keywords* y añadir la línea siguiente:

```
=media-gfx/gimp-1.3
```

Con esto debería dejarnos instalar el programa mask con control total de emerge.

Mantenimiento de paquetes instalados.

Gentoo lista la última versión disponible e instalada de un programa con la orden -s

```
emerge -s package
```

Pero la herramienta emerge no muestra los paquetes instalados, por lo que es bueno descargarse la aplicación “gentoolkit”

```
emerge gentoolkit
```

Esta aplicación es un aditivo que instala comandos carentes en el sistema que nos facilitará mucho el trabajo con nuestro sistema. Una de esas aplicaciones es qpkg:

```
qpkg -I
```

Que mostrará los paquetes que tenemos instalados. La base de datos se encuentra en la ruta */var/db/pkg/*.

Para ver los cambios de la última versión disponible respecto a la versión que nosotros tenemos se usa la sintaxis:

```
etcat -c mozilla
```

¿Qué paquete ha instalado el program x?

Con tantos paquetes instalados ya no sabemos a cual pertenece el ejecutable x, por lo que si lo queremos saber tenemos fácil solución:

```
qpkg -f/bin/ls
```

que nos muestra que el ejecutable ls pertenece a:

```
sys-apps/fileutils *
```

¿Que paquetes dependen del paquete x?

Aparte de saber los programas instalados y a que paquete pertenece, también podemos saber a que paquete pertenece otro paquete, mediante la flag -d

```
etcat -d paquete
```

Busqueda de duplicados o versiones antiguas.

A veces no se instalan bien las actualizaciones, o se nos a pasado ponerle la orden -u, por lo que podemos listar dichos paquetes duplicados con la orden:

```
qpkg --dups -v
```

Otra opción es hacerlo con la flag -prune ó -P abreviado:

```
emerge -Pvp
```

:: El mundo es nuestro !!

Existe un indice de los programas que conforman nuestro sistema, llamado **world**, del que ya se ha comentado algo anteriormente. Este indice es un archivo de texto formateado en Python y se encuentra en la ruta `/var/cache/edb/world` y lo que hace es llevar un registro de los programas que tenemos en el sistema.

Una forma de actualizar dichos programas es usando la sintaxis:

```
emerge -u world
```

A veces ocurre que hay programas que no los actualiza aunque dispongamos de ellos o simplemente no los encuentra. Si las versiones a las que quieres actualizar no están *masked*, el problema es que no tienes esos paquetes en el **world**. Puedes probar con un **regenworld** que lee el archivo `emerge.log` y después con el `emerge -u world`.

A veces se quiere actualizar un paquete y no nos lo permite aunque hayan actualizaciones. Esto se debe a que se hayan instalado no porque las hayas emergido tú directamente sino que las ha emergido *Portage* por ser dependencias de otros paquetes, sobretodo si se hace mención en las USE flags.

En ese caso, no se registran en **world** y *Portage* no los actualiza en un `emerge world`. Para esto un simple `/var/cache/edb/world | grep paquete` podría darte la respuesta.

Es el caso por ejemplo del `mplayer` y el `xmms` se te instalase como dependencia. En ese caso no se registra en el **world**. La solución sería “emergelo” a mano y se registrará, y la próxima vez se actualizará con el `emerge -u world`.

Otra opción es un `emerge -Du world`, que actualiza también las dependencias, incluso las profundas.