GNOME, preguntas de uso frecuente, PUF (FAQ)

Todd Graham Lewis

David "Gleef" Zoll

GNOME, preguntas de uso frecuente, PUF (FAQ)

by Todd Graham Lewis and David "Gleef" Zoll

v1.0.6a Edition

Éstas son las PMF sobre GNOME, el entorno de trabajo en red orientado a objetos GNU ('GNU Network Object Model Environment').

Table of Contents

1. Introducción a GNOME....8
1.1. ¿Qué significa GNOME?....8
1.2. ¿Qué es GNOME?....8
1.3. ¿Dónde puedo encontrar más información, ayuda o trucos sobre GNOME?.10
1.4. ¿Cómo uso esas listas de correo GNOME de las que he oido hablar?....10
1.5. ¿Cómo accedo a los ficheros de la lista de correo?....11
1.6. ¿Cómo pronuncio GNOME?....11
1.7. ¿Por qué GNOME?....12
1.8. ¿Por qué no CDE o KDE o GNUStep?....12
1.9. ¿Qué plataformas?....13
1.10. ¿Puedo todavía ejecutar mis programs en GNOME?....14
1.11. ¿Qué es un gestor de ventanas, y qué tiene que ver con GNOME?....14
1.12. Has dicho que GNOME es Free Software. ¿Qué significa Free Software?..15
1.13. ¿Es GNOME Open Source(tm) Software?....16
2. Consiguiendo GNOME....18
2.1. ¿Cuáles son los requisitos de sistema para GNOME?....18
2.2. ¿Cuál es el mejor formato para conseguir GNOME en mi máquina?....18
2.3. ¿Qué necesito en mi ordenador antes de instalar GNOME?....19
2.4. ¿En qué orden me ocupo de los paquetes de GNOME?....20
2.5. ¿Como consigo los Tarballs de GNOME?....21
2.6. ¿Cómo consigo GNOME desde el CVS?....22
2.7. Mi único acceso a internet es el correo electrónico. ¿Cómo consigo GNOME? 22
2.8. Yo utilizo RedHat Linux, ¿dónde consigo los RPMs?....22
2.9. Yo utilizo Debian GNU/Linux, ¿cómo consigo los paquetes .deb?....23
2.10. Yo utilizo S.u.S.E. Linux, ¿cómo consigo los paquetes RPM?....24
2.11. Yo uso Caldera OpenLinux, u otras distribuciones de Linux basadas en RPM, ¿cómo consigo los paquetes RPM?....25
2.12. Yo utilizo FreeBSD, ¿cómo puedo conseguir GNOME via Ports?....25
2.13. Yo uso Solaris, AIX, HP-UX, BSDI, SCO, IRIX, u otros sitemas Unix o similares, ¿puede conseguir los paquetes binarios?....26

2.14. Yo utilizo Microsoft Windows, ¿puedo conseguir GNOME? 26
3. Compilando GNOME.....28
3.1. ¿En qué orden compilo los paquetes?.....28
3.2. ¿Como configuro el código fuente para mi sistema? (o... ¿Dónde se esconden los Makefiles?).....28
3.3. ¿Dónde se esconde configure?.....29
3.4. De acuerdo, ahora tengo un Makefile. ¿Cómo consigo instalar GNOME en mi sistema?.....29
3.5. ¿Cómo pongo GNOME en algún sitio especial, como `/opt/gnome` ?.....30
3.6. ¿Cómo compilo GNOME con todos los ficheros en el lugar correcto para que sea Compatible con FHS? ¿Cómo recoloco los directorios etc y var para conectar GNOME a una red?.....32
3.7. Ya tengo este módulo, ¿cómo me deshago de todos los ficheros object y otras "cruft" del árbol del fuente?.....33
3.8. ¿Cómo desinstalo un módulo?.....33
4. Y ahora ya tengo GNOME, ¿qué hago con él? 34
4.1. ¿Dónde miro primero?.....34
4.2. ¿Cómo inicio GNOME? 34
4.3. ¿Cómo ejecuto los programas?.....35
4.4. ¿Cómo ejecuto el Panel de GNOME? 35
5. ¡¡¡¡¡¡¡¡ Ayuda !!!!!!!! 37
5.1. Ayuda General.....37
5.1.1. Algo no funciona, estoy confundido, ¿qué hago?.....37
5.1.2. ¿Cómo informo de un bug?.....39
5.1.3. ¿Qué es un prefix (o un $prefix, ${PREFIX}, o <prefix>)? 39
5.1.4. ¿Qué significa `Fallo de Segmentación` ?.....40
5.1.5. ¿Qué es un core dump? ¿Qué es este fichero llamado core que no paro de encontrarme?.....40
5.1.6. ¿Qué es un backtrace? Alguien me pidió que le enviara uno, ¿cómo lo hago?.....41
5.1.7. ¿Qué es una strace? Alguien me pidió que le enviara una, ¿cómo lo hago?.....44
5.2. Problemas Específicos de Compilación.....44

5.2.1. ¿Qué significa `undefined reference to 'compress'` ?.....45
5.2.2. ¿Qué es `AC_TRY_RUN` , y por qué está mi máquina advirtiendome de ello?.....45
5.3. Problemas con el Runtime.....46
5.3.1. ¿Cómo reseteo las propiedades del manejo de sesiones?.....46
5.3.2. Inicié GNOME con manejo de sesiones, ¿por qué me aparece esta pantalla gris (ni panel, ni nada)?.....47
5.3.3. ¿Cómo le indico al manejo de sesiones qué gestor de ventanas usar?.47
5.3.4. Todo parece funcionar, pero la mayoría de mis iconos son simplemente cuadrados negros, o manchas. ¿Cómo hago aparecer mis iconos?.....48
6. Componentes de la Arquitectura.....49
6.1. Librerías.....49
6.1.1. ¿Por qué necesito tantas librerías para GNOME?.....49
6.2. CORBA, ORBit y Baboon.....49
6.2.1. ¿Qué es CORBA?.....49
6.2.2. ¿Cuáles son algunos de los componentes de COBA? What are some CORBA components?.....51
6.2.3. ¿Cuáles son los servicios de CORBA?.....52
6.2.4. ¿Qué versión de CORBA usa GNOME?.....53
6.2.5. ¿Dónde puedo encontrar más información sobre CORBA?.....53
6.2.6. ¿Qué papel juega CORBA en GNOME?.....54
6.2.7. ¿Qué implementación de CORBA está usando GNOME?.....54
6.2.8. ¿Qué es orbit?.....55
6.2.9. ¿Por qué no ha hecho GNOME un mayor uso de CORBA?.....55
6.2.10. ¿Cómo se usará CORBA en GNOME? o ¿qué es BABOON?.....56
6.3. Gráficos en GNOME.....56
6.3.1. ¿Qué es GTK+?.....57
6.3.2. ¿Qué papel juega GTK+ en GNOME?.....57
6.3.3. ¿Qué lenguajes soporta GTK+?.....57
6.3.4. ¿Qué es imlib?.....58
6.3.5. ¿Qué es eso de los temas (themes)?.....58
6.3.6. ¿Cómo hago para que funcionen los themes?.....59
6.3.7. ¿Qué es OpenGL?.....59

6.3.8. ¿Y qué me dices del soporte Drag and Drop - Arrastrar y Soltar? 60
6.4. Manejo de Sesiones.....60
6.4.1. ¿Qué es el manejo de sesiones? 60
6.4.2. ¿Cómo uso el manejo de sesiones en mi aplicación GNOME?.....61
6.5. DocBook 61
6.5.1. ¿Qué es DocBook?.....61
6.5.2. ¿Qué es SGML?.....61
6.6. Internacionalización y Adapatación Local (I18N & L10N) 62
6.6.1. ¿Que son la Internacionalización(I18N) y la Adaptación Local(L10N)? 62
6.6.2. ¿Que API de I8N y L10N usa GNOME?.....63
6.6.3. ¿Cómo programo usando I18N L10N, o dónde puedo encontrar más información sobre estos temas? 63
6.7. Guile.....63
6.7.1. ¿Qué es Guile? 63
6.7.2. ¿Qué papel juega Guile en GNOME?.....64
7. Asuntos de desarrolladores 65
7.1. Qué tiene que hacer un gestor de ventanas para ser compatible-GNOME.....65
7.2. ¿Qué es Autoconf? ¿Qué tiene que ver con GNOME?.....65
7.3. ¿Qué es Libtool? ¿Qué tiene que ver con GNOME?.....66
8. Convirtiéndose en un GNOMEr.....67
8.1. ¿Cómo me uno al movimiento GNOME? 67
8.2. ¿Cómo consigo las ultimísimas versiones CVS de GNOME? 67
8.3. Quiero el ultimísimo GNOME, pero no puedo usar CVS. ¿Cómo lo consigo? 68
8.4. ¿Cómo consigo una cuenta que me permita contribuir con la versión CVS de GNOME? 68
8.5. ¿Por qué faltan directorios algunas veces cuando actualizo un módulo desde el CVS? 69
9. Sobre las FAQ 70
9.1. ¿Dónde está la copia oficial de estas FAQ? 70
9.2. ¿Cómo añado una pregunta y/o respuesta a estas FAQ?.....70
10. Créditos.....71

10.1. Copyright y Disclaimer.....71
10.2. Trademarks and Naming Credits 71

Chapter 1. Introducción a GNOME

Esta sección abarca preguntas relacionadas con GNOME como concepto global.

1.1. ¿Qué significa GNOME?

GNOME es el GNU 'Network Object Model Environment', entorno de trabajo en red orientado a objetos. GNU significa GNU no es Unix ("GNU's Not UNIX).

1.2. ¿Qué es GNOME?

GNOME es el escritorio GUI del proyecto GNU (http://gnu.org) .

Según un extracto del anuncio original en comp.os.linux.announce, GNOME pretende ser un conjunto completo y gratuito de aplicaciones y accesorios de escritorio intuitivos para el usuario, similar a CDE o KDE pero basado enteramente en software libre.

A mí, el amable responsable de las PMF, me gusta definir GNOME como todo aquello que se espera disponer en un entorno moderno de programación. En este aspecto, es aproximademente equivalente a CDE, win32, OpenStep, o KDE. La gran diferencia es que, a diferencia de los ejemplos mencionados anteriormente, cada componente individual de GNOME es software "libre" o de "código abierto" (http://gnu.org/philosophy/free-sw.html) .

Para los cartógrafos de ahí a fuera, he aquí algunas de las más altas cumbres que encontrarás en la tierra de GNOME:

- GNOME usa la arquitectura universal de agentes de petición de objetos (CORBA) del grupo de manejo de objetos (OMG) que permite a los componentes software interoperar 'sin costuras', independientemente del lenguaje en el que estén implementados, ¡e incluso de la máquina en la que se ejecuten! (Véase la sección CORBA)

- La comunidad GNOME está trabajando duro en el desarrollo de un modelo de objetos llamado ‘baboon’. Basado en CORBA y similar al enlazador y acoplador de objetos de Microsoft, v2 (OLE2), baboon permitirá a los programadores exportar y/o importar recursos ’componetized’. Ésto, por ejemplo, permitirá a los usuarios utilizar cualquier editor que deseen en su entorno de desarrollo, a condición de que su editor soporte a través de CORBA un interfaz de editor estándar (Véase la sección BABOON)
- GNOME no está atado a ningún gestor de ventanas. Puedes escoger tu gestor de ventanas favorito y usarlo con GNOME. (Véase la sección Gestor de ventanas)
- GNOME usa el kit de herramientas Gimp como kit gráfico para todas las aplicaciones gráficas. GTK tiene toneladas de ingeniosas funcionalidades, pero mis tres favoritas son: (1) soporte para multiples lenguajes, incluyendo C, C++, objective C, scheme, perl, y otros; (2) ’themability’, por la cual un usuario puede cambiar el aspecto de todas las aplicaciones GTK en una máquina mientras las aplicaciones están ejecutándose, lo que es increíblemente interesante; (3) y finalmente, la licencia de GTK es LPGL, lo que significa que es totalmente código abierto, o software "libre", al igual que el resto de GNOME. (Véase la seccion GTK+)
- Además de GTK, GNOME usa imlib, una biblioteca para X de tratamiento de imágenes que soporta multiples formatos de imagenes, desde XPM a PNG, y múltiples resoluciones, desde Color Real de 24-bit hasta B/N de 1-bit, todo de forma transparente al programador. (Véase la sección Imlib)
- Todas las aplicaciones GNOME pueden manejarse a traves de sesiones. Ésto significa, p.e., que si cierras el (todavía no escrito) procesador de textos GNOME y lo arrancas de nuevo, abrirá el documento que tenías abierto previamente y situará el cursor de nuevo en la misma posición. Ésto se logra a través del sistema de gestión de sesiones X, tal y como se implementa en el gestor de sesiones GNOME. (Véase la sección Gestión de sesiones)
- GNOME usa el estándar DocBook SGML para toda la documentación, lo que permite a los programadores escribir la documentación de una forma sencilla. Los documentos pueden verse con el visualizador de ayuda de GNOME, o tranformarse a HTML y verse con un visualizador de web, o convertirse a LaTeX o postscript e imprimirse. (Véase la sección SGML)

- GNOME soporta los métodos de adaptación local e internacionalización estándar de Uniforum, permitiendo incorporar soporte para nuevos lenguajes incluso sin tener que recompilar la aplicación. (Véase la sección Internacionalización & Adaptación Local)
- las aplicaciones GNOME soportarán varios protocolos de "arrastar y soltar" (Drag & Drop) para una máxima interoperatibilidad con otras aplicaciones. (Véase la sección Arrastrar y soltar)
- GNOME soporta la programación 3-D usando OpenGL, permitiendo a la gente usar gráficos tridimensionales correctamente con sus aplicaciones GNOME, ya sean éstas cientificas o juegos. GNOME usa Mesa, que es una implementación de código abierto del estándar OpenGL. (Véase la sección OpenGL)

Si todo ésto parece ambicioso, ¡es porque lo es! ¡Sigue leyendo para enterarte de más cosas!

1.3. ¿Dónde puedo encontrar más información, ayuda o trucos sobre GNOME?

La pagina de Gnome es, por supuesto un buen sitio para empezar: http://www.gnome.org (http://www.gnome.org) .

También puedes pensar en unirte a la lista de correo gnome-list, es un lugar muy bueno para pedir ayuda.

1.4. ¿Cómo uso esas listas de correo GNOME de las que he oido hablar?

El mejor lugar para conseguir ayuda e información actualizada sobre GNOME es en las listas de correo. Puedes encontrar un listado general en: http://www.gnome.org/mailing-lists/index.shtml

(http://www.gnome.org/mailing-lists/index.shtml) . La lista de propósito general para ayudas y discusiones es gnome-list@gnome.org (mailto:gnome-list-request@gnome.org) . Para apuntarse a esta lista (o cualquier otra), envia un mensaje a la *request address* , esta dirección es en general la de la lista con el prefijo "-request" añadido al nombre. Por ejemplo la request address para la gnome-list es gnome-list-request@gnome.org (mailto:gnome-list-request@gnome.org) . El subject del correo *debe* ser **`subscribe`** .

En pocos minutos (posiblemente algunos más si no estás en Norte América), deberías obtener una respuesta automática pidíendote la confirmación de la subscripción. Sigue las instrucciones de este correo electrónico y estarás subscrito. Finalmente, el sistema te mandará dos correos con ayuda sobre como usar el sistema (de listas).

1.5. ¿Cómo accedo a los ficheros de la lista de correo?

Hay dos maneras. La más fácil es ir a la página de la lista de correo en http://www.gnome.org/mailing-lists/index.shtml (http://www.gnome.org/mailing-lists/index.shtml) . Haciendo clic en cualquiera de la lista de direcciones de la izquierda, podrás visualizar los archivos de la lista.

Si necesitas realizar una búsqueda en los archivos, puedes usar la dirección de request (request address) de esa lista. Uno de los primeros correos enviados cuando te subscribes detalla las instrucciones sobre como realizar esta tarea. Si has perdido estas instrucciones, email a gnome-list-request@gnome.org (mailto:gnome-list-request@gnome.org) con el subject **`archive help`** . Obtendrás por respuesta un archivo con instrucciones sobre como usar el sistema de archivos.

1.6. ¿Cómo pronuncio GNOME?

GNOME significa “ GNU Network Object Model Environment (Entorno de trabajo en red orientado a objetos GNU) ” . GNU significa “ GNU’s Not Unix (GNU’ No es Unix) ” , y siempre ha sido pronunciado “ guh-NEW ” para minimizar la confusión. Puesto que GNU es el primer nombre de GNOME, GNOME se pronuncia oficialmente “ guh-NOME ” .

Sin embargo, mucha gente pronuncian GNOME como “ NOME ” (como esos pequeños seres de leyenda, los gnomos), nadie se va a enfadar si encuentras esta pronunciación más fácil.

1.7. ¿Por qué GNOME?

Primero, el sistema operativo GNU necesita un entorno de escritorio que sea 100% free software (software libre). GNOME satisface esta necesidad.

La segunda meta, por medio de la primera, es que la gente que use sistemas operativos libres tenga un agradable entorno de usuario en el que trabajar (Esto se aplica también a otros sistemas basados en UNIX no gratuitos).

Finalmente, GNOME tiene un montón de características realmente bien diseñadas que seguro que los ’geeks’ disfrutarán. En palabras de uno de los revisores internacionales de estas FAQ:

> Todd, GNOME tiene méritos tecnicos incluso más allá del "open source" y el "agradable entorno" y la "dominación mundial", como tú bien sabes... Por ejemplo, el clon de COM y el uso de CORBA para la interoperación gives me a woody.

Esto es realmente cierto (el hecho de que GNOME tenga un gran mérito técnico; descozco el aspecto en el woody front). ¡Pruébalo tu mismo y verás!

1.8. ¿Por qué no CDE o KDE o GNUStep?

¡Porque GNOME es mejor! Ya más enserio, cada uno de estos entornos plantean cuestiones que nos animaron a comenzar el proyecto GNOME.

CDE no es Free Software, ni de lejos se le parece. Mucha gente además lo encuentra carente de características especiales, rendimiento y funcionalidad.

En el momento en el que surge GNOME, KDE tenía serios problemas de licencia, que todavía se están intentando solucionar. KDE está basado en una librería no libre llamada Qt, que tiene una licencia muy liberal, pero que no es Free Software (http://gnu.org/philosophy/free-sw.html) . TrollTech (la empresa propietaria de Qt) ha prometido cambiar la licencia de Qt, pero todavía no lo han hecho. Además habían problemas de compatibilidad entre la licencia de KDE y la de Qt. Más que insistir en este tema, te recomiendo leer los artículos Posición de RedHat sobre Qt (http://charlotte.redhat.com/redhat/qtlicense.html) y Posición de Debian sobre KDE (http://www.debian.org/News/1998/19981008) . A la gente de GNOME nos gusta la gente de KDE, y consideramos la actual una situación desafortunada en proceso de arreglarse. Con un poco de suerte, esto dejará pronto de ser un tema de actualidad, y GNOME y KDE podrán competir amigablemente en méritos técnicos y diseño.

Este tema ha sido hashed out una y otra vez en la lista de correo general de GNOME. Plantear esta cuestión en la 'gnome-list' es un comportamiento totalmente desaconsejado. Si quieres ir a algún sitio y empezar una guerra de flames (flame war) sobre este asunto, entonces por favor hazlo en algún lugar muy muy lejano donde no tengamos que oirte.

Hay importantes diferencias de diseño entre KDE y GNOME. La primera de la lista es la diferencia en el widget set. Encontramos a GTK+ más agradable, más adaptable, más amigable para los desarrolladores en varios lenguajes, y más flexible que Qt; otros puede que no estén de acuerdo. En conjunto, los proyectos son lo suficientemente diferentes como para que ambos puedan coexistir e incluso colaborar entre sí.

GNUStep es otro entorno de escritorio que tiene muchas cosas buenas que funcionan muy bien. Otro cosa respecto a GNUStep es que están intentando reimplementar el escritorio y API OpenStep, con GNOME preferiríamos desarrollar algo nuevo y bueno más que simplemente rehacer algo viejo, incluso algo viejo y bueno. Además, así como Qt empuja con fuerza al los desarrolladores hacia C++, GNUStep empuja a los desarrolladores hacia **Objective C. GNOME quiere permanecer agnostico frente a los

lenguajes, y soportar desarrolo en cualquier lenguaje que desees utilizar.

1.9. ¿Qué plataformas?

GNOME fue iniciado por varias personas bien conocidas en las comunidades Linux y GNU, pero está concebido para funcionar en cualquier sistema moderno y funcional basado en Unix. Su funcionamiento ha sido probando en los siguientes:

GNU/Linux
BSD (FreeBSD, NetBSD y OpenBSD)
Solaris
IRIX
HP-UX
AIX
probablemente en algunos otros; a los que no sigo la pista.

Si pruebas GNOME en otras plataformas y funciona, entonces por favor asegúrate de hacérmelo saber para que pueda incluirlas aquí. Si lo pruebas en otra plataforma y tienes problemas, entonces por favor envíanos las dificultade que encontraste, para que los desarrolladores de GNOME puedan trabajar en el problema. (Mira en Informando sobre Bugs .)

1.10. ¿Puedo todavía ejecutar mis programs en GNOME?

GNOME no reemplaza tu sistema, funciona sobre él. Todos los programas que funcionan sin GNOME deberían funcionar bien con GNOME.

1.11. ¿Qué es un gestor de ventanas, y qué

tiene que ver con GNOME?

Un gestor de ventanas es un programa dentro del entorno X Windows que controla la posición y apariencia de las ventanas en la pantalla. Mira un xterm o algo. La caja del terminal de texto es el actual xterm, el borde, barra de título, botones y demás estan controlados por el gestor de ventana. Algunos gestores de ventanas tienen aplicaciones extras, pero todos se encargan de la decoración y posición de la ventana.

Hay muchos gestores de ventana ahí fuera, y GNOME debería funcionar correctamente con todos ellos. Hay unas cuantas características de GNOME (como el manejo de sesiones, y el GNOME pager), que no funcionarán con todos ellos. Para obtener los mejores resultados, hazte con un gestor de ventanas compatible con GNOME. Los dos más compatibles actualmente son Enlightenment y IceWM, con FVWM, SCWM, Window Maker, AfterStep y QVWM dando los últimos retoques a la compatibilidad con GNOME mientras lees esto.

1.12. Has dicho que GNOME es Free Software. ¿Qué significa Free Software?

La mayoría de licencias de software están diseñadas para limitar la libertad de los usuarios a usar, examinar, modificar y distribuir el software. En el proyecto GNOME, y en la comunidad del Free Software en general, creemos que esas restricciones no son éticas, y nos esforzamos por producir software de calidad que esté libre de esas restricciones. Esto es lo que llamamos Free Software. Tradicionalmente, se ha referido a software que permite tres tipos de libertad:

- Libertad para estudiar cómo funciona el programa y adaptarlo a tus necesidades.
- Libertad para redistribuir copias para que puedas compartirlo con tu vecino.
- Libertad para mejorar el programa, y presentar tus mejoras al público, para que toda la comunidad se beneficie.

Más recientemente, una cuarta libertad ha comenzado a ser restringida por las licencias, la libertad para usar el software. Debido a que las licencias de software están basadas el la ley de propiedad (copyright), y esta ley sólo se dirige a la distribución, no al uso, asumimos que la la libertad de uso estaba implícita. No hace falta decir que mientras la libertad de uso no es parte de la definición tradicional del Free Software, el concepto sí incluye esa libertad.

* *Note to translators, please translate the following note, including all instances of the word "free" carefully, except leave the* `free` *in English.*

> **Note:** Si el siguiente párrafo es difícil de entender en la traducción, perdonadme. A diferencia de muchos idiomas, en inglés la palabra usada en “ free beer (cerveza gratuita) ” y “ free speech (libertad de expresión) ” es `free` . Esto causa alguna confusión cuando se está hablando de Free Software.

El Free Software es un asunto de libertad, no de precio. Para entender el concepto, piensa en "Free Speech" y no en "Free Beer". La comunidad Free Software no está de ningún modo en contra del concepto de vender copias del software, o de que los programadores se ganen la vida. No estamos en contra del software comercial, estamos en contra del software propietario. Para más información, mira los artículos en http://gnu.org/philosophy (http://gnu.org/philosophy) .

En el proyecto GNOME, protegemos la libertad del usuario usando la GNU General Public License (GPL) y la GNU Lesser General Public License (LGPL, anteriormente conocida como la GNU Library GPL). Estas licencias son cuidadosamente diseñadas con el legal advice para producir un “ copyleft ” . Que es un trabajo con copyright que garantiza que será siempre redistribuible libremente. Esto se consigue asegurándonos que las modificaciones y trabajos derivados están también cubiertos bajo la GPL.

Las principales aplicaciones de GNOME (y unas pocas librerías) están cubiertas bajo la GPL. La mayoría de las librerías core de GNOME usan la LGPL, para permitir que puedan ser utilizadas otras licencias de software libres (y propietarias) cuando se realize software para GNOME, aunque siempre protegiendo las librerías de ser restringidas.

1.13. ¿Es GNOME Open Source(tm) Software?

En primer término se puede decir que sí. La GPL y LGPL usadas para GNOME están explicitamente listadas como ejemplos de la definición de "Open Source". Sin embargo, intentamos evitar usar el término Open Source para describir a GNOME, ya que no sólo esconde este término deliberadamente la libertad inherente en el Free Software, sino que además el término está siendo utilizado para describir software que no es Free (Software).

El término fue creado para intentar acercar el Free Software a los negocios que no querían oir hablar de Libertad (Freedom). Personalmente, he descubierto que a la mayoría de negocios les gusta oir hablar sobre libertad. La libertad para personalizar su software a sus necesidades, la libertad para cambiar a los desalloradores de software sin cambiar su software, la libertad para dar copias de su software a sus clientes y subcontratistas, son todas estas libertades que el Free Software ofrece, y el propietario normalmente no, las que hacen un *good business sense* . Para saber más sobre la distinción entre Open Source y Free Software, le el artículo de Richard M. Stallman en http://gnu.org/philosophy/free-software-for-freedom.html (http://gnu.org/philosophy/free-software-for-freedom.html) .

Chapter 2. Consiguiendo GNOME

Hay muchas maneras de introducir GNOME en tu sistema. Esta sección cubre las preguntas más comunes que implica el proceso.

2.1. ¿Cuáles son los requisitos de sistema para GNOME?

Actualmente, necesitas una máquina con un sistema operativo Unix (o basado en Unix) instalado, con el Sistema X Window (X11R5 o superior). Necesitas al menos 16MB de RAM, aunque supongo que serás más feliz con 32MB o más. Si sólo estás instalando los binarios, necesitarás cerca de 30MB de espacio en el disco, no esperes necesitar menos 200MB de espacio de disco si vas a compilarlo desde los fuentes.

2.2. ¿Cuál es el mejor formato para conseguir GNOME en mi máquina?

Depende de la clase de ordenador que tengas, y lo que quieras hacer con GNOME. Si tienes dudas, el método oficial y con soporte para instalar GNOME es desde el último paquete de tarballs (los ficheros acabados en " .tar.gz ") desde ftp://ftp.gnome.org (ftp://ftp.gnome.org) o sus servidores réplica (mirrors). Busca en el directorio /pub/GNOME/sources para comprobar los diversos paquetes existentes. La lista completa de sitios FTP réplica está en http://www.gnome.org/ftpmirrors.shtml (http://www.gnome.org/ftpmirrors.shtml) , en cso de emergencia, prueba con ftp://gnomeftp.wgn.net/pub/gnome (ftp://gnomeftp.wgn.net/pub/gnome) .

Si quieres la ultimísima versión en desarrollo de GNOME, CVS es lo que necesitas. No hay garantías de que la versión CVS pueda como mínimo ser compilada, ni mucho menos que funcione, pero los bug reports son muy apreciados. Si crees que esto es para

tí, puedes encontrar mas información sobre como obtener la versión CVS en http://www.gnome.org/devel/whatiscvs.shtml (http://www.gnome.org/devel/whatiscvs.shtml) y http://www.cinternet.net/~soren/gnome/cvs.shtml (http://www.cinternet.net/~soren/gnome/cvs.shtml) .

Si no quieres ocuparte de compilar GNOME, y estás en un sistema común, deberían haber paquetes binarios precompilados disponibles para tí. Actualmente hay paquetes para sistemas RedHat, SuSE 6.0, Debian y LinuxPPC. Hasta hace poco habían paquetes para Solaris también, tan pronto como tengamos unos buenos ofreceremos estos también. Más sobre las diversas distribuciones en http://www.gnome.org/start/getting.shtml (http://www.gnome.org/start/getting.shtml) .

2.3. ¿Qué necesito en mi ordenador antes de instalar GNOME?

Hay unas pocas cosas que GNOME presupone que tienes, si no las tienes, o tienes una versión más antigua, instala o actualiza los siguientes paquetes antes de comenzar el proceso de instalación de GNOME.

libjpeg (6b o superior)
libpng (1.0.1, 1.0.3 o superior *no 1.0.2*
libungif (3.0 o superior)
libtiff (3.4 o superior)
libgr (2.0 o superior)
ImageMagick (4.0.5 o superior)
zlib (1.1.2 o superior)
guile (1.2 o superior)
Berkeley DB (1.85 o superior)

Además, hay unas pocas librerías que, aun no siendo tecnicamente una parte de GNOME, son habitualmente distribuidas con él. Asegúrate de que tienes:

glib (1.2 o superior)

gtk+ (1.2 o superior)
ORBit (0.4.0 o superior)
Imlib (1.9.3 o superior)

Algunos de los timbres y sonidos requieren un gestor de ventanas compatible con GNOME para funcionar. Actualmente estos son Enlightenment, IceWM, y Window Maker. Además tanto FVWM como SCWM están dando sus últimos retoques a su soporte para GNOME. GNOME no incluye un gestor de ventanas, y funcionará correctamente con cualquier gestor de ventanas, estos simplemente aportan unos pocos extras.

Últimamente, hay algunas librerías que son necesarias para unos pocos, programas no críticos en GNOME. Si te estas quedando si espacio o tiempo, puedes saltar esto sin ningún riesgo. Si quieres que suenen todas los timbres y sonidos , querás también lo siguiente:

libgtop (0.99 o superior)
esound (0.2.8 o superior)
gtk-engines

2.4. ¿En qué orden me ocupo de los paquetes de GNOME?

No hay una única respuesta para esta cuestión, pero hay un orden que debería funcionar bien:

glib
gtk+
imlib
orbit
gtk-doc
libgtop
gtk-engines
gnome-libs

gnome-core
control-center
gnome-print
mc
gnome-guile
gnome-python
gnome-objc
gtk–
anything else, as needed, in whatever order

2.5. ¿Como consigo los Tarballs de GNOME?

Las nuevas versiones de los Tarball (p.e. archivos comprimidos del código usando tar y gzip) de GNOME aparecen en ftp://ftp.gnome.org (ftp://ftp.gnome.org) bajo el directorio /pub/GNOME/ (ftp://ftp.gnome.org/pub/GNOME) . Estos archivos son entonces copiados en los servidores FTP réplica, listados en http://www.gnome.org/ftpmirrors.shtml (http://www.gnome.org/ftpmirrors.shtml) . Si tanto el sitio ftp y web están caídos, un mirror relativamente seguro está en ftp://gnomeftp.wgn.net/pub/gnome (ftp://gnomeftp.wgn.net/pub/gnome) .

Estos tarballs son las únicas versiones oficiales de GNOME, y la única versión de la que los desarrolladores tienen garantizado un control directo. Actualmente, las versiones más recientes de cada paquete pueden encontrarse en sources/latest (ftp://ftp.gnome.org/pub/GNOME/sources/latest) , el actual grupo de tarballs para la version GNOME-1.0 están en gnome-1.0/sources (ftp://ftp.gnome.org/pub/GNOME/gnome-1.0/sources) .

Si quieres los últimos tarballs en desarrollo de GNOME, Jim Pick Software (http://jimpick.com) envía snapshots diarias del árbol de desarrollo en formato tarball . Puedes conseguirlos en ftp://ftp.jimpick.com/pub/gnome/snap (ftp://ftp.jimpick.com/pub/gnome/snap) . Generalmente conserva unos dias cada paquete en un archivo, y archivo diff para quepuedas disponer de los snapshots de cualquier otro dia. Ten en cuenta que estas no son versiones oficiales, son snapshots

inestables, y no deberían usarse en un sistema en producción a menos que sepas exactamente lo que estás haciendo.

2.6. ¿Cómo consigo GNOME desde el CVS?

El CVS es una herramienta muy útil, no solo para los desarrolladores, sino también para los usuarios que quieran tener acceso a lo último de lo último del árbol del desarrollo. Sin embargo, el uso del CVS no es un tema para ser tratado en esta sección de de las FAQ. Dos buenos sitios para empezar son la sección " What is CVS? (¿Qué es el CVS?) " del sitio web de GNOME en http://www.gnome.org/devel/whatiscvs.shtml (http://www.gnome.org/devel/whatiscvs.shtml) , y la siguiente sección Cómo hacerse un GNOMEr .

2.7. Mi único acceso a internet es el correo electrónico. ¿Cómo consigo GNOME?

Si estás tan desconectado de internet que sólo tienes acceso al correo, existen remailers en la red que te permiten usar el FTP via correo e. Envía un mensaje escribiendo en el body **help** , sin comillas o signos de puntuación a bitftp@pucc.princeton.edu o webmail.ucc.ie. Puede ser lento, pero te permitirá bajarte los tarballs de GNOME via email. Visita http://www.csdl.tamu.edu/~cchung/cpsc689/ftpmail/ftpmtoc.html (http://www.csdl.tamu.edu/~cchung/cpsc689/ftpmail/ftpmtoc.html) o http://www.cix.co.uk/~net-services/mrcool/stats.htm (http://www.cix.co.uk/~net-services/mrcool/stats.htm) para conseguir una lista actualizada de los sitios FTPMail disponibles.

Los sitios FTP a los que tendrías que acceder están detallados más arriba, en " ¿Como consigo los Tarballs de GNOME? " , simplemente tendrías que utilizar la interfaz de FTPMail más que hacer un FTP directamente al sitio.

2.8. Yo utilizo RedHat Linux, ¿dónde consigo los RPMs?

De forma periódica, RedHat Labs produce los RPMs de las versiones de GNOME para RedHat Linux, y los ponen en su web. La última versión está actualmente en gnome-1.0/redhat (ftp://ftp.gnome.orgpub/GNOME/gnome-1.0/redhat) . Estos se crean para plataformas alpha, sparc y i386. Asegúrate también que consigues los RPMs del directorio noarch, sea cual sea la plataforma en la que estás. Puedes encontrar instrucciones paso a paso para estos RPMs en http://www.gnome.org/start/getting_rh.shtml (http://www.gnome.org/start/getting_rh.shtml) .

Estos RPMs deberían funcionar correctamente en Mandrake Linux, y también en las distribuciones MacMillian y CheapBytes. Es muy posible que estos RPMs tengan problemas con Caldera, S.u.S.E., y otras distribuciones basadas en RPM; sigue leyendo para obtener más ayuda para estos sistemas.

Estos RPMs están compilados con soporte para sonido, lo que es fantástico a menos que no tengas tarjeta de sonido en tu sistema. Si alguien quiere mantener una distribución de RedHat Linux compilada *sin* soporte para sonido, con mucho gusto lo pondremos en el FTP también.

2.9. Yo utilizo Debian GNU/Linux, ¿cómo consigo los paquetes .deb?

* *¡Gracias Havoc! :-)*

Los paquetes Debian más actulizados son distribuidos en los sitos Debian (y mirrors) en debian-unstable (http://www.debian.org/Packages/unstable/x11/) . Instrucciones paso a paso en http://www.gnome.org/start/getting_debian.shtml (http://www.gnome.org/start/getting_debian.shtml) .

Para obtener los paquetes Debian, haz lo siguiente:

```
$

apt-get update


$


apt-get install gnome-control-center gnome-panel gnome-
session gnome-games gnome-terminal gnome-utils
```

Los encargados de los paquetes Debian son voluntarios, y siempre prueban las cosas a fondo antes de lanzar una versión, por lo que si no ves la última versión en formato deb en seguida, por favor se paciente. Mejor aún, si quieres ayudar a preparar los paquetes Debian para su distribución, hay una lista de correo en debian-gtk-gnome@lists.debian.org (mailto:debian-gtk-gnome@lists.debian.org) , e instrucciones sobre cómo acceder a la staging area en http://www.debian.org/~jim/debian-gtk-gnome/README (http://www.debian.org/~jim/debian-gtk-gnome/README) .

Debido a que los paquetes Debian de GNOME ya están siendo distribuidos con Debian, actualmente no los replicamos aquí. No parece haber necesidad de ello. Si hubiera una necesidad real de replicar estos paquetes, simplemente comunicanoslo y lo haremos.

2.10. Yo utilizo S.u.S.E. Linux, ¿cómo consigo los paquetes RPM?

Hay una excelente página de GNOME para SuSE en http://gnome.linuxbe.org (http://gnome.linuxbe.org) (Bélgica) y http://www.tu-harburg.de/skf/Pub/ifmpc118.ifm.uni-hamburg.de/gnome.html (http://www.tu-harburg.de/skf/Pub/ifmpc118.ifm.uni-hamburg.de/gnome.html) (Alemania). Actualmente tienen las instrucciones y los RPMs para SuSE v6.0, con la v5.3 de los RPMs en camino. Visítala para más información.

2.11. Yo uso Caldera OpenLinux, u otras distribuciones de Linux basadas en RPM, ¿cómo consigo los paquetes RPM?

Sólo porque dos distribuciones usen RPM no quiere decir que puedan usar los mismos RPMs, por ejemplo, RedHat y Caldera usan distintas versiones de libc, por lo que probablemente los binarios RPM de RedHat no funcionarán en Caldera. Un truco que puede funcionar es coger los SRPMS de la distribución de RedHat, y usar **rpm –rebuild** en ellos, y entonces instalar los RPMs binarios resultantes. De otra manera, probablemente tengas que seguir las instrucciones de los tarballs.

2.12. Yo utilizo FreeBSD, ¿cómo puedo conseguir GNOME via Ports?

* *¡Gracias Stanislav Shalunov! :-)*

FreeBSD tiene portes y paquetes para varias aplicaciones de GNOME (incluyendo Gnumeric) así como un porte para el proyecto GNOME entero que instalará todos los programas específicos de GNOME via dependencias. Si tienes el CDROM montado bajo `/cdrom` (el nombre del punto de montaje *es* importante) o tienes una conexión de internet en marcha y el tiempo para bajar durante un rato, puedes instalar GNOME en tu FreeBSD box (3.1 o superior, posiblemente inferior también) escribiendo:

```
$
```

```
cd /usr/ports/x11/gnome && make install
```

Para más información sobre la Colección de Portes de FreeBSD puedes visitar http://www.freebsd.org/ports (http://www.freebsd.org/ports) . Si no tienes conexión a internet, pero sí una distribución de docs , puedes también encontrar más información en `/usr/share/doc/handbook23.html` . Este documento también te explicará

como instalar GNOME con paquetes pre-built (binarios) usando **pkg_add** , si tienes el CDROM. Actualmente (para la 3.1-RELEASE), necesitarás instalar los siguientes paquetes: `gnomelibs` , `gnomeaudio` , `gnomecore` , `gnomegames` , `gnomeprint` , `gnumeric` , y `libgtop` . Un listado de las últimas versiones disponibles se encuentra en http://www.freebsd.org/ports/gnome.html (http://www.freebsd.org/ports/gnome.html) .

2.13. Yo uso Solaris, AIX, HP-UX, BSDI, SCO, IRIX, u otros sitemas Unix o similares, ¿puede conseguir los paquetes binarios?

Existían unos paquetes binarios para Solaris que alguien hizo, pero los paquetes parece que han desaparecido (de todas formas era una versión atigua de GNOME). Si aparece un buen conjunto de paquetes binarios para alguno de estos sistemas, probablemente los podamos poner en el sitio FTP.

En ausencia de una versión binaria, vas a necesitar usar los tarballs. Si tu sistema no tiene un compilador, el GNU Project (http://gnu.org) ofrece el GNU C Compiler (gcc) libremente para casi todas las prataformas de ahí fuera. Ve a http://gnu.org/software/gcc/gcc.html (http://gnu.org/software/gcc/gcc.html) para más información.

2.14. Yo utilizo Microsoft Windows, ¿puedo conseguir GNOME?

* *¡Gracias David Wheeler!*

Actualmente la respuesta es no, GNOME no ha sido portado a la la plataforma Windows, hay ya mucho trabajo como para hacer ese porte posible. Si realmente necesitas una versión para Windows, cuenta con tener que hacer mucho trabajo y

esfuerzo realionado.

Tor Lillqvist ha estado portando varias librerías gráficas usadas por GNOME a la plataforma Win32 (Microsoft Windows 95/98/NT). Puedes encontrar más información sobre sus esfuerzos en http://www.gimp.org/~tml/gimp/win32 (http://www.gimp.org/~tml/gimp/win32) . Ya ha portado glib, gtk+, gdk_imlib, y unas pocas otras librerías relacionadas con la generación de gráficos a Win32.

Su meta es hacer funcionar Gimp totalmente en la plataforma (ya funciona parcialmente), pero su trabajo acerca a GNOME a la posibilidad de ser portable a Windows. Varias aplicaciones de GNOME necesitan sus librerías (posiblemente con un poco más de debugging), libpng, gnome-libs y ORBit. Si alguien portara lo anterior, muchas aplicaciones de GNOME podrían ser compiladas y ejecutadas en Windows.

Si estás interesado en trabajar en dicho proyecto, asegúrate de leer cuidadosamente la documentación de Autoconf (http://gnu.org/software/autoconf/autoconf.html) , ya que así es como diferenciamos entre diferentes capacidades de sistema en GNOME. También, presta atención a las librerías y utilidades de Cygnus Solution's CygWin32 (http://sourceware.cygnus.com/cygwin/) . Harán esta tarea mucho más sencilla.

Chapter 3. Compilando GNOME

3.1. ¿En qué orden compilo los paquetes?

Es importante que compiles e instales los paquetes de GNOME en los ordenes correctos. Hay algunas dependencias muy importantes dentro del árbol de GNOME. Este es el orden que sugiero, en el que deberías compilar e instalar cada paquete antes de continuar con el siguiente. En sistemas basados en ELF (como GNU/Linux y FreeBSD), deberías ejecutar también **ldconfig** después de cada paquete, simplemente para asegurarte de que todas las nuevas librerías son detectadas.

glib
gtk+
imlib
orbit
gtk-doc
libgtop
gtk-engines
gnome-libs
gnome-core
control-center
gnome-print
mc
gnome-guile
gnome-python
gnome-objc
gtk–
anything else, as needed, in whatever order

3.2. ¿Como configuro el código fuente para mi

sistema? (o... ¿Dónde se esconden los Makefiles?)

GNOME usa configure (generado por Autoconf) para configurar el código fuente para ser compilado correctamente en tu sistema Configure investigará como está configurado tu sistema, y definirá constantes de compilación para añadir o quitar diversos trozos de código según sea necesario.

Configure acepta varias opciones, escribe **./configure –help** para una lista completa. La más importante es **`–prefix`** , que le dice al sistema donde se va a instalar GNOME. Si omites el prefix, configure asume que lo harás en `/usr/local` . La mayoría de RPMs usan el prefix `/usr` . Muchos usuarios prefieren usar el prefix **`/opt/gnome`** , lo que requiere unos retoques previos, descritos más abajo.

Configure también chequea ciertas variables de entorno. Pon tus CFLAGS a igual sean cuales sean las opciones de compilación que deseas para asegurarte que las pasas, por ejemplo **`–O6`** maximizará la optimización, **`–g`** incluirá signos de debugging, etc.

3.3. ¿Dónde se esconde configure?

Has obtenido el código del CVS, ¿verdad? Lee la sección Convirtiéndose en un GNOMEr que se encuentra un poco más abajo. Si conseguiste un tarball, configure debería estar en el directorio superior del fuente. Si no está ahí, eso es un bug. Mira la pregunta ¿Cómo informo de un bug? .

3.4. De acuerdo, ahora tengo un Makefile. ¿Cómo consigo instalar GNOME en mi sistema?

Desde aquí, todo es bastante sencillo. Escribes **make** , seguido de **make install** . En muchos sistemas (incluido GNU/Linux), esto debería ser seguido de **ldconfig** . Si tienes mucha prisa, a veces puedes saltar **make** , ya que **make install** normalmente hará todo lo que no halla sido hecho todavía. Si quieres empezar todo el proceso y olvidarte de ello, y usas la Bourne shell o sus derivadas (p.ej. ksh o bash), pon los cuatro comandos en una línea separados por **&&** . Por ejemplo:

```
$
```

```
./configure –
prefix=/usr/local && make && make install && ldconfig
```

El **&&** se asegura de que el primer comando ha finalizado correctamente antes de seguir con el siguiente.

Una vez que has hecho **make install** (y **ldconfig** si fuera necesario), puedes continuar con el siguiente paquete.

3.5. ¿Cómo pongo GNOME en algún sitio especial, como `/opt/gnome` ?

A mucha gente le gusta poner GNOME a parte, en un directotio propio. Esto hace que sea más fácil de desinstalar ante una emergencia, más fácil de controlar quien lo usa, te permite exportar el directorio de gnome a otros sistemas, etc, etc. Hacer esto requiere una cierta preparación, y conocimiento de tu sistema. En este ejemplo, pondré GNOME en el directorio `/opt/gnome` . Simplemente reemplaza `/opt/gnome` por cualquier otro directorio, el que quieras.

La cuestión principal es asegurarse de que el sistema puede encontrar los binarios de GNOME en `/opt/gnome/bin` , las librerías en `/opt/gnome/lib` , y los archivos man en `/opt/gnome/man` .

Para decirle al sistema que busque en `/opt/gnome/bin` para encontrar los binarios de GNOME, simplemente asegúrate de que está incluido en la variable de entorno

PATH . por ejemplo: **PATH="/bin:/usr/bin:/opt/gnome/bin"** .

Después, normalmente hay que indicar al loader donde se encuentran las librerías. En la mayoría de Unix comerciales, añade `/opt/gnome/lib` a la variable de entorno LD_LIBRARY_PATH , de la misma manera que anteriormente al PATH . En algunos sistemas, en vez de la anterior se usa LD_RUN_PATH . En los sistemas que usan binarios en formato ELF, como GNU/Linux o FreeBSD, añade el path de la librería al archivo `/etc/ld.so.conf` , y ejecuta **ldconfig** . En sistemas estáticamente linkados, no tienes que preocuparte de nada de lo anterior, sino comprar más espacio de disco, que seguro que necesitarás. Consusta la documentación de tu sistema si no estás seguro de como decirle al loader en que directorios buscar.

También deberías indicarle a man donde encontrar las páginas man instaladas por GNOME. Simplemente inserta una línea en `/etc/man.config` , donde ponga **MANPATH /opt/gnome/man** .

La otra cuestión es que hay un par de directorios en los que GNOME pondrá cosas, y que estarían mejor en cualquier otro lugar. La manera más sencilla de resolver esto es mediante links simbólicos. Escribe lo siguiente:

```
$

mkdir /opt/gnome/share


$

cd /opt/gnome/share


$

ln -s /usr/share/locale locale


$
```

```
ln -s /usr/share/aclocal aclocal
```

Ahora ya está todo preparado para usar `/opt/gnome` como prefijo (prefix) al compilar. Para hacer esto, simplemente da en el comando **./configure** la opción `-prefix=/opt/gnome` . Por supuesto, si quieres algún sitio especial diferente a `/opt/gnome` , simplemente reemplazalo con en directorio a utilizar para todo lo mencionado anteriormente.

3.6. ¿Cómo compilo GNOME con todos los ficheros en el lugar correcto para que sea Compatible con FHS? ¿Cómo recoloco los directorios etc y var para conectar GNOME a una red?

* *¡Gracias Raja R. Harinath!*

Por defecto, GNOME pone todos sus archivos bajo su prefix directory. Aunque que esto hace que GNOME sea fácil de aislar, no es necesariamente el mejor lugar para ciertos archivos en algunas situaciones. En particular, si buscas conectar GNOME, o para una estricta compabilidad con FHS (http://www.pathname.com/fhs/2.0/fhs-toc.html) , puede que quieras recolocar algunos directorios. Para recolocar el directorio etc de GNOME, usa la opción `-sysconfdir` . Para recolocar el directorio var, usa la opción `-localstatedir` .

Por ejemplo, una instalacion de GNOME con FHS podría tener el prefix `/opt/gnome` . Sin embargo, sus archivos etc deberían estar en `/etc/opt/gnome` , e igualmente, sus archivos var deberían estar en `/var/opt/gnome` . Esto tiene la ventaja de permitirte conectar el directorio `/opt/gnome` , mantenerlo como sólo lectura (read-only), y permitir a unas máquinas tener configuración local e información sin

preocuparse de otras máquinas. Para hacer esto, el comando de configuración sería el siguiente: **`./configure -prefix=/opt/gnome -sysconfdir=/etc/opt/gnome -localstatedir=/var/opt/gnome`** .

3.7. Ya tengo este módulo, ¿cómo me deshago de todos los ficheros object y otras "cruft" del árbol del fuente?

Tienes tres opciones, **make clean** , **make distclean** o borrar el directorio. Usar **make clean** borrará todos los object files, built executables, y miscellaneous cruft, es sencillo y seguro. La siguiente opción, **make distclean** debería borrar todo lo que no estubiera en el tarball original. Si haces esto, no podrás desinstalar el módulo . Y lo mismo para borrar el source directory. Basicamente, **make clean** es bueno para un uso general, las otros deberían ser consideradas medidas drásticas.

3.8. ¿Cómo desinstalo un módulo?

Si estás usando un sistema de manejo de paquetes (como RPM), simplemente sigue su documentación para desinstalarlo.

Si has instalado un módulo desde el código fuente, y piensas que quizás quieras desinstalarlo, *asegúrate que mantienes tu source tree* . Simplemente ve al fuente del módulo y ejecuta **make uninstall** . Debería funcionar sin problemas y borrar todos los archivos que instaló.

Chapter 4. Y ahora ya tengo GNOME, ¿qué hago con él?

Ahora vamos a hablar de dar los primeros pasos con GNOME, y las cosas que puedes hacer con él.

4.1. ¿Dónde miro primero?

Hay un tour ligeramente anticuado pero todavía útil en http://www.gnome.org/start/gnometour.shtml (http://www.gnome.org/start/gnometour.shtml) . Realmente vale la pena echarle un vistazo.

También, asegúrate que visitas la Guía de Usuario GNOME en http://www.gnome.org/users-guide (http://www.gnome.org/users-guide) .

4.2. ¿Cómo inicio GNOME?

La forma correcta de iniciar GNOME con todas su características es poner la línea **`exec gnome-session`** al final de tu X startup script. Este script puede llamarse `~/.xinitrc` , `.xsession` o `.Xclients` , dependiendo de tu plataforma y configuración. Por supuesto, asegúrate de que los binarios de GNOME están en tu path, y no hay ninguna línea que empieze por **`exec`** antes de la que acabas de añadir. Con las gnome-core versions 1.04 y posteriores, puedes seleccionar el gestor de ventanas a ejecutar poniendo la variable WINDOW_MANAGER en la que tu elijas.

Ejecutar **gnome-session** tal cual iniciará la sesión por defecto, la primera vez. Esto incluye el gestor de ventanas de tu elección (o el que sea más apropiado si no lo especificaste), el panel, GNOME Midnight Commander, y el visor de ayudas. A medida que uses GNOME, el manejador de sesiones guardará tu escritorio actual. Si

sales (log out), y vuelves a entrar (log back in) más tarde, tu último escritorio aparecerá como cuando saliste.

Si no deseas usar el manejo de sesiones, GNOME funcionará sin problemas sin él. Pon la línea **`exec gnome-wm`** al final de tu X startup script, como anteriormente. El sistema utilizará la variable de entorno WINDOW_MANAGER para seleccionar tu gestor de ventanas, pero no te molestará con la gestión de sesiones. Ejecuta el panel, gmc, o lo que tú quieras, bien manualmente, o en el script de inicio antes de la linea de exec. Por ejemplo:

```
panel &
gmc &
exec gnome-wm
```

4.3. ¿Cómo ejecuto los programas?

Primero de todo, los programas de GNOME pueden ser ejecutados como cualquier otro programa. Vas a la linea de comandos, escribes su nombre, y el programa se inicia.

Lo segundo, puedes ejecutar programas desde el menu del panel. La mayoría de programas GNOME (y algunos no GNOME también) se instalan en el menú principal del panel de GNOME. Algunos programas a destacar de este menú; el Gestor de Archivos ejecuta la versión para GNOME del Midnight Commander, Settings->Editor de Menues te permitirá añadir o quitar cosas del Principal (y otros menus),Settings->Centro de Control GNOME mostrará el programa para configurar la mayoría de GNOME (y algunas cosas como Enlightenment), Juegos->Gnobots es un juego muy adictivo.

4.4. ¿Cómo ejecuto el Panel de GNOME?

El panel de GNOME arranca normalmente desde **gnome-session** , si no ocurre esto, simplemente teclea **panel** en cualquier linea de comandos para iniciar el panel.

Añade cosas al panel mediante el menu principal del panel, o mediante el menu que aparece haciendo clic con el botón derecho en el panel. La Guía de Usuario GNOME (http://www.gnome.org/users-guide) tiene un Capítulo (http://www.gnome.org/users-guide/paneluse.html) entero sobre como usar el Panel.

Chapter 5. ¡¡¡¡¡¡¡ Ayuda !!!!!!!

Incluso aunque hayas seguido todas las instrucciones al pie de la letra, algunas veces surgen problemas imprevisibles. Cualquier cosa puede ir mal, desde un malentendido en las instrucciones, a un bug en el software, o un bug en el software de otra persona. Aquí, intentamos hacer frente a problemas que han tenido con GNOME algunas personas.

5.1. Ayuda General.

Aquí se recogen las cuestiones más generales que surgen a menudo. Los problemas específicos se tratarán más adelante, en este capítulo.

5.1.1. Algo no funciona, estoy confundido, ¿qué hago?

Bueno, lo primero que debes hacer es una pausa, relájate, y respira profundamente. Seguidamente, revisa estas FAQ, la Guía de Usuario, y cualquier otra documentación que tengas a mano y mira a ver si alguna se aproxima a la resolución de tu problema. Lo siguiente es mirar en los archivos de la lista de correo (las instrucciones se dieron anteriormente) para ver si alguien más ha comentado algo sobre tu problema. Después prueba a ver, sea lo que sea lo que estabas haciendo *una vez más* , si la solución de repente aparece por si misma. Si nada de eso funciona, es hora de preguntar en la lista.

Antes de nada, estás suscrito a la gnome-list, ¿verdad? Si no lo estás, las instrucciones sobre como suscribirte se dieron anteriormente . Ahora, cuando pidas ayuda en la GNOME-list, *da muchos detalles* . Escribir " El panel no funciona, ¡esto es una mierda! " no nos ayuda ni a tí ni a nosotros. Aun tendrás el mismo problema, y nosotros tendremos un posible bug del cual no tendremos información. Es más útil (pero todavía insuficiente) decir " Cada vez que selecciono el lo-que-sea del menu principal, el panel entero desaparece de repente; por cierto, ¿qué es este fichero que GNOME no deja de poner en mi directorio personal, y que se llama core ?. "

Lo ideal, es que tengamos cierta información necesaria sobre tí. Lo primero, exactamente ¿que tipo de sistema operativo usas? No, no todo el mundo tiene el mismo sistema que tú. Necesitamos el tipo de procesador: Sparc, MIPS, i386 (esto incluye Pentiums, AMD, Cyrix, etc), PPC, Alpha, ARM, 680x0, et al; necesitamos el sistema operativo: distribución Linux, FreeBSD, HURD, OpenBSD, Solaris, IRIX, CrayOS, OS/400, o lo que sea. También incluye la versión del sistema operativo. Por ejemplo, la máquina en la que estoy ahora mismo es una GNU/Linux RedHat/i386 Rawhide machine, un amigo mio usa una máquina Solaris Sparc 2.6.

Seguidamente, necesitamos algunas vitales y relevantes estadísticas de que hay en tu sistema. En un sistema GNU/Linux, el número de versión del kernel y de libc son casi siempre importantes. En la mayoría de problemas de GNOME necesitamos las versiones de glib, gtk+, Imlib, ORBit y gnome-libs. Si no estás seguro si necesitas algunas de las anteriores, inclúyelas de todas formas. Otras cosas que a veces son importantes son las versiones de Perl, X11, Python, Guile y libpng, a menos que veas un mensaje de error que mencione alguna, puedes no incluirlas. Otra cosa importante es cómo instalaste GNOME. ¿Usaste binarios?, si es así ¿cuáles? ¿Usaste tarballs? ¿Usaste el CVS?, si es así que fechas y/o tags? Si usaste una mezcla de los anteriores, menciónalo también.

Por último, necesitamos el problema en sí. Necesitamos saber exactamente qué hiciste para que apareciese el problema. Necesitamos saber si el problema es permanente o intermitente. Necesitamos saber si el programa mostró algo con altelación al problema. Necesitamos saber exactamente que sucede cuando ocurre el problema. Si quieres ser ambicioso, prueba con algunas herramienta de diagnostico de las que hablo más adelante, como backtrace, e incluye esa información en el mail.

Ahora antes de te prepares para mandar el correo, necesitamos encontrar un buen subject. La Gnome-list suele puede tener muchos mensajes, y tú quieres que el tuyo resalte que necesitas ayuda, y que pase desapercibido entre la gente que no pueda ayudar. Un subject como "GNOME no funciona" es candidato a ser ignorado; uno como "GSM es una mierda (sucks)" pondrá a la gente de malas, y con pocas ganas de ayudar; algo como "La mayoría de los iconos del panel en HP-UX son sólo cuadros negros" es más probable que tenga la clase de atención que necesita.

Pon todo esto junto en un correo electrónico largo y envíalo a la lista. Si seguiste mi

consejo, es muy probable que obtengas una rápida, y meditada respuesta. Si por algún motivo no obtienes una rápida respuesta, no te lo tomes como algo personal, probablemente el mail se perdió entre la masa. Espera un par de días, mira a ver si puedes añadir algún detalle, o una quizás una cabecera más descriptiva, y envíalo otra vez.

5.1.2. ¿Cómo informo de un bug?

Si encuentras un bug, queremos conocerlo, queremos seguirlo, y queremos arreglarlo. Para hacer este proceso lo más eficiente posible, hemos tomado prestado de Debian su excelente sistema de seguimiento de bugs. Instrucciones completas sobre como utilizarlo en http://bugs.gnome.org/Reporting.html (http://bugs.gnome.org/Reporting.html) .

Hay unas pocas cosas que hay que tener en cuenta antes de enviar bugs al sistema. Primero asegúrate de comprobar la página http://bugs.gnome.org (http://bugs.gnome.org) para asegurarte de que tu bug no se conoce todavía en el sistema. Si es un nuevo bug, envíalo a submit@bugs.gnome.org (submit@bugs.gnome.org) . Asegúrate de que incluyes las cabeceras del Paquete y Version como se describe en las instrucciones, esto asegura que será dirigido rápida y automaticamente a la persona adecuada. Asegúrate que incluyes todos los detalles. Mira la cuestión previa para tener una idea de lo que es útil incluir. Si un backtrace o strace es apropiado, desde luego que lo has de incluir. Las instrucciones sobre como hacer esto se dan más adelante.

Después de que envies el bug al sitema. Obtendrás comentarios sobre el bug via email. También puedes seguir la evolución del bug mediante el sistema de seguimiento de bugs en formato web.

5.1.3. ¿Qué es un prefix (o un $prefix, ${PREFIX}, o <prefix>)?

Cuando compilas GNOME, necesitas especificar donde se situará en el sistema de archivos. La opción específica para usar esta característica se llama `-prefix` , por lo que la llamamos el prefijo de GNOME , o prefijo (prefix) para abreviar. La localización de los archivos de GNOME debería ser la misma de un sistema a otro, excepto por el prefijo, por lo que un archivo en `/usr/share/gnome/pixmaps` en un sistema podría estar en `/opt/gnome/share/gnome/pixmaps` en otro, y nos referiríamos a la localización como `$prefix/share/gnome/pixmaps` .

Si instalaste desde tarballs (o CVS), pero nunca les pasaste a **configure** (o a **autogen.sh**) un prefijo, por defecto es `/usr/local` . Si instalaste los RPMS de RedHat, estos usan el prefijo `/usr` . Si todavía no estás seguro, escribe **which panel** , y quita `/bin/panel` del final. Lo que queda es tu prefix.

5.1.4. ¿Qué significa `Fallo de Segmentación` ?

La mayoría de sistemas Unix y similares tienen una característica llamada protección de memoria. Esto significa que un programa puede acceder a algunas partes de la memoria (a menudo llamadas segmentos), pero no a otras. Si el programa intenta acceder a una zona fuera de los segmentos que tiene permitidos, el sistema detecta el problema y muestra un error. Este error se denomina un Segmentation Fault (abreviado " segv " en algunos sistemas), y generalmente, el programa se cerrará de repente, normalmente dejando tras de sí un core dump (ver bajo).

Todo esto se reduce a la existencia de un bug sea cual sea el programa que estabas ejecutando. El bug podría estar en el código fuente de GNOME, podría estar en las librerías compartidas que usa tu programa, o podría estas en el modo en que se compiló el programa. Si no estás seguro si es un bug en GNOME o en otra parte, pregúntalo en la lista. Si estás seguro que el bug recae en GNOME, por favor asegúrate de que está en el sistema de seguimiento de bugs.

5.1.5. ¿Qué es un core dump? ¿Qué es este fichero

llamado core que no paro de encontrarme?

En los primeros días de los ordenadores electrónicos, la memoria de los ordenadores era un manojo de aros magnéticos llamados cores, montados sobre cables. Este tipo de memoria se llamaba memoria core. Cuando un programador quería ver lo que estaba pasando en su programa, extraía una copia de lo que había en la memoria core para así analizarla. Esta copia se llamaba un core dump, y el término ha permanecido hasta mucho después de que las memorias core se quedaran obsoletas.

En muchos sistemas Unix y similares, el sistema producirá automaticamente un core dump cuando un programa falle catastroficamente, por ejemplo cuando se produce un Segmentation Fault (ver arriba). Generalmente pone el core dump en un fichero llamado `core` en el directorio donde estuvieras cuando ejecutaste el programa. Este fichero es muy útil para hacer un debugging y saber por qué fallo el programa, y puede ser usadoo para producir un backtrace (ver abajo). Por otra parte, este fichero puede ser bastante grande, particularmente si no lo usas para el debugging. Su no puedes permitirte tener core dumps, consulta la documentación de tu sistema para averiguar como desabilitar esta característica.

5.1.6. ¿Qué es un backtrace? Alguien me pidió que le enviara uno, ¿cómo lo hago?

Una herramienta de debugging incleiblemente útil es el backtrace. En esencia es una toma de donde estas exactamente en el programa.Se realiza normalmente cuando un programa "crashes" (cierra inesperadamente, a menudo con un segmentation fault u otro error) o se cuelga (deja de funcionar pero no se cierra). En esta respuesta, voy a tener que asumir que estás usando el debugger GDB del proyecto GNU, ya que ese es el más común incluido con GNOME, y el que yo he usado. Si usas otro debugger, consulta tu documentación.

Hay dos maneras de hacer un backtrace. Puedes usar un archivo core que dejó un programa cuando se cerró, o puedes ejecutar el programa desde el debugger. La experiencia me demuestra que ejecutar el programa desde el debugger es más seguro, pero también comentaré como usar los archivos core.

Para ejecutar un programa dentro de GDB, ve a un xterm (o equivalente), y teclea **`gdb <nombre completo y localización del programa>`** , por ejemplo, para hacer un debug sobre gtalk, deberías hacer ***to debug gtalk, you might do **`gdb /usr/local/bin/gtalk`** . GDB no busca el path, por lo que tienes que especificarle donde está el progama. Si quieres ejecutar el programa con argumentos, *no los pongas aquí* . Puedes especificar argumentos de linea de comando después.

GDB se identificará, y mostrará un prompt `(gdb)` . Desde aquí, escribe **`run`** para arrancar el programa. Si quieres ejecutar el programa con argumentos, ponlos aquí. Por ejemplo, para hacer un debug sobre gtalk con el sonido desactivado, deberías escribir **`run -disable-sound`** . Entonces el programa arrancará. Ten presente que el programa irá más lento de lo normal, y usará mucha memoria.

Ahora intenta recrear el problema que tuviste. Haz lo que fuera que hicieres para causar el problema anterior. Una vez que el programa cierre (por crash), deberías ver inmediatamente algo en la sesión de debug, y un prompt `(gdb)` . Si el programa se cuelga, espera hasta estar seguro de que se colgó en el lugar correcto, y entonces aprieta Control-C en la sesión de debug. Tambíen aquí debería aparecer algo de debugging, seguido de un prompt `(gdb)` . En este prompt (sea cual sea la forma en que llegaste), escribe **`bt`** . Entonces respoderá con el backtrace propiamente dicho.

Aquí tienes un ejemplo de una sesión de debug. Si te piden un backtrace, deberías enviarlo todo:

```
$
```

`gdb /usr/local/bin/gtalk`

```
GNU gdb 4.17
Copyright 1998 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public Li-
cense, and you are
welcome to change it and/or distribute copies of it under cer-
tain conditions.
Type "show copying" to see the conditions.
```

```
There is absolutely no warranty for GDB.  Type "show war-
ranty" for details.
This GDB was configured as "i386-redhat-linux"...
```

```
(gdb)
```

run -disable-sound

```
Starting program: /usr/local/bin/gtalk -disable-sound
Program received signal SIGINT, Interrupt.
0x4049290d in __poll (fds=0x808c110, nfds=3, timeout=-1)
at ../sysdeps/unix/sysv/linux/poll.c:45
../sysdeps/unix/sysv/linux/poll.c:45: No such file or directory.
```

```
(gdb)
```

bt

```
#0  0x4049290d in __poll (fds=0x808c110, nfds=3, timeout=-1)
at ../sysdeps/unix/sysv/linux/poll.c:45
#1  0x403c54e9 in g_main_poll (timeout=-1, use_priority=0, priority=0)
at gmain.c:991
#2  0x403c516e in g_main_iterate (block=1, dispatch=1) at gmain.c:789
#3  0x403c58a1 in g_main_run (loop=0x80882e8) at gmain.c:912
#4  0x401cbf1d in gtk_main () at gtkmain.c:475
#5  0x804a629 in main (argc=2, argv=0xbffffaa4) at main.c:54
#6  0x40407c77 in __libc_start_main (main=0x804a570 <main>, argc=2,
argv=0xbffffaa4, init=0x8049e10 <_init>, fini=0x804c7dc <_fini>,
rtld_fini=0x40009c10 <_dl_fini>, stack_end=0xbffffa9c)
at ../sysdeps/generic/libc-start.c:78
```

```
(gdb)
```

Hacer un backtrace con un core dump es muy parecido. Ve al directorio donde está el core dump, y entonces ejecuta **`gdb -c core <nombre completo y localización del programa>`** . De nuevo, GDB no mira en tu path, tienes que especificarle donde encontrar el programa. Todo lo que tienes que hacer es escribir **`bt`** en el prompt, y enviar la sesión entera.

5.1.7. ¿Qué es una strace? Alguien me pidió que le enviara una, ¿cómo lo hago?

El comando **strace** realiza una "Signal Trace". Usarlo te permitirá ejecutar un programa mientras obtienes las señales y llamadas del sistema. Esta es una herramienta mucho más de debugging tipo "verbose" que el backtrace, pero algunas veces necesitas información extra que un strace puede darte. Obtener un strace es fácil, simplemente escribe **`strace -o <archivo de salida> <comando con argumentos>`** en un xterm. Por ejemplo, para hacer un strace al programa gtalk con el sonido deshabilitado, yo haría **`strace -o /tmp/gtalk.strace gtalk -disable-sound`** . Esto guardaría la salida del strace en `/tmp/gtalk.strace` . Strace sí que usa el path, y acepta argumentos de programa en la linea de comandos.

Una advertencia. Strace produce una gran salida (con el anterior ejemplo de strace se obtuvieron 5.498 lineas en cinco segundos). No envies straces a las listas de correo, y solamente envíaselas a los desarrolladores si te las piden especificamente

En sistemas Solaris, el comando **strace** hace algo diferente. El comando **truss** es probablemente lo que espere la persona que te pidió el strace.

5.2. Problemas Específicos de Compilación

5.2.1. ¿Qué significa `undefined reference to 'compress'` ?

En general, significa que se estan teniendo problemas para encontrar o linkar la librería de compresión (libz). Primero, asegúrate de que está instalada. Si no es así, entonces encuentrala e instálala. Debería ir incluida en todas las distribuciones más importantes de sistemas Unix y similares.

Si está instalada correctamente, y todavía aparece el error, entonces comprueba si existe alguna otra copia de libz danzando por ahí. Un lugar donde se esconde a menudo cuando la gente tiene problemas es en `/usr/X11R6/lib` , ya que XFree86 pone una copia allí. Es seguro simplemente borrarla, pero si no estás seguro, renómbrala como algo parecido a `libz.so.renamed` , de esa forma el linker no la encontrará, y si tienes problemas en otra parte por haberla renombrado, siempre puedes volverla a renombrar.

5.2.2. ¿Qué es `AC_TRY_RUN` , y por qué está mi máquina advirtiendome de ello?

Algunas veces, cuando ejecutamos aclocal, autogen o make, aparecerán lineas parecidas a: `configure.in:171: warning: AC_TRY_RUN called without default to allow cross compiling` . No es algo para estar preocupado, y puedes ignorarlo sin riesgos (a menos que estés cross compiling).

Si todavía sientes curiosidad, un cross compiler es un compilador basado en una plataforma que puedes usar para crear código para ser ejecutado en una plataforma diferente. Por ejemplo, alguna gente tiene una versión de GCC instalada en una box GNU/Linux con procesadores Intel que crean binarios para boxes SGI con procesadores MIPS. En este caso estaríamos hablando de un cross compiler.

La macro AC_TRY_RUN se llama para intentar compilar y ejecutar un fragmento de código. Se usa para comprobar si un determinado trozo de código tiene la posibilidad de funcionar en otra máquina o no, y ajustar la configuración consecuentemente. Sin embargo, si estás cross compiling, no puedes ejecutar el fragmento de código, ya que es de una máquina diferente a la que estás. La macro AC_TRY_RUN incluye un parámetro para que diga que hacer en ente caso, pero no hemos probado a cross-compiling GNOME todavía, por lo que sería prematuro configurar este valor. Debido a la ausencia de este valor, aparece un warning.

5.3. Problemas con el Runtime

5.3.1. ¿Cómo reseteo las propiedades del manejo de sesiones?

Mucha gente encuentra el manejo de sesiones muy útil. Sin embargo, algunas veces el manejo de sesiones puede despistarse, y hacer cosas indeseadas como recordar tus propiedades después de que tu hermano pequeño atacara tu ordenador. Si GNOME está correctamente instalado, volver tus propiedades a las por defecto es fácil.

Primero, sal completamente de X. Si estás usando el manejo de sesiones, deberías ser capaz de usar la opción Terminar sesión del menú principal para salir. Si eso no funciona, aprieta las teclas control y alt, y presiona retroceder (borrar). Si eso tampoco ha funcionado, y ahora aparece un login prompt tipo XDM o GDM, cambia a un login modo texto (control-alt-F1 en muchos sistemas, consulta la documentación si no estás seguro, loguea como un usuario diferente y usa **su** en un xterm si estás realmente en un apuro.

Después, para quitar tus propiedades, estando fuera de X, quita el archivo `~/.gnome/session` . La próxima vez que arranques las X, o la próxima vez que loguees desde GDM, deberías empezar con el gestor de ventanas, panel por defecto, el gmc y visor de ayuda.

5.3.2. Inicié GNOME con manejo de sesiones, ¿por qué me aparece esta pantalla gris (ni panel, ni nada)?

Si usas el manejo de sesiones, y no se puede encontrar una sesión, esto es lo que aparece. Primero, ¿has probado a resetear las propiedades del manejo de sesiones (Ver arriba)? Si esto no lo arregla, entonces la **gnome-session** tiene un problema para encontrar el archivo `default.session` . En las versiones gnome-core 1.03 y posteriores el archivo debería estar en `$prefix/share/gnome` . Antes de eso debería estar en `$prefix/share` . Si el fichero no está, probablemente deberías instalar el gnome-core.

5.3.3. ¿Cómo le indico al manejo de sesiones qué gestor de ventanas usar?

Esta es una pregunta más difícil de lo que parece, ya que la respuesta depende de que versión de GNOME estés utilizando. La respuesta más fácil es que uses las propiedades del apartado Window Manager en **control-center** . Si por alguna razón no te funciona, aquí tienes como hacerlo.

En la versión 1.03 del gnome-core y anteriores, el gestor de ventanas se selecciona poniéndolo en un archivo llamado `default.wm` . El archivo que crea el sistema por defecto debería estar en el directorio `$prefix/share/gnome` o en el `$prefix/share` , sea cual sea el que contiene al directorio `default.session` .

El usuario puede poner una versión específica de `default.wm` en el directorio `~/.gnome` . El archivo `default.wm` debería parecese a:

```
[Default]
WM=pera
```

Donde **`pera`** debería ser remplazada por el gestor de ventanas que quieras. Si quieres cambiar el gestor de ventanas por defecto del sistema, también tendrás que cambiar el archivo `default.session` .

En las versiones 1.04 y posteriores de gnome-core, es mucho más fácil. Pon la variable de entorno WINDOW_MANAGER con el gestor de ventanas que quieras, entonces arranca X. Si la variable no está establecida, GNOME ejecutará un script para tratar de encontrar con acierto un buen gestor de ventanas. Puedes editar tambier el archivo `~/.gnome/default.wm` como en las versiones antiguas.

5.3.4. Todo parece funcionar, pero la mayoría de mis iconos son simplemente cuadrados negros, o manchas. ¿Cómo hago aparecer mis iconos?

La mayoría de iconos de GNOME son imágenes de formato gráfico PNG (http://www.cdrom.com/pub/png) , mostradas via libpng y la librería de graficos Imlib. El problema es, que esta parte de Imlib parece ser muy melindrosa, y expresa su malestar mostrando basura (los cuadrados negros son basura más limpia).

Hay unas cuantas cosas que pueden ser la causa de que esto ocurra. El problema más común es una libpng erronea, confundida o inexistente. Si tienes libpng versión 1.0.2, asume que no es la correcta. Usa sólo las versiones 1.0.1, 1.0.3 o posteriores. Que no esté es fácil, si no existe ningún archivo llamdo "libpng.so" o "libpng.so.2", y estás en un sistema con librerías dinámicas, entonces te falta, consigue una copia e instálala. Para ver si está confundida, mira a ver si tienes más de una copia de "libpng.so" o "libpng.so.2", si es así, borra las dos y reinstala la correcta.

Otra posibilidad es la que tengo yo en una de mis máquinas (Una box RedHat con un procesador AMD5x86 y egcs). Si uso alguna optimización cuando compilo glib o imlib, aparece el problema descrito (imlib usa el gmodule portion de glib para cargar el código PNG). Compilar esos dos paquetes sin ninguna optimización funciona bien Eso lo arregla. Por lo que yo puedo decir, esto es un bug en mi compilador, no en GNOME.

También he recibido un informe donde se comenta que el problema podría ocurrir por interferencias de versiones de ImageMagick. Si estás desorientado, quizá quieras intentar quitar o actualizarla.

Chapter 6. Componentes de la Arquitectura

Esta sección trata las diversas piezas que componen GNOME, y qué hacen.

6.1. Librerías

6.1.1. ¿Por qué necesito tantas librerías para GNOME?

GNOME requiere las librerías que requiere por dos grandes razones. Queríamos que GNOME tuviera muchas características útiles, y no queríamos empezar desde cero. Usar librerías comunes para muchas de las características de GNOME le permite a GNOME funcionar mejor y más consistentemente con otras aplicaciones no GNOME.

6.2. CORBA, ORBit y Baboon

Network Objects (en red orientado a objetos) es el nombre intermedio de GNOME. Aquí te mostramos como funcionan.

6.2.1. ¿Qué es CORBA?

CORBA (http://www.omg.org/news/begin.htm) es el Common Object Request Broker Architecture Del documento del Open Management Group (OMG, el grupo de estándares que controla CORBA) ¿Qué es CORBA? (http://www.omg.org/about/wicorba.htm) recibe la siguiente respuesta:

> La Common Object Request Broker Architecture (CORBA), es la respuesta del Object Mangement Group a la necesidad de interoperatividad entre la rápida proliferación de numeroso hardware y software disponible hoy en dia. Llamado simplemente CORBA, este permite a las aplicaciones comunicarse unas con otras independientemente de donde esten localizadas o quien las halla diseñado(...)
>
> El (ORB) es el middleware que establece las relaciones cliente-servidor entre objetos. Usando un ORB, un cliente puede invocar transparentemente un metodo en un objeto servidor, que puede estar en la misma máquina o en una red. El ORB intercepta la llamada y es responsable de encontar un objeto que implemente la petición, pase los parámetros, invoque su método, y devuelva los resultados. El cliente no tiene que saber donde se encuentra el objeto, su lenguaje de programación, su sistema operativo o interfaz. Haciendo esto, el ORB proporciona interoperativilidad entre aplicaciones en diferentes máquinas en entornos distribuidos heterogeneamente y conecta "sin costuras" multiples sistemas de objetos.
>
> En el terreno de las aplicaciones típicas cliente/servidor, los desarrolladores usan sus diseños propios o estándares reconocidos para definir el protocolo que se va a usar entre los dispositivos. Las definiciones del protocolo dependen de el lenguaje de implementación, transporte en red y una docena de factores. ORB simplifica este proceso. Con ORB, el protocolo se define a través de interfaces de aplicación por medio de una especificación independiente del lenguaje de implementación, el IDL (Ed. Interface Description Languaje). Y los ORBs proporcionan flexibilidad. Dejan a los programadores elegir el sistema operativo más apropiado, entorno de ejecución e incluso el lenguaje de programación a usar para cada componente de un sistema en construcción. Más importante, permiten la integración de componentes existentes. En una solución basada en ORB, los desarrolladores simplemente modelan la herencia del componente usando el mismo IDL que usan para crear nuevos objetos, después escriben el código "wrapper" que traduce entre el bus estandarizado y los interfaces heredados.

Vale, ya es suficente. Aquí está la explicación de Todd.:

> ¿Recuerdas la RPC? Ya sabes, las Remote Procedure Calls (llamadas a procedimiento remoto) . Sun las usó como la base para el NFS y el NIS. Microsoft usó el estándar Distributed Computing Environment como la base para su DCOM.
>
> Bien, una llamada a procedimiento remoto es realmente bastante simple. Primeto, defines el procedimiento en algún formato estándar. What is there to a procedure definition? Bien, tienes el nombre del procedimiento, los argumentos que van dentro, y los resultados que

produce. Entonces produces el lado cliente, que le dice a los clientes como introducir argumentos y obtener resultados, y después creas el lado del servidor, que toma los argumentos y devuelve los resultados.

Puedes usar este modelo para crear sistemas bastante poderosos. Simplemente coje el interfaz estándar del sistema de archivos Unix, conviértelo en RPC, y b**ammo, ya tienes NFS. Ten en cuenta que siempre y cuando las entradas y salidas are thrown around de un modo estándar, los clientes que llaman al procedimiento y los servidores que lo atienden no tienen que estar escritos en el mismo lenguaje, ejecutarse en la misma máquina, o incluso funcionar con el mismo sistema operativo o hardware. **This is, truth be told, the killer feature of RPC.

Sin embargo, a medida que la procedural programming (C, Perl) ha sido sustituida por la programación orientada a objetos (Objective C, Java, C++), necesitas algo más que procedimientos en RPC. Necesitas algo que soporte objetos: creando objetos, accediento a los datos del objeto, accediendo a los métodos del objeto, destruyendo objetos. Aquí es donde CORBA aparece: piensa en CORBA como un RPC de próxima generación, simplemente extendido para soportar la programación orientada a objetos. En vez de tener bajo RPC lo siguiente:

```
void foo(int bar); void baz(){return(-1);}
```

tienes bajo CORBA lo siguiente:

```
interface bubba{ void foo(in int bar); void baz() raises (InValid-
Context); }
```

¿La lección con todo esto? CORBA permite un programa independiente del lenguaje, interfaces transparentes a la localización y orientadas a objetos entre componentes de software. Mola...

6.2.2. ¿Cuáles son algunos de los componentes de COBA? What are some CORBA components?

Me gustaría incluir aquí definiciones de algunos de los términos más extraños de CORBA, para qeu la gente pueda captar la esencia de lo que es CORBA y tener una idea general de los componentes que lo componen. Cuando me refiera a un término no

obvio de CORBA, intentaré definirlo aquí.

- *Object Request Broker (agente de petición de objetos)* - (u ORB) un ORB es una pieza de "middleware", como se le llama, a lo que se situa en medio de los clientes y servidores y hace posible la fácil comunicación entre ellos. El ORB es un ente conceptual, que algunas veces toma la forma de una librería compartida y otras toma la forma de un programa externo. El ORB es el responsable de establecer y destruir las sesiones entre clientes y servidores y dirigir (o no), y transportar mensajes entre ellos durante una sesión.
- *Object Adaptor (adaptador de objetos)* - (u OA) un Object Adaptor proporciona el canal por el cual un object server (como el panel gnome) se comunica con el Object Request Broker (ORBit).
- *GIOP/IIOP* - CORBA toma todos estos language-isms and handles orientados a objetos llevándolos entre componentes software orientados a objetos (marshaling, transporte, y demarshaling). Si las dos piezas de software están en diferentes lugares, entonces se usará el protocolo GIOP para mover la información entre ellos. IIOP es una especificación de GIOP para el conjunto de protocolos de Internet; teoricamente se podría hacer funcionar GIOP sobre otra cosa distinta al IP, y en este caso se le llamaría de otra manera distinta. I for one can’t figure out why you would want to. 8^) Si alguna versión de GIOP a parte de IIOP alguna vez se volviera importante, entonces estoy seguro de que ORBit la soportaría, también. (IIOP es a GIOP lo que el HTML es al SGML)

6.2.3. ¿Cuáles son los servicios de CORBA?

Adicionalmente, hay dos partes realmente en CORBA. Está la parte que ya he descrito, que te dice como hacer el RPC objetivizado, como manejar argumentos, como escribir ficheros IDL, como funciona GIOP, etc. y después están los "Servicios de CORBA". Estos son unos servicios built on top de la insfraestructura básica de CORBA que facilitan la programación de objetos distribuidos.

Quizá el más importante de todos estos es el servicio de nombres. P.e., digamos que

tenemos un programa que por alguna razón necesitase servicios de corrección ortográfica (spell-checking services). Algo realmente limpio que puedes hacer (si tu implementación de CORBA es lo suficientemente elegante) es llamar a tu ORB y decir ¡Oye! Necesito un servicio de corrección ortográfica Vé y buscamelo". Entonces el ORB usa el servicio de nombres para ver que objetos están registrados que proporcionen un servicio de corrección ortográfica. Puede despertar al programa que ha realizado la llamada y decir "Puedes encontrar un servicio de corrección ortográfica en el #867-5309". Entonces llamas al 867-5309 y, ¡mirad!, ahi un servicio de corrección ortográfica allí. Ahora, esta puede ser una librería compartida que got mapped into your address space and y ahora estás haciendo llamadas directas de funciones a ella, o podría ser un servicio comunitacio de corrección ortográfica ofrecido como servicio público por el grupo de usuarios de BSD en Mongolia gracia a internet via IIOP. No lo sabes y, excepto por un posible pequeño retardo inducido por el viaje a Mongolia, realmente no te preocupas. Mola, ¿no?

El servicio de nombres, entonces, permite a los objetos registrarse para un uso futuro. Hay otros servicios CORBA, que incluyen un servicio de transacciones, un servicio de seguridad, un servicio horario, un "servicio de relaciones", que puede o no ser lo que crees que es, y mucho más. Mi copia del documento de ServiciosCORBA de marzo del 1997 lista 14 servicios.

Puedes encontra una copia de la especificación de ServiciosCORBA en la página de la OMG en http://www.omg.org/corba/csindx.htm (http://www.omg.org/corba/csindx.htm) . Si estás interesado en implementar alguno de ellos, entonces estoy seguro de que a los desarrolladores de ORBit les encantaría hablar contigo. (Ver ORBit)

6.2.4. ¿Qué versión de CORBA usa GNOME?

El trabajo está actualmente underway debido a que la OMG está desarrollando la versión 3.0, pero por ahora GNOME usa la versión 2.2 de CORBA, que es el estándar actual. No he oido lo que GNOME hará cuando aparezca la 3.0.

6.2.5. ¿Dónde puedo encontrar más información sobre CORBA?

Una buena página jumping-off es la página de CORBA de Linas Vepstas en http://linas.org/linux/corba.html (http://linas.org/linux/corba.html) .

Debido a que parece haber mucho interés en la diferencia entre CORBA y COM, mencionaré este otro documento: http://www.research.att.com/~ymwang/papers/HTML/DCOMnCORBA/S.html (http://www.research.att.com/~ymwang/papers/HTML/DCOMnCORBA/S.html) , DCOM y CORBA Lado a Lado, Paso a Paso, y Capa a Capa (Side by Side, Step by Step, and Layer by Layer).

Finalmente, el proyecto TAO en la Universidad de Washington en St. Louis tiene su propio ORB, TAO. Ese grupo está dirigido por Dr, Douglas Schmidt. Puedes encontrar un montón de información útil en su página sobre CORBA: http://siesta.cs.wustl.edu/~schmidt/corba.html (http://siesta.cs.wustl.edu/~schmidt/corba.html) . Asegúrate de echarles un vistazo a las página de ACE y TAO mientras estás allí; son muy buenas.

6.2.6. ¿Qué papel juega CORBA en GNOME?

CORBA permite la arquitectura de componentes de GNOME. Realiza un papel similar al de COM/DCOM bajo win32.

6.2.7. ¿Qué implementación de CORBA está usando GNOME?

Al principio, el plan era usar ILU (ftp://parcftp.parc.xerox.com/pub/ilu/ilu.html) . ILU tenia varias características beneficiosas, y sobre todas el soporte multilenguaje. Mientras que ILU era (y es) terriblemente limpio, la actitud de Xerox hacia él no era nada clara, y los términos de licencia resultaron ser fatales: ILU no es free software, y

el equipo GNOME resultó incapaz de conseguir que Xerox modificase la licencia. El proyecto GNOME no adoptará tecnología que no sea software libre, y ahí acabó todo.

Entonces el proyecto se estableció sobre MICO (http://diamant-atm.vsb.cs.uni-frankfurt.de/~mico/) . Los principales atractivos de MICO son que tiene un Adaptador de Objetos, es compatible con IIOP, y su licencia se acoge a los términos de la GPL. Sin embargo, actualmente no soporta más lenguaje que C++, y usaba una incleible cantidad de memoria.

Todavia insatisfechos, los GNOMErs comenzaron a trabajar en nuestro popio ORB, llamado ORBit.

6.2.8. ¿Qué es orbit?

ORBit está diseñado para ser multilingual; ILU es una prueba de que esto es posible. Ahora mismo sólo soporta C, pero en el futuro soportará otros lenguajes. (Realmente, ¡lo decimos en serio! Es un proyecto muy muy nuevo, y este es el único motivo por el que está sólo en C). También soporta GIOP/IIOP, el protocolo CORBA de la OMG que permite a diferentes ORBs ser capaces de hablar entre sí.

Finalmente, ORBit está concebido para ser de alto rendimiento. Eso significa poca memoria y alta velocidad. Aquí, ORBit está siguiendo el ejemplo de Flick (http://www.cs.utah.edu/projects/flux/flick/) . Elliot Lee, el instigador del proyecto ORBit, piensa que puede, si violar el spec de CORBA, acercar el coste de una CORBA call darn close al de una llamada a librería estandar, donde una implementación local de un servicio pedido está disponible. Ya veremos.

6.2.9. ¿Por qué no ha hecho GNOME un mayor uso de CORBA?

La razón más importante es que con MICO no se podía trabajar. Consumía demasiada memoria, era sólo C++, etc. ad naus. Con la reciente versión del GNOME ORBit, podemos empezar a desplegar CORBA en el GNOME.

6.2.10. ¿Cómo se usará CORBA en GNOME? o ¿qué es BABOON?

(Esta sección es casi literal de Miguel de Icaza, Jefe de los GNOMErs).

CORBA será usado en varios contextos. Un conjunto de interfaces y librerías de rutinas llamadas "Baboon" serán usadas para simplificar e integrar aplicaciones:

- Componentes: Hay sevidores/librerias pre empaquetados que proporcionan algunas funcionalidades que pueden ser usadas por cualquier aplicación como los servidores de corrección ortográfica, servicios de calendario o otros servicios no gráficos.
- Application embedding and in-place activation: Baboon es un conjunto de interfaces para proporcionar **application embedding y in-place activation que están siendo definidos. Los interfaces Baboon e interacciones están modelados depués de los interfaces OLE2 y OpenDoc.
- Aplicaciones GNOME como provedoras de servicio: las aplicaciones GNOME exportarán sus funcionalidades internas a través de CORBA para permitir a cualquiera reusar su funcionalidad. Esto permite el scripting del escritorio GNOME como en cualquier aplicación que use CORBA o lenguaje de script pueda invocar métodos en un servidor GNOME (por ejemplo, que un script en Perl pueda controlar remotamente para invocar una hoja de cálculo para hacer un gráfico o para hacer un informe de ventas). Alguna gente llama a lo anterior "Automation". A diferencia del mundo COM, el uso de CORBA hace esto de forma transparente a la implementación de un servidor.
- Controles Reutilizables: Otro conjunto de interfaces de Baboon se ocupan de los controles rehutilizables. Esto es similar a los JavaBeans de Sun y las Active-X de Microsoft.

BABOON significa: Baboon Allows Baboon Objects Over Networks (Baboon permite objetos baboon sobre redes).

6.3. Gráficos en GNOME

6.3.1. ¿Qué es GTK+?

GTK+ es el Gimp Toolkit. Fue escrito originalmente como parte del proyecto Gimp, pero se ha convertido en un excelente widget set de propósito general. Más información en http://www.gtk.org (http://www.gtk.org)

6.3.2. ¿Qué papel juega GTK+ en GNOME?

GTK+ es la interfaz que las aplicaciones de GNOME usan para interactuar con el usuario. Es similar al papel de Motif en CDE, Qt en KDE o el-widget-set-que-no-tiene-nombre en Win32.

6.3.3. ¿Qué lenguajes soporta GTK+?

¡Puedes escribir programas GTK+ en millones de lenguajes! El humilde autor de estas FAQ conoce los siguientes bindings:

C (original)
C++
***Objective C
Python
Perl 5
Scheme (Guile)
TOM
Gwydion Dylan
Ada95

Si hay otros, entonces por favor haznos saber cuáles son. (Ver Contactar con el Encargado de las FAQ)

Te animamos a añadir bindings a otros lenguajes que te apetezca. N.b., que los bindings

de Scheme ya tienen una representación formal en el API, y por lo tanto si quieres empezar a trabajar con nuevos bindings, quizá te gustaría empezar por aquí.

Habiendo escrito lo anterior, me tengo que preguntar si no podríamos de alguna manera usar CORBA para cortar este nudo gordiano. Si los desarrolladores de ORBit pudieran igualar el coste de una llamada local CORBA al de una librería compartida normal entonces ¿por qué no CORBificar GTK y trabajar para añadir bindings de nuevos lenguajes a ORBit? Sé, que la sabiduría popular dice que CORBA es demasiado pesado para cosas de mucho volumen como las llamadas a X. Oye, no me culpes por lanzar ideas locas: ¡la razón por la que mantengo las FAQ es que no puedo programar!

6.3.4. ¿Qué es imlib?

Junto con GTK, GNOME usa imlib, una librería de imágenes para X que soporta múltiples formatos de imagen de forma transparente para el programador. Imlib puede incluso usar programas convertidores externos como ImageMagick o pmbplus para convertir formatos de imagen desconocidos a conocidos, por lo que si existe un convertidor para un tipo de imagen, ¡puedes usar casi con toda seguridad ese tipo de imagen en tu aplicación! Imlib también soporta múltiples profundidades de bit, desde de 24 bits hasta 1 bit en monocromo, otra vez de forma transparente al programador, y realiza localización inteligente de mapa de colores y dithering automático para conseguir el mejor rendimiento de tu pantalla. Cualquiera que halla intentado alguna vez hacer esto con sus apps X, o quien halla tenido que soportar el destello de la pantalla cundo sus apps tienen que allocate a private color map, seguro que os gustará esta característica.

Puedes encontrar más información sobre imlib en http://www.labs.redhat.com/imlib (http://www.labs.redhat.com/imlib) .

6.3.5. ¿Qué es eso de los temas (themes)?

Los themes te permiten cambiar el aspecto y sensación de las aplicaciones GTK+ sin recompilarlas. Con los themes, eres capaz de cambiar el theme de todos los apps GTK+

en un display dado a la vez.

En realidad existen dos partes en un theme en GTK+, el motor y la configuración. El motor es una librería compartida especialmente construida con instrucciones sobre la manera en que los themes deben dibujar los widgets (los controles de una ventana, como los botones, barras de desplazamiento y menús). La configuración incluye información sobre fuentes, colores, que motor usar, y algunas veces imágenes gráficas que incorporar en el theme. Juntos, los dos constituyen el Theme, y le dan a GTK+ su flexible aparaciencia.

Para ver algunos ejemplos de cómo es un app GTK+ tematizado, échale un vistazo a: http://gtk.themes.org (http://gtk.themes.org) .

6.3.6. ¿Cómo hago para que funcionen los themes?

Primero, necesitas una versión reciente de GTK+, yo recomiendo actualmente la versión 1.2.1, pero la 1.2.0 funciona bien también. Después consigue los themes que quieras. Una buena selección de themes está disponible en el módulo de motores gtk. Incluso hay más disponibles en http://gtk.themes.org (http://gtk.themes.org) .

Muchos de los módulos incluyen un motor. Necesitas compilar el motor, e instalarlo. Con los motores gtk, esto se realiza de igual manera que con cualquier otro módulo GNOME. Los themes de la website pueden tener distintas instrucciones, consulta su documentación. Algunos de los themes de la website son meras configuraciones alternativas de otro motor, muchos de ellos son para el motor pixmap en los motores gtk.

Una vez que el tema está instalado, puedes elegirlo a través del Centro de Control GNOME, bajo Escritorio-> Selector de Temas. El selector te permitirá elegir entre todos los temas instalados automaticamente, previsualizar la apariencia del tema, y en la versión 1.04 o posterior incluso permitirte la elección de fuente.

* *Puedes ver el estilo de los apps GTK temificados por el autor de las FAQ en: http://www.mindspring.com/~tlewis/screen.jpg (http://www.mindspring.com/~tlewis/screen.jpg) . Comentado debido a un enlace muerto.*

6.3.7. ¿Qué es OpenGL?

OpenGL es un API de renderización de 2D y 3D desarrollado por Sillicon Graphics que se ha convertido en el estándar de facto de la industria. Basicamente es una herramienta para tomar una imagen en 3D, y producir una imagen 2D adecuada para dibujarla en pantalla. Copiando el FAQ de SGI:

> OpenGL is a rendering only, vendor neutral API providing 2D and 3D graphics functions, including modeling, transformations, color, lighting, smooth shading, as well as advanced features like texture mapping, NURBS, fog, alpha blending and motion blur. OpenGL works in both immediate and retained (display list) graphics modes.

GNOME usa una implementación del API OpenGL llamado Mesa. Mesa no es OpenGL per se, ya que es un proyecto Free Software que no ha realizado el proceso de obtención de licencia de marca registrada OpenGL, pero es una fiel representación del API OpenGL. Mesa no es OpenGL(tm) de identica forma que GNU/Linux y FreeBSD no son Unix(tm).

6.3.8. ¿Y qué me dices del soporte Drag and Drop - Arrastrar y Soltar?

GNOME usa actualmente el protocolo Xde de Drag and Drop. XDE está siendo remplazado con una implementación del Protocolo Drag-And-Drop de las X Windows para interoperar con otros toolkits. Más información en http://www.cco.caltech.edu/~jafl/xdnd/ (http://www.cco.caltech.edu/~jafl/xdnd/) .

6.4. Manejo de Sesiones

6.4.1. ¿Qué es el manejo de sesiones?

El manejo de sesiones es una extensión del sistema X Window que permite a las aplicaciones guardar y restablecer su estado. Esto permite al usuario que su escritorio quede como lo dejó, y no que se encuentren con un escritorio limpio cada vez que reentra en la GNOMesfera.

6.4.2. ¿Cómo uso el manejo de sesiones en mi aplicación GNOME?

Dr. Mike de RedHat ha escrito un tutorial sobre este tema:
http://www.gnome.org/devel/start/sm.shtml
(http://www.gnome.org/devel/start/sm.shtml) .

6.5. DocBook

6.5.1. ¿Qué es DocBook?

DocBook es un DTD SGML (un conjunto de etiquetas -tags- para un archivo de texto) que es muy útil para escribir documentación. La mayoría de la documentación de GNOME está escrita en DocBook. Para más información, mira en las páginas de Oasis's DocBook en http://www.oasis-open.org/docbook (http://www.oasis-open.org/docbook) . También hay un buen tutorial para DocBook en http://nis-www.lanl.gov/~rosalia/mydocs/docbook-intro.html (http://nis-www.lanl.gov/~rosalia/mydocs/docbook-intro.html) .

La versión de DocBook, y herramientas relacionadas, como las usadas por GNOME se pueden encontrar en ftp://ftp.cygnus.com/pub/home/rosalia/docware (ftp://ftp.cygnus.com/pub/home/rosalia/docware) . Tanto los tarballs como los RPMs binarios están disponibles allí.

6.5.2. ¿Qué es SGML?

Por rellenar.

6.6. Internacionalización y Adapatación Local (I18N & L10N)

6.6.1. ¿Que son la Internacionalización(I18N) y la Adaptación Local(L10N)?

De la página info del GNU gettext:

Por *internacionalización* , nos referimos al la operación por la cual un programa, o un conjunto de programas metidos en un paquete, es consciente y capaz de soportar multiples lenguajes. Este es un proceso de generalización, por la que los programas son liberados del uso único de cadenas inglesas u otros hábitos específicos del inglés, y conectados a formas genéricas de hacer lo mismo. Los desarrolladores de programas usan varias técnicas para internacionalizar sus programas, algunos de ellos han sido estandarizados. GNU gettext ofrece uno de estos estandars.

Por *adaptación local (localization)* , nos referimos a la operación por la cual, en un conjunto de programas ya internacionalizados, se le suministra al programa toda la información necesaria para que pueda valerse para manejar su objetivo de la manera adecuada para el lenguaje nativo y sus hábitos culturales. Este es un proceso de particularización, por el cual los métodos genéricos ya implementados en un programa internacionalizado se usan de manera específica. El entorno de programación pone varias funciones a disposición de los programadores que ayunda a esta **configuración runtime. La descripción formal de un conjunto específico de hábitos culturales para un país, junto con todas las traducciones asociadas orientadas al mismo lenguaje nativo, se conoce como el locale para este lenguaje o país. Los usuarios consiguen la adaptación local de los programas configurando los valores correctos para variables de entorno

especiales, más que ejecutando esos programas, identificando que locale se devería usar.

6.6.2. ¿Que API de I8N y L10N usa GNOME?

GNOME usa el estándar de internacionalización de Uniforum. Este es el "estándar" inventado por Sun Microsystems y adoptado por el proyecto GNU en su paquete "gettext". ***N.b., this is different from the competing X/Open standard.

6.6.3. ¿Cómo programo usando I18N L10N, o dónde puedo encontrar más información sobre estos temas?

Échale un vistazo al manual online del paquete GNU gettext en http://gnu.org/manual/gettext/html_chapter/gettext_toc.html (http://gnu.org/manual/gettext/html_chapter/gettext_toc.html) .

De manera adicional, puedes encontrar informativo el FAQ sobre I18N: http://www.vlsivie.tuwien.ac.at/mike/i18n.html (http://www.vlsivie.tuwien.ac.at/mike/i18n.html) .

6.7. Guile

6.7.1. ¿Qué es Guile?

Guile es la implementación realizada por el proyecto GNU de Scheme, un dialecto Common Lisp. Más info en http://gnu.org/software/guile/guile.html (http://gnu.org/software/guile/guile.html) . Hay también una excelente información en http://www.red-bean.com/guile (http://www.red-bean.com/guile)

Los documentos de Guile son bastante buenos, pero el estándar Scheme (el maravillosamente, oscuramente llamado R5RS) se puede encontrar en: http://www-swiss.ai.mit.edu/~jaffer/r5rs_toc.html (http://www-swiss.ai.mit.edu/~jaffer/r5rs_toc.html) .

6.7.2. ¿Qué papel juega Guile en GNOME?

Guile es el lenguaje de scripting o "pegamento" para el proyecto GNOME. Del anuncion original:

> Planeamos usar las ***bindings de GTK/Scheme para escribir pequeñas utilidades y aplicaciones. Cuando estas *bindings estén más maduras, debería ser posible escribir aplicaciones completas en Scheme.

Chapter 7. Asuntos de desarrolladores

7.1. Qué tiene que hacer un gestor de ventanas para ser compatible-GNOME

Hay tres grandes cosas:

- Debe ser completamente compatible con ICCCM, especialmente con un apropiado soporte para el Manejo de Sesiones X.
- Debe soportar *hints MWM
- Debe soportar hints GNOME.

Para obtener una información más detallada, visita la página Compatibilidad de Gestores de Ventana para GNOME en
http://www.gnome.org/devel/gnomewm/book1.html
(http://www.gnome.org/devel/gnomewm/book1.html)

7.2. ¿Qué es Autoconf? ¿Qué tiene que ver con GNOME?

Autoconf es una herramienta para producir *shell scripts que automáticamente configuran el código fuente para adaptarse al sistema que estés usando. Autoconf es de valor incalculabre, ya que hace el desarrollo de esos ***configure scripts drásticamente fácil. GNOME lo usa junto a la herramienta relacionada, Automake, que permite a Autoconf producir Makefiles personalizados para tu sistema. Autoconf usa m4 para operar. M4 es un poderoso (pero arcano) macro lenguaje usado por muchas utilidades GNU. La mayoría de las macros usadas para generar estos ficheros se encuentran o en `/usr/share/aclocal` o en el subdirectorio `macros` del arbol fuente.

Si estás usando los tarballs, Autoconf y Automake ya has sido ejecutadas por ti, por lo que no las necesitas en tu sistema. Si usas el CVS, no llegarás muy lejos sin tener versiones recientes de estas herramientas. Puedes encontrar más información sobre Autoconf en http://gnu.org/software/autoconf/autoconf.html (http://gnu.org/software/autoconf/autoconf.html) , y Automake en http://gnu.org/software/automake/automake.html (http://gnu.org/software/automake/automake.html) .

Ten en cuenta, que `configure` , `config.h` , `Makefile` , e incluso `Makefile.in` son todos archivos construidos. Si necesitas cambiarlos, no deberías parchearlos directamente, si no mejor modificar los ficheros usados para crearlos: `configure.in` , `config.h.in` , y `Makefile.am` .

7.3. ¿Qué es Libtool? ¿Qué tiene que ver con GNOME?

Libtool es una importante utilidad de portabilidad. Porporciona una interfaz común para el mecanismo de *building, debugging y mantenimiento de librerías estáticas y dinamicas en la mayoría de sistemas en uso. Funciona en tres fases, el comando **libtoolize** añade soporte libtool al source tree añadiendo el script **ltconfig** . Este script, ***in turn, genera un script específico del tipo de sistema en el source tree llamado **libtool** . El script **libtool** es la parte que en realidad realiza todo el trabajo.

Como con Autoconf y Automake, si usas el CVS, necesitas tener Libtool instalada en tu sistema. Si usas los tarballs, **libtoolize** ya ha sido ejecutada, y **ltconfig** *is self-contained, por lo que no necesitas tener Libtool instalada. Puedes encontrar más información sobre libtool en http://gnu.org/software/libtool/libtool.html (http://gnu.org/software/libtool/libtool.html)

Chapter 8. Convirtiéndose en un GNOMEr

Información general sobre como introducirse más en el proyecto GNOME.

8.1. ¿Cómo me uno al movimiento GNOME?

Si quieres introducirte seriamente en GNOME, realmente tienes que unirte a la lista de correo gnome-list (Ver Listas de correo , anterior). Comprueba la lista de listas de correo para ver que otras listas te interesan. También, asegúrate de que lees el Manifesto GNOME en http://www.gnome.org/about/manifesto.shtml (http://www.gnome.org/about/manifesto.shtml) y los artículos sobre la Filosofía del Proyecto GNU en http://gnu.org/philosophy (http://gnu.org/philosophy) para asegurarte de que sabes en qué clase de proyecto te estás metiendo. No tienes que estar de acuerdo con toda la filosofía para contribuir con GNOME, pero al menos respetar que algunos de nosotros estemos entregados a las ideas mostradas en estos documentos.

Debido a que GNOME es un proyecto de software, es obvio que siempre podemos necesitar a más programadores, pero no tienes que ser un programador para hechar una mano en GNOME. Siempre podemos necesitar a más gente que envie claros informes de bugs. Siempre hay más documentación que escribir. Si hablas más de un idioma, siempre podemos necesitar traductores, tanto con la documentación como con las pestañas, botones y entradas de menú en las aplicaciones. La gente dispuesta a hacer paquetes binarios para sus sistemas serán acogidos con gusto, particularmente si nadie más esta realizando todavía los binarios para ese sistema. Animamos con entusiasmo a la gente dispuesta y capaz de replicar nuestros servidores WWW, FTP y/o CVS a que lo hagan.

8.2. ¿Cómo consigo las ultimísimas versiones

CVS de GNOME?

Si desarrollas para GNOME, o si quieres simplemente mantenerte más al día que con la última versión, deberías tener en cuenta usar nuestro sistema CVS. Hay un servidor principal CVS, para el que necesitas una cuenta, y una red de servidores réplicas de este, a los que cualquiera puede acceder. Esta red de mirrors se llama en conjunto CVS Anónimo (Anonymous CVS), y puedes acceder a ellos mediante anonscvs.gnome.org (con un round-robin DNS incorporado para compensar la sobrecarga). Si pretendes colaborar con código fuente para el proyecto GNOME, al final querrás una cuenta en el servidor CVS principal, pero primero acostumbrate al sistema de CVS Anónimo, para familiarizarte con el uso que hace GNOME del CVS.

Las direcciones sobre como acceder al sistema CVS Anónimo están en http://www.gnome.org/devel/whatiscvs.shtml (http://www.gnome.org/devel/whatiscvs.shtml) . Hay también una muy útil guia en http://www.cinternet.net/~soren/gnome/cvs.shtml (http://www.cinternet.net/~soren/gnome/cvs.shtml) . Los servidores CVS anómimos deberían actualizarse solos

Para usar la versión CVS, necesitas tener GNU tools instalado en tu sistema. Necesistas por lo menos GNU Autoconf, Automake, Libtool y m4. Si tu sistema no las incluye, puedes conseguirlas del sitio web GNU en http://gnu.org (http://gnu.org) .

8.3. Quiero el ultimísimo GNOME, pero no puedo usar CVS. ¿Cómo lo consigo?

Hay algunas ocasiones en las que CVS no es factible. Quizá quieres asegurarte de que GNOME funciona usando simplemente la herramientas nativas de tu sistema en vez de las GNU tools. Quizá estés al otro lado de un cortafuegos que bloquea el puerto CVS. Sea cual sea tu razón, Jim Pick Software ofrece *snapshots diarios de la versión CVS, en formato tarball, disponibles via FTP en ftp://ftp.jimpick.com/pub/gnome/snap (ftp://ftp.jimpick.com/pub/gnome/snap) . La conversión a formato tarball evita la necesidad de las GNU tools también.

8.4. ¿Cómo consigo una cuenta que me permita contribuir con la versión CVS de GNOME?

Si planeas contribuir con código al proyecto, entonces necesitarás una cuenta en el servidor CVS. Miguel de Icaza (miguel@nuclecu.unam.mx (mailto:miguel@nuclecu.unam.mx)) determina quien consigue una cuenta en el servidor CVS; si quieres una, entonces envíale un mensaje detallando que código planeas escribir, junto con un password encriptado. Puedes conseguirlo haciendo lo siguiente:

```
perl -e 'print crypt ("YourPassword", "salt");'
```

Donde la cadena "salt" preferiblemente debería ser una cadena aleatoria de dos caracteres.

8.5. ¿Por qué faltan directorios algunas veces cuando actualizo un módulo desde el CVS?

El CVS posee una excelente característica que tiene un molesto capricho. El CVS permite a un módulo incluir otro módulo, una característica que GNOME usa para organizar mejor las cosas y reducir la duplicidad de código. Desafortunadamente, el comando **cvs update** no se preocupa de comprobar y ver si se han incluido algún nuevo módulo aparte del principal. Si crees que falta alguna pieza, usa **cvs get** en vez del anterior, ya que comprobará la existencia de nuevos módulos incluidos. No te preocupes, siempre y cuando el source tree esté ahí, no intentará hacer un nuevo y total chequeo, si no que actuará al igual que **cvs update** así pues descargando lo que sea nuevo o esté actualizado.

Chapter 9. Sobre las FAQ

9.1. ¿Dónde está la copia oficial de estas FAQ?

La copia oficial y formal de estas FAQ se encuentra en foramto DocBook (SGML) en el servidor CVS en `cvs.gnome.org:/cvs/gnome` , el archivo en sí se encuentra en el módulo `gnome-libs` en el fichero `devel-docs/gnome-faq.sgml` . Todas las demás versiones son copias, portes, reformas, etc.

9.2. ¿Cómo añado una pregunta y/o respuesta a estas FAQ?

La manera más fácil es enviar un correo al coordinador de las FAQ, en gnomefaq@gnu.org (mailto:gnomefaq@gnu.org) . Si la sugerencia es apropiada, editaremos tu información, y la incorporaremos en el lugar apropiado.

Si tienes acceso al CVS, puedes modificar las FAQ por ti mismo, pero preferiría que limitases las modificaciones directas a rápidos cambios de errores evidentes, más que a cambios importantes o incorporaciones.

Chapter 10. Créditos

10.1. Copyright y Disclaimer

Este trabajo es propiedad de Todd Lewis y Gleef (David Zoll) 1998. Esta documentación es free software; puedes redistribuirla y/o moficarlo bajo los términos de la GNU General Public License publicados por la Free Software Foundation; tanto la versión 2 de la licencia, como (a tu elección) cualquier versión posterior.

Este artículo se proporciona como es sin ninguna garantía expresa o implícita. Aunque se ha realizado todo el esfuerzo posible para asegurar la exactitud de la información contenida en este artículo, el autor/mantenidor/contribuidores no asume(n)n responsabilidad alguna por los errores u omisiones, o por daños resultantes del uso de la información aquí contenida.

10.2. Trademarks and Naming Credits

Aquí están algunas de las trademarks menos obvias usadas en este documento.

Linux es una trademark de Linus Torvalds.
Debian es una trademark de Software in the Public Interest
Copyleft y Free Software no son trademarks de nadie, pero son importantes términos en la comunid
Open Source es una service mark de Software in the Public Interest o de la Open Source Initiative o l
Unix , y The X Window System son trademarks del Open Group.
OpenGL es una trademark de Silicon Graphics, Inc.
Todas las demás trademarks son propiedad de sus respectivos dueños.

Si hay alguna trademark que necesita ser listada formalmente en las anteriores, háganolo saber por todos los medios para que pueda ser añadida. NOTA PERSONAL DEL TRADUCTOR REVISOR (version 0.1): Dios mio, despues de mucho trabajo y una agotadora sesion bajo los efectos del alcohol (dicho sea de paso), he terminado la revisión de estas FAQ de GNOME... 28 - oct - 1999 ¡Viva LINUX! ¡Viva GNOME!

¡Vivan los traductores del GDP-es! ¡Viva Barreiro! (por su paciente saber estar...) Oh Dios mio no lo he guardado...