



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Una aproximación a la Telefonía 2.0: Asterisk™

Manuel Camargo Lominchar

13 de mayo de 2012



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERO TÉCNICO EN INFORMÁTICA DE SISTEMAS

Una aproximación a la Telefonía 2.0: Asterisk™

- Departamento: Lenguajes y sistemas informáticos
- Director del proyecto: Manuel Palomo Duarte
- Autor del proyecto: Manuel Camargo Lominchar

Cádiz, 13 de mayo de 2012

Fdo: Manuel Camargo Lominchar

Agradecimientos

Me gustaria agradecer y/o dedicar este texto a ...

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright © 2012 Manuel Camargo Lominchar.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Notación y formato

Aquí incluiremos los aspectos relevantes a la notación y el formato a lo largo del documento. Para simplificar podemos generar comandos nuevos que nos ayuden a ello, ver `comandos.sty` para más información.

Cuando nos refiramos a un programa en concreto, utilizaremos la notación:
asterisk.

Cuando nos refiramos a un comando, o función de un lenguaje, usaremos la notación:
`quicksort`.

Índice general

1. Motivación y contexto del proyecto	1
2. Desarrollo del proyecto	3
2.1. Elección del Servidor	3
2.2. Elección e instalación del Sistema Operativo en el Servidor	4
2.3. Virtualización para Asterisk	7
2.4. Instalación de la Máquina Virtual	8
3. Caso de Uso: UCA Autos	13
3.1. Introducción	13
3.1.1. Instalación del sistema Asterisk	14
3.1.2. Configurando equipos y el entorno SIP	16
3.1.3. Configuración del Primer Plan de Marcación	19
3.1.4. Introduciendo al proveedor de Telefonía	21
Bibliografía y referencias	25
Software usado	25
Instalación de L^AT_EX	28
GNU Free Documentation License	33
1. APPLICABILITY AND DEFINITIONS	33
2. VERBATIM COPYING	34
3. COPYING IN QUANTITY	34
4. MODIFICATIONS	35
5. COMBINING DOCUMENTS	36
6. COLLECTIONS OF DOCUMENTS	37
7. AGGREGATION WITH INDEPENDENT WORKS	37
8. TRANSLATION	37
9. TERMINATION	37
10. FUTURE REVISIONS OF THIS LICENSE	38
11. RELICENSING	38
ADDENDUM: How to use this License for your documents	38

Indice de figuras

2.1. Capacidad inicial del disco duro en la Maquina Virtual	11
3.1. Capacidad del disco duro tras la instalación completa de Asterisk	15
3.2. Configuración de una Extensión con el Programa X-Lite v4.1	20
3.3. Emacs + AucTeX	29
3.4. Interfaz de Dia	30

Indice de tablas

Capítulo 1

Motivación y contexto del proyecto

Capítulo 2

Desarrollo del proyecto

2.1. Elección del Servidor

A la hora de plantearse que Servidor va a ser más adecuado para nuestro sistema aunque tenemos múltiples opciones, hay que tener presentes varios aspectos. Hay que considerar que aunque Asterisk, sea ampliamente reconocido por funcionar en cualquier sistema, dado que realmente se trata de un software que funciona bajo un sistema de tipo *NIX, el rendimiento no depende tanto en exclusiva de las prestaciones de la máquina sino de las posible incidencias colaterales, y a su vez de ciertos aspectos que deben preservar un sistema de funcionamiento crítico como es el sistema de telefonía en términos generales (Considerando que hasta hoy en día, ni siquiera una persona física puede prescindir un solo día de su teléfono en activo).

Entre los equipos más populares destacan las marcas Dell y HP por su extensiva y conocida fiabilidad entre la comunidad Asterisk. Es bien conocido uno de los problemas populares surgidos en general con los servidores, dificultades con el sistema de interrupciones IRQ, dado que algunos servidores de alta producción pueden llegar a contener múltiples tarjetas PCI de comunicaciones y hacer un uso intensivo de las mismas. Pero también hay que considerar que hoy en día, gracias a las nuevas tarjetas PCI-x y también la posibilidad de realizar las comunicaciones a través de dispositivos externos al servidor (Gateways) ya no es esto tan crucial. Por eso a nivel de marcas, siguen prevaleciendo estas dos, pero solo por el soporte de calidad que ofrecen.

Considerando la opción Dell por la relación calidad-precio principalmente, lo correcto sería plantearse aun el más bajo de gama, algo con prestaciones exclusivas de servidor. Por un lado la posibilidad de tener fuentes de alimentación redundantes, además de tener dos o cuatro sockets para múltiples procesadores, y una placa base con opción a poner memorias registradas. En otro caso hasta un equipo MIY, podría hacerle la competencia a cualquier servidor de Marca como veremos mas adelante.

Otro aspecto interesante, pero no imprescindible, es la capacidad de acceder a una tarjeta específica de las marcas para realizar un sistema RAID por hardware, y opcionalmente, la disponibilidad específica, de cajas tipo enrackables para adaptarse óptimamente a nuestros respectivos CPD.

Me gustaría ofrecer una opinión personal en términos generales. No merece la pena gastarse el dinero que cuesta un servidor si no vamos a tener las prestaciones especiales que ofrece una placa de servidor y todo lo que le rodea (posibilidad de fuentes redundantes, memorias registradas, dos o cuatro sockets, hotswaps, controladoras RAID de alta gama, etc.). Hoy en día por un bajo coste, podemos tener un equipo del nivel de un servidor. Más curioso aun, los servidores en formato de caja de torre de las marcas, no aportan un precio mejorado a mayores prestaciones, si bien, no requieren de rack, en caso de poseerlo,

es hasta un mayor inconveniente de espacio y desorden.

Para entornos de producción pero con necesidades reducidas, quizá la opción más interesante, podría ser tener un servidor de bajo nivel tipo MIY y con opción de registrar todos los terminales en otro servidor paralelo redundante, a través de comunicaciones de datos, en caso de avería.

Para este PFC, me planteo un servidor cualquiera de pruebas, y hacer experimentos.

Pero me gustaría ir un poco más allá. La primera “practica” va a ser montar un pequeño servidor de Producción que cumpla las funciones generales que aportan los servidores Asterisk: Servidor de FAX, Sistema IVR, un pequeño Call-Center para dar soporte a la empresa a nivel IT a través de la VoIP y las líneas analógicas que poseeremos, etc. . .

Para ello cuento con un equipo dedicado a efectos de servidor, con los siguientes componentes internos, procesador con 4 núcleos, 8Gb de RAM y 2 discos duros SATA2 (para funciones de RAID), y demás componentes clásicos.

Además, conectada internamente posee una tarjeta de bajo calibre, una Digium Wildcard B410P con 1 puerto FXO y 1 puerto FXS (para acceso telefonía analógica PSTN).

2.2. Elección e instalación del Sistema Operativo en el Servidor

Como he comentado anteriormente, una de las “bondades” de Asterisk es que aun no siendo realmente multiplataforma, debido a que esta diseñado exclusivamente para sistemas *NIX, dado el basto abanico que disponemos para este tipo de sistemas, existen unos mas interesantes que otros en cuestiones específicas de optimización de recursos. Hoy en día el 99 % de las instalaciones de Asterisk (y de otros tipos de sistemas servidor, como servidores ftp, web, etc.), se realizan sobre sistemas GNU/Linux. Pero para este apartado, el tema a tocar, seria concretamente la decisión de hacerlo, ¿sobre que distribución?

Siguiendo la línea, realmente hoy en día todas las distribuciones son validas, incluso distribuciones basadas en UNIX, desde Open Solaris, incluso distribuciones para dispositivos con un sistema operativo embebido como openWRT, y evidentemente, cualquier otra distribución GNU/Linux específica, tipo RedHat, OpenSuse, Fedora y demás.

Pero en el mundo Asterisk las distribuciones mas destacadas por la comunidad son CentOS y Debian por igual. De todas formas esto no resulta nada nuevo ya que hoy en día, CentOS y Debian se reparten el pastel de los servidores, incluso por encima de Fedora y RedHat Enterprise.

Debian es para mí personalmente, la distribución que mas he tenido la oportunidad de manejar. Y además entre la comunidad de desarrollo, Debian esta cogiendo cada vez más popularidad, aunque existen ciertas Distribuciones, específicas de Asterisk y cada vez más reconocidas como Elastix, o Asterisk NOW! basadas en CentOS.

Pero realmente, existe una distribución dentro del mundo Debian que simplifica masivamente la vida de configuración a nivel Hardware, Ubuntu en su versión específica para servidores, Ubuntu Server. Es cierto que a día de hoy no es una personificación del máximo rendimiento y la optimización de los recursos del sistema ya que de por si añade un consumo “cabecera” (overhead) a la CPU que no es nada positivo para un sistema con una función específica, como cumplirá en este caso lo que aquí estamos

tratando, un sistema PBX.

Pero ahora volviendo atrás al momento de la decisión de configuración de Hardware me encontraba con una simple cuestión: nuestra maquina de pruebas, tampoco iba a servir como un sistema de propósito específico y de alto rendimiento. A efectos prácticos de prueba, no es la intención servir como Call Center para cubrir 500 llamadas simultaneas. Además Asterisk infraescalado en prestaciones, ni aun con un sistema como Debian en su versión mas ligera, seria capaz de soportar esto. Es por esto quizá, por lo que la facilidad de autoadaptación del Hardware en el sistema fue por ultimo mi decisión final de utilizar Ubuntu Server como sistema.

El instalador, en términos generales, es relativamente sencillo de seguir, y tras instalar Ubuntu Server pude comprobar como prácticamente todo el Hardware fue reconocido, y autoconfigurado, desde la Controladora RAID, hasta la VGA en alta resolución (no es gran cosa, pero personalmente me resulta muy útil trabajar en 1920x1080 incluso en consola si el sistema ofrece esta posibilidad como es el caso. Evidentemente, solo faltaba por reconocer la tarjeta Digium B410P descrita anteriormente.

La segunda parte quizá mas significativa de la configuración del sistema fue la del planteamiento de estabilidad del sistema a nivel de Disco Duro. En este ejemplo disponemos de dos discos duros de semejantes características, por lo que se veía motivada la necesidad de montar un RAID. Es determinante, que la mayoría de las placas base, a pesar de traer un controlador RAID, realmente no es un controlador puro y dedicado como podrían ser los controladores PERC de Dell. Considero un controlador dedicado, a ese controlador capaz de liberar a la CPU de la carga de realizar todas las tareas para que el RAID se viera satisfecho. En este caso mi intención era montar un RAID 1 (tipo Espejo).

En general la mayoría de las placas base suelen traer controladores Silicon Image, ATI Nvidia, VIA, etc., pero realmente no dejan de ser controladores de tipo software, gestionados desde la BIOS con solo las características básicas para poder proveer de información suficiente de como debería trabajar (pero no capaces de trabajar autónomamente). Esto podría resultar interesante, excepto por una cuestión muy importante: Si esta controladora se averiase (o mas común, si la misma placa se averiase), necesitaríamos para recomponer ese RAID una controladora exactamente igual, o de semejantes características, capaz de recomponer eso. Hoy en día, esto puede resultar excesivamente complejo, ya que la mayoría de las piezas quedan descatalogadas en poco menos de 2 años, y pese a que continua su producción, a partir de los 4 años es prácticamente imposible conseguir una a no ser que la busquemos a través de un Bróker, o que la pidamos a la marca bajo demanda (con un coste 10 veces mas caro de lo que costó originalmente repercutiendo en definitiva sobre el TCO del sistema severamente). Esta dependencia al Hardware, para mi me resulta inviable. Si tuviéramos la opción de tener controladoras especificas como las PERC de Dell que comentaba antes, que siguen ciertos estándares al menos dentro de la misma marca, y que su producción se extiende mas allá de los 5 años, además de ofrecer un hardware dedicado y liberando a la CPU de esta carga, entonces definitivamente si podría decidir, que merece totalmente la pena.

Entonces, en este caso, teniendo dos discos duros iguales, y la intención es montar un RAID 1, la solución bajo mi criterio más fiable nos la ofrece el propio sistema operativo. GNU/Linux provee la solución, Md Raid, con mdadm. El RAID software por excelencia.

Para configurar esto en un sistema de pequeña escala la idea es muy sencilla:

Ubuntu ofrece en el propio instalador la opción de configurar esto de manera grafica y eventualmente es muy práctico ya que se establece justo desde el momento de la inserción del primer dato del sistema, pero no necesario, ya que mdraid provee de un sistema de sincronización desde el momento de la puesta

en marcha. En cualquier caso, por si a alguien le interesara configurarlo en modo Consola el concepto lo describo a continuación:

1. Para cada disco tenemos que crear primero una partición de tipo RAID arrancable (exactamente iguales, ocupando el disco completo, y con el flag B de bootable).

Podríamos utilizar una herramienta de particionado tipo cfdisk. Simplemente, toca, crear una nueva partición con **New**, en Tipo, ponerle **fd** (Particion Linux Raid), en flag, ponerle **bootable**, y finalmente **“Write”** para escribir las modificaciones sobre la partición. Repetir este proceso para el otro disco.

2. En segundo lugar, toca “instalar” el Raid software con mdadm. En el instalador simplemente hay que especificar algunos parámetros, como el tipo de raid (1, mirror, espejado, misma información en los dos discos duros), el numero de discos involucrados, el numero de discos sobrantes (Spares, como en el sistema JBOD por si acaso) y seleccionar los discos involucrados. Y el sistema raid estará listo en unos segundos.

En el caso shell es quizá aun más sencillo:

Considerando que los discos son SDA y SDB (sata disk A y sata disk B siendo ambos discos tipo sata, si fuera IDE/PATA, seria hda, y hdb)

mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1 (el 1 es la partición 1 que creamos anteriormente, una sola partición).

En este caso habremos llamado a nuestro recién creado RAID md0 a nivel de sistema.

3. Acto seguido, toca particionar nuestro recién creado RAID. En este caso, seria como si tuviéramos un solo disco duro. Podríamos directamente particionarlo con ReiserFS, Ext3 o Ext4, o podríamos elegir un formato mas avanzado como LVM2. Yo personalmente soy más partidario de utilizar LVM por la opción de hacer una copia de seguridad del sistema completo utilizando Snapshots. Un RAID 1 aunque ayuda al uptime del servidor, ya que aunque haya un error de disco, el sistema seguirá en pie, pero no asegura que la información se destruya en un momento determinado (o peor aun, se corrompa). La única forma de preservar la fiabilidad de la información es utilizar las copias de seguridad. Y los snapshots de LVM son la forma idónea de realizarlas sin tener que parar el sistema, y mantener el 100 % de tiempo de actividad en nuestro sistema.

Sobre la estructura de particionado, existen muchas formulas, pero una de las posibles seria la siguiente. En primer lugar, una partición para el sistema de arranque. Esto no es estrictamente necesario pero, realmente es una buena práctica poner el sistema de arranque en un tipo de partición más sencilla como ext2 que la partición del sistema principal, que en este caso seria ext4. Yo suelo darle bastante cantidad de espacio, del orden de 10 veces más de lo que necesita. Con 50 Mb es suficiente, pero si en un momento determinado vamos cargando múltiples kernels para cualquier asunto, o actualizaciones sucesivas, y no tener la necesidad de ir borrando lo antiguo, suelo poner unos 500 Mb. Con los discos duros de hoy en día no supone un gasto indebido y nos curamos en salud

En segundo lugar el espacio Swap. Hay que considerar una cosa: Si tenemos que utilizar el sistema Swap en nuestra maquina Asterisk, vamos a tener serios problemas. Yo personalmente me plantearía elegir dedicar 0 Mb. Pero con mi política de reservar espacio para todo y que en un momento dado, aunque el sistema vaya mal, al menos no vaya a pique con facilidad, me gusta reservar tanta memoria

Swap como memoria RAM disponible haya.

El resto, volumen principal, en ext4 que es el más reciente en estos momentos, y cumple su función. Los sistemas Ubuntu lo permiten y Asterisk funciona perfectamente en este tipo de sistema.

2.3. Virtualización para Asterisk

Generalmente, siempre que sea posible evitarlo, no es una gran idea Virtualizar Asterisk, por la principal deficiencia que presentan todos los sistemas de Virtualización: la escasa optimización de los recursos por parte de los sistemas virtualizados. Se suele optar por sistemas “sobre el metal” (bare-metal) a la hora de diseñar plataformas Asterisk puesto que es la forma de obtener el máximo exponencial dedicado para su uso.

En contrapartida, la virtualización, ofrece grandes bondades que en ciertos entornos pueden superar ampliamente, a la principal deficiencia. La característica más positiva que nos ofrecen, es la capacidad de restablecer un sistema en tiempo record y que simultáneamente, es un símbolo de protección por su efecto de back-up.

Existen múltiples sistemas de virtualización, pero los mas reconocidos, en el mundo del software libre son tres: OpenVZ, Xen, y KVM.

OpenVZ surge como evolución del sistema Virtuozzo, extensamente conocido en el mundo de los Alojamientos web. La principal ventaja de OpenVZ es que se integra con el sistema host perfectamente, y una de las limitaciones es que tanto el host como el cliente han de ser sistemas GNU/Linux (esa es la principal razón de esta magnifica integración).

Por otro lado, Xen, nacido en 2003, es un sistema de virtualización basado en maquinas cuyos sistemas operativos están diseñados específicamente para trabajar sobre él. Ha habido múltiples evoluciones en lo que este requerimiento se ha flexibilizado bastante (específicamente, para sistemas operativos cuyo código no es abierto como Microsoft Windows, para poder “saltar” esta barrera original). Se convirtió en la solución de código libre mas distendida en términos de virtualización.

En última instancia surgió KVM como nuevo competidor, a raíz de que el Kernel de Linux (versión 2.6.20 año 2007) permitía por primera vez utilizar las extensiones Intel-VT y AMD-V de virtualización. Pretendía que entrar en el mercado dominado por VMWare en el mundo propietario, y Xen en el mundo Open Source. Por otro lado también emergían nuevas soluciones como Virtual Box de Sun Microsystems, que a posteriori sería absorbida por Oracle.

En el origen surgían varios inconvenientes, era necesario tener un procesador con soporte para virtualización por hardware, como son los procesadores con tecnología AMD-V e Intel-VT, el número de seguidores evidentemente era limitado, así como el apoyo económico y tecnológico que ofrecían las grandes marcas (RH, Novell, Citrix...) y además surgió exclusivamente como un sistema basado en línea de comandos, bastante complejo, sin posibilidades de adaptarlo a interfaces gráficas de forma relativamente aceptable.

Pero hoy en día todo esto quedó atrás. Por un lado, casi todos los procesadores soportan la virtualización por hardware. A lo mejor maquinas antiguas sufren esto, pero ya es realmente raro mantener un servidor de tal antigüedad con semejantes características, y ni siquiera plantearse el hecho de mandarlo a la

guerra de las maquinas virtuales con semejante antigüedad.

Es curioso como la curva de seguidores, en una balanza, lleva tiempo inclinándose a favor de KVM. Desde la lectura de estas líneas KVM ha superado ampliamente a la comunidad de XEN, de hecho XEN cada vez pierde mas “seguidores” a costa de KVM como puede verse en la mayoría de las estadísticas (ejemplo, Google Trends).

Sobre los apoyos tecnológicos, económicos, y técnicos, la verdad es que KVM empezó a gozar del apoyo de uno de los principales actores en el ámbito GNU/Linux, RedHat, y desde entonces empieza a hacerle frente incluso a VMWare

Y conjunto a esto, también han surgido las interfaces graficas para dar un apoyo definitivo e impulso al sistema KVM. También surgieron distribuciones específicas de Linux, como Proxmox, que implementan eficazmente, maquinas virtuales KVM y las dotan de complementos muy interesantes para facilitar la vida al gestor.

Hay que decir para terminar que KVM es de los pocos sistemas de virtualización que soporta el mecanismo PCI Passthrough (paso directo), esto ofrece, la opción de poder soportar tarjetas PCI “exóticas”, como tarjetas de sonido, o lo que aquí nos atañe, tarjetas del sistema Asterisk, de forma nativa, sin tener que manipular la información que circula entre ellas y la máquina. Con esto, a Asterisk se le ofrece la capacidad de obtener una fuente de sincronización pura, y podrían verse las funciones que hasta la fecha eran imposibles o demasiado complejas para verse implementadas directamente en maquinas virtuales.

Concluyendo, para este PFC, se ofrecerá una máquina virtual de ejemplo, conteniendo un sistema Asterisk y toda su funcionalidad, aprovechando todo lo descrito hasta aquí, y esta será creada a partir del sistema KVM.

2.4. Instalación de la Máquina Virtual

Como he comentado, vamos a utilizar el sistema KVM sobre nuestro Ubuntu instalado (podría instalarse en cualquier sistema de hecho, y aplicarse la MV para cualquier entorno dado). A continuación describo el proceso de configuración del mismo:

En el caso de haber elegido Ubuntu, como es el caso aquí descrito, existe la posibilidad de realizar la instalación completa de KVM a través de tasksel, con la opción Virtual Machine Host, que engloba todos los paquetes apt asociados a este sistema de virtualización.

Los paquetes principales son:

1. **QEMU-KVM:** Los sistemas de virtualización KVM basado en QEMU(forma parte del proceso de virtualización de KVM)
2. **Libvirt-bin:** Librerías encargadas de administrar KVM y QEMU
3. **Ubuntu-VM-Builder:** Herramienta de línea de comandos para administrar maquinas virtuales basadas en KVM
4. **Bridge-Utils:** Proporciona la conexión “puente” entre la red que proviene del sistema host y las maquinas virtuales. Este es el clásico método de compartir la conexión de forma autónoma y aislada para las maquinas virtuales, en el mundo de la virtualización.

El sistema de conexión puente, aun así no funciona por defecto en la instalación de KVM, y requiere de QEMU para operar, y para la instalación de nuestra maquina Asterisk será importante que tenga un acceso independiente. Para ello tenemos que instalar más paquetes:

1. **QEMU**: Sistema de virtualización
2. **Libcap2-bin**: Librerías de las herramientas de capacidades de Linux, setcap, sucap, etc...

Y con el siguiente comando adquirimos la propiedad CAP_NET_ADMIN necesaria para esta funcionalidad de acceso puente:

setcap cap_net_admin=ei /usr/bin/qemu-system-x86_64

Por otro lado, necesitamos configurar la interfaz de red para que funcione como posible puente para nuestras maquinas virtuales. Dentro del fichero de configuración, en caso de Debian /etc/network/interfaces hay que hacer algunas modificaciones para adaptar la interfaz (eth0 clásicamente) y crear el puente br0. A continuación un posible ejemplo:

```
auto eth0
iface eth0 inet manual
auto br0
iface br0 inet static
    address 192.168.1.200
    netmask 255.255.255.00
    gateway 192.168.1.1
    network 192.168.1.0
    broadcast 192.168.1.255
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
    bridge_maxwait 0
```

Realmente ya no es necesario mucho más que esto a nivel de configuración, a partir de aquí ya solo queda la creación de las máquinas. Existen múltiples métodos, y con el tiempo como comentaba antes, sobre las virtudes de KVM, se han desarrollado no solo interfaces graficas, sino sistemas para la línea de comandos que facilitan la creación enormemente. A nivel command-line puedo destacar dos principales: Virt-Manager para la creación de maquinas virtuales en general, y Ubuntu-VM-Builder, que nos interesa concretamente para lo que nos atañe con este proyecto.

Con Ubuntu-VM-Builder (vmbuilder) tenemos la posibilidad de generar una nueva maquina virtual, basada en una versión de Ubuntu a nuestra voluntad, y con una instalación totalmente desatendida. Es lo mas parecido a una maquina virtual con OpenVZ que nos podemos encontrar para KVM. Este “asistente” fue diseñado por Canonical específicamente para Ubuntu, he ahí su potencial. Para la creación con este sistema tenemos dos opciones:

1. Mediante la creación de un fichero de texto, que utilizaremos en conjunto al comando para que lo tome por defecto. Esta opción es muy interesante, si tenemos intención de diseñar maquinas nuevas en un futuro, y no queremos pasar por el tedioso estado de tener que releer toda la información específica acerca de esta aplicación. Es de alguna forma, un sistema recordatorio para que en el futuro veamos que parámetros elegimos para nuestras maquinas virtuales

2. Hay otra opción, mas rápida, sencilla y directa, y es aplicando el comando con todas sus opciones. Y es la que voy a aplicar para este caso.

Las características que voy buscando en la maquina a crear para este PFC son las siguientes:

1. Maquina basada en Ubuntu, versión 11.10 (Oneiric Ocelot)
2. Arquitectura AMD64
3. Dirección IP: 192.168.1.100
4. Puerta de Enlace: 192.168.1.1
5. Usuario: asterisk
6. Contraseña: asterisk
7. Memoria RAM: 2 Gb
8. Disco Duro: 4,5 Gb (un DVD completo en el que alojare la maquina virtual final)
9. La red ira a través de la interfaz puente que creamos anteriormente

Vamos al directorio donde queremos “instalar” la maquina virtual. Un ejemplo clásico seria el directorio especifico de libvirt: `/var/lib/libvirt/images/`, supongamos que creamos aquí un directorio llamado **pfc-asterisk** para contener nuestra maquina virtual.

Para definir el particionado del disco virtual a crear, dentro de este directorio vamos a crear un fichero llamado `vmbuilder.partition` con la siguiente información:

```
root 4608
```

Y por otro lado, para asegurarnos una conexión sencilla mediante acceso SSH, vamos a crear un fichero que se ejecutara en el primer arranque de nuestro sistema, en el mismo directorio también, un fichero llamado `boot.sh`

```
apt-get update
apt-get install -qqy --force-yes openssh-server
```

Entonces el comando quedaría así:

```
vmbuilder kvm ubuntu --suite=oneiric --flavour=virtual --arch=amd64
--mirror=http://de.archive.ubuntu.com/ubuntu -o --libvirt=qemu:///system
--ip=192.168.1.100 --gw=192.168.1.1 --part=vmbuilder.partition
--user=asterisk --name=Asterisk --pass=asterisk --addpkg=unattended-upgrades
--addpkg=acpid --addpkg=aptitude
--firstboot=/var/lib/libvirt/images/pfc-asterisk/boot.sh --mem=2048
--hostname=pfc-asterisk -bridge=br0
```

Con el comando **vmbuilder kvm ubuntu --help** tenemos acceso al significado de cada uno de los parámetros aquí utilizados aunque

Una vez terminado el proceso, que no debe tardar demasiado, ya solo queda arrancar el sistema, a través de la consola de maquinas virtuales, Virsh.

```
# virsh --connect qemu:///system
virsh # list --all
virsh # start pfc-asterisk
```

Desde este momento, y pasado el rato que termina el proceso de arranque de Ubuntu, podríamos acceder con cualquier cliente SSH a nuestra maquina virtual apuntando a la dirección que habíamos definido en las opciones del ejemplo: 192.168.1.100, y debería conectar sin ningún problema.

El fichero de configuración con toda la información con la que hemos diseñado esta maquina virtual (y que podríamos editar a voluntad), se encuentra ubicado en la ruta, /etc/libvirt/qemu/pfc-asterisk.xml

Este fichero es muy práctico, dado que es el fichero que nos facilita la portabilidad para mover nuestra máquina (junto evidentemente, al disco duro virtual ubicado en la anterior ruta) a otra maquina física, e instalando todo el sistema KVM, tenerla activa en solo unos minutos, con total abstracción de los drivers específicos de la nueva máquina. He aquí donde realmente radica el verdadero potencial de las maquinas virtuales.

Tenemos múltiples opciones para trabajar con las distintas aplicaciones de manejo y gestión de maquinas virtuales, y con la creación de las mismas concretamente basadas en KVM-QEMU, pero este resumen es en líneas generales, el procedimiento seguido por mi parte enfocando en el propósito específico de este proyecto.

```
asterisk@pfc-asterisk:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        4,3G  637M  3,4G  16% /
udev            1001M   4,0K 1001M   1% /dev
tmpfs            402M   188K   402M   1% /run
none             5,0M     0   5,0M   0% /run/lock
none            1005M     0 1005M   0% /run/shm
```

Figura 2.1: Capacidad inicial del disco duro en la Maquina Virtual

Capítulo 3

Caso de Uso: UCA Autos

Con todo el sistema, preparado para empezar a trabajar con Asterisk, a continuación presento un ejemplo como caso de uso para una empresa ficticia, la cual tiene un enorme interés en implementar un sistema Asterisk, dadas las características que este le aportara para su negocio.

3.1. Introducción

Don Zutano Doe, gerente, y propietario de un nuevo concesionario de automóviles llamado UCA Autos, basado en una prestigiosa marca a nivel nacional, es un gran aficionado a todos los adelantos tecnológicos y piensa que estos influirán en gran medida en el amplio desarrollo de su negocio. A través de un compañero de su club de tenis, que trabaja en una empresa de telefonía, le ha dado a conocer un nuevo sistema de comunicaciones que le podría aportar una serie de funcionalidades que le resultaron bastante interesantes para implementar en su nuevo negocio. Gracias a este contacto, tuvo la oportunidad de establecer relaciones con una modesta empresa local, con cierta experiencia en este ámbito.

En la primera reunión con el responsable de proyectos, el Sr. Doe pudo transmitir sus pretensiones iniciales. Dada que la estructura de la empresa aun estaba por ser determinada, de momento solo necesitaba funcionalidades básicas para cubrir los aspectos fundamentales de servicios primarios a nivel de comunicaciones para el negocio.

A priori la estructura departamental estaba definida, y había un listado de personal para poder definir una primera instantánea de cómo debería funcionar todo.

El responsable de proyecto pudo describirlo de la siguiente forma.

Existían 6 departamentos:

1. Departamento Administración
2. Departamento Comercial
3. Departamento de Almacén y Logística
4. Departamento de Postventa y Servicio
5. Departamento de Marketing y Calidad

Cada departamento tenía un responsable, y todos ellos dependían directamente, del Sr. Doe. Dentro de cada departamento existía un número variable de operarios, administrativos, comercial y personal en general que dependían de cada responsable.

Cada departamento debía tener un número de teléfono de contacto, y además, existía un teléfono general cara al publico. En total 6 números de teléfono directos.

La empresa todavía no era muy grande, y disponía de un número limitado de personal de los cuales los que realmente necesitaban acceso telefónico, se distribuían de la siguiente forma:

Departamento de Administración: Responsable + 3 administrativos

Departamento Comercial: Responsable + 4 comerciales

Departamento de Almacén: Responsable + 3 operarios

Departamento de Posventa: Responsable + 2 recepcionistas

Departamento de Marketing: Responsable + 2 operadores.

Más el Gerente

En total 20 usuarios.

De momento, el Sr. Doe no tenía demasiado clara una posible estructura jerárquica con restricciones de llamadas, y relaciones entre departamentos, así que la primera idea, era tener un sistema de telefonía simple, en el que todos pudieran llamar y recibir llamadas abiertamente, pero no descartaba en un futuro, implementar esas restricciones, e incluso tener algún sistema de control.

3.1.1. Instalación del sistema Asterisk

Dado que el nivel Hardware es algo excesivamente cambiante, voy a obviar las características técnicas del entorno y del servidor, haciendo referencia al apartado de elección de servidor para los aspectos determinantes a la hora de elegir una u otra preferencia.

La instalación del sistema Asterisk, se realizara sobre la versión actual mas estable, la versión 1.8, momento en que escribo esta información. He de determinar, que en la actualidad aun existen múltiples ramas dentro de la elección de versionado para el sistema Asterisk. Principalmente existen 5 fundamentales:

1. Rama extremadamente conservadora, y un Fork de Asterisk del sistema base llamado Asterisk-RSP (Real Solid Patchset), que se basa en la teoría de conservarse en un sistema Asterisk totalmente obsoleto a nivel de funcionalidades emergentes (siempre considerando que Asterisk es una solución de comunicaciones, y no te telefonía exclusivamente). Esta fundamentada en la versión 1.4, versión que decidieron era el momento de parar la vorágine de avances, y centrarse en la resolución de deficiencias de seguridad del sistema
2. Rama por necesidad o moderadamente conservadora, basada en la anterior versión de Asterisk, 1.6.2. Realmente en esta situación se encuentran instalaciones de Asterisk con desarrollos a medida que no pueden actualizarse porque el sistema se volvería inestable. Realmente todo sistema Asterisk puro, es recomendable actualizarlo dado que la versión 1.6.2 tiene un soporte limitado.
3. Basada el progreso de Asterisk, en este caso todas las instalaciones basadas en la versión 1.8 la cual trae bastantes mejoras, y simplificaciones a nivel de configuración. Se considera por Digium, la versión estable del sistema

4. Basada en el progreso extremado, instalaciones basadas en Asterisk 1.10 (o como le llaman ahora, Asterisk 10). Realmente son versiones de prueba, y jamás recomendadas para instalaciones en entornos de producción como viene siendo habitual en la mayoría de los entornos de administración de sistemas.
5. Liberada recientemente a la comunidad, una rama similar a Asterisk-RSP pero mantenida por Digium, y basada en las versiones mas recientes del sistema que se encuentran en fase estable. Podría considerarse una versión de Soporte a Largo Plazo, y es denominada Asterisk-Certified. En estos momentos esta empezando a imponerse entre la mayoría de los usuarios profesionales que venían siguiendo por sistema la opción número 3.

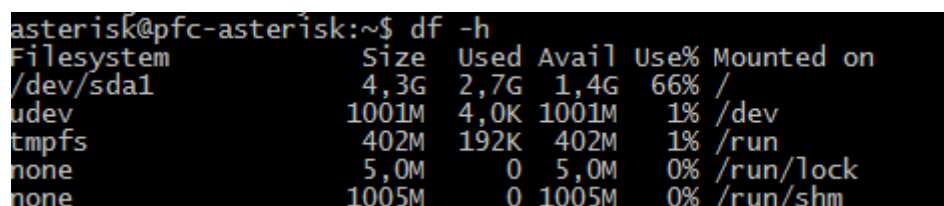
Para nuestro cliente, en este caso, vamos a seguir la opción número 5, ya que se trata de un servidor de producción y queremos ofrecerle un nivel 2 de servicio según el acuerdo en la capa de servicio estandarizado ¹

Para el momento, la versión Certificada mas reciente es la versión 1.8.11-cert1 asi que descargaremos las fuentes de la siguiente URL:

<http://downloads.asterisk.org/pub/telephony/certified-asterisk/releases/certified-asterisk-1.8.11-cert1.tar.gz>

Y seguiremos el método de instalación descrito en el artículo Wiki asociado correspondiente.

Una vez con todo el sistema Asterisk desplegado, nos detenemos a comprobar el consumo de recursos en ese momento determinado para comprobar que podremos realizar el resto del despliegue sin mayor inconveniencia.



Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	4,3G	2,7G	1,4G	66%	/
udev	100M	4,0K	100M	1%	/dev
tmpfs	402M	192K	402M	1%	/run
none	5,0M	0	5,0M	0%	/run/lock
none	1005M	0	1005M	0%	/run/shm

Figura 3.1: Capacidad del disco duro tras la instalación completa de Asterisk

Todavía tenemos suficiente espacio para añadir componentes de menor magnitud. Existen dos componentes adicionales que cumplirán funciones locales específicas y se relacionarán con nuestra máquina Asterisk prácticamente desde los inicios de su configuración.

Se trata del paquete LAMP (Linux + Apache como servidor Web + MySQL como SGBD de nuestra BD + PHP como lenguaje de programación en entorno web que se conjugará con nuestro servidor web para poder ofrecer aplicaciones combinables con Asterisk como veremos mas adelante) y por otro lado algún servidor de Correo como Postfix. Para ello nos valemos de la herramienta **tasksel** de Ubuntu que realiza la instalación completa de la forma más ágil para nuestro propósito. Con estas instalaciones apenas consumimos poco más de 100MB así que seguimos teniendo suficiente espacio para continuar el proceso tranquilamente.

¹Wikipedia, Soporte Técnico. http://es.wikipedia.org/wiki/Soporte_t%C3%A9cnico

3.1.2. Configurando equipos y el entorno SIP

Ahora daríamos paso a la configuración en primera instancia de los requerimientos solicitados por el Sr. Doe en función de la estructura departamental.

Vamos a considerar, que todavía el presupuesto es demasiado ajustado para la empresa del Sr. Doe como para andar comprando teléfonos que operen con el protocolo SIP (conocidos popularmente como Teléfonos VoIP). Pero hemos comprado una remesa de Auriculares con micrófono integrado para cada una de las extensiones a configurar, que serán adaptados al cada PC de cada usuario y funcionales, utilizando un software capaz de transmitir audio e inicializar sesión en nuestra máquina Asterisk utilizando el protocolo SIP.

Creemos en primer lugar el fichero de configuración dentro de `/etc/asterisk/` llamado `sip.conf` donde añadiremos la configuración específica de cada una de las 20 extensiones solicitadas por el gerente:

`/etc/asterisk/sip.conf`

```
[general]
; Nuestro idioma
language = es
; Cambiamos el puerto para ofrecer seguridad por ocultacion
bindport = 35060
allowguest = no

[telefonos](!)
; Vamos a impedir el Transcoding utilizando solo el codec Alaw
disallow = all
allow = alaw
dtmfmode = rfc2833
host = dynamic
qualify = yes
type = friend

[gerencia](!)
context = gerencia

[manager](!)
context = manager

[resto](!)
context = extensiones

; 1 - Administracion
; 2 - Comercial
; 3 - Almacen
; 4 - Posventa
; 5 - Marketing

; Gerente
[10](telefonos,gerencia)
md5secret=2b45660a7b1155943f0132f05bd0e34d
mailbox = 0@admin

; Responsable Administracion
```



```
[11] (telefonos,manager)
md5secret=6acb3eb4a4939b22ea6709e87ec79311
mailbox = 1@admin

[12] (telefonos,resto)
md5secret=bb24a2de8a36fdd14e358493cd5abac5
mailbox = 2@admin

[13] (telefonos,resto)
md5secret=501b420ea96938f8ba34ecce52a4c654
mailbox = 3@admin

[14] (telefonos,resto)
md5secret=6745f8a4a34c38a2f6b554334bb76483
mailbox = 4@admin

; Responsable Comercial
[21] (telefonos,manager)
md5secret=5dbe664c6d998967ad9bfe4ea37521a1
mailbox = 1@ventas

[22] (telefonos,resto)
md5secret=2f9ddd39a552c3c7709f07b288cf1db3
mailbox = 2@ventas

[23] (telefonos,resto)
md5secret=0a4bb4d4b4f92ba79f48b28a706bbe7b
mailbox = 3@ventas

[24] (telefonos,resto)
md5secret=6736a7be4a1ef7a6f111441d5d59187e
mailbox = 4@ventas

[25] (telefonos,resto)
md5secret=10f36f1ed829ec2a8fac56d0b9c3080d
mailbox = 5@ventas

[31] (telefonos,manager)
md5secret=400e3d6f04bc6e13bb9a58f0f53c1a26
mailbox = 1@almacen

[32] (telefonos,resto)
md5secret=caff9208a95aa83357ae5e5d3949a681
mailbox = 2@almacen

[33] (telefonos,resto)
md5secret=58a1e10c20da84d4c323f5018b1f7f26
mailbox = 3@almacen

[34] (telefonos,resto)
md5secret=902dc9faff73e90187292a036a8c1d86
mailbox = 4@almacen

[41] (telefonos,manager)
md5secret=21c85ea8cc8b45b430742f1cf9eb55bb
mailbox = 1@posventa
```

```
[42] (telefonos, resto)
md5secret=d7e2c8ec48e1647412503d82659dd9d2
mailbox = 2@posventa

[43] (telefonos, resto)
md5secret=988cd566697d6310e7a26f21ecc7007f
mailbox = 3@posventa

[51] (telefonos, manager)
md5secret=2bbcc664bdf4e1bb29cd7f1bdbf91327
mailbox = 1@market

[52] (telefonos, resto)
md5secret=0c32d549b9c60fa6a7df077e4bb545e5
mailbox = 2@market

[53] (telefonos, resto)
md5secret=7b099d95a97bb4c8d6a1d08c7c047167
mailbox = 3@market
```

Para la creación de este fichero, hemos seguido las reglas básicas de configuración para el protocolo SIP como podemos ver en el correspondiente artículo de la Wiki.² Significativamente podemos observar que la estructura que hemos seguido para denominar las extensiones, aun no siendo demasiado recomendada, es la siguiente:

Las extensiones constarán de dos dígitos. El primero se encarga de definir el departamento, y el segundo, el puesto. La extensión 10, como extensión especial para definir concretamente la del Sr. Doe, el resto, si el número para designar el puesto es el número 1, correspondería al Responsable de Sección, y los sucesivos dígitos, corresponden al resto de los empleados. Además hemos creado cuatro plantillas para un cometido concreto:

La primera, teléfonos, es una plantilla que engloba todos los parámetros comunes a las extensiones de la centralita. En este caso, nos interesa trabajar en exclusiva con el códec Alaw (G.711 EU), dado que tenemos una conexión bastante buena, pero nuestra máquina no es demasiado potente, además no queremos invertir en códecs licenciados.

Las tres siguientes plantillas, hacen referencia al "nivel" de los usuarios, y en función de este, los mandaremos a un contexto u otro del Dialplan que construiremos a continuación. A priori, el señor Zutano no nos hizo referencia concreta, sobre la posibilidad de que unos usuarios pudiera hacer cosas que otros no, pero anticipándonos por nuestra experiencia, empezamos a "habilitar" esta funcionalidad.

El sistema de contraseñas, teniendo a lo más básico, aun siendo una mala práctica, hemos utilizado la contraseña por defecto 1234 para todas las extensiones, pero al menos encubierto bajo contraseñas de algo más de seguridad con un patrón MD5, para mostrar algo más de compromiso.

Finalmente y volviéndonos a adelantar a una probable futura necesidad, hemos asociado a cada extensión su correspondiente buzón de voz, que configuraremos a voluntad si surge el caso más adelante.

Ahora pasamos a comprobar que todo ha ido bien. Accedemos a la CLI de Asterisk y con el comando:

²SIP. <http://wikiasterisk.com/index.php?title=SIP>

```
CLI> sip reload
```

Vemos como nuestra configuración carga adecuadamente. Ahora vamos a comprobar que los pares han sido configurados correctamente. Observando el despliegue de sistemas de la empresa UCA Autos, comprobamos que todos los puestos trabajan en un sistema operativo de Microsoft. En este caso decidimos que vamos a utilizar un teléfono por software, dado que el Sr. Doe de momento no se ha decidido en el modelo de teléfono físico que quiere comprar. El Softphone que hemos elegido por su reconocida estabilidad y calidad es uno gratuito, pero no libre, llamado X-Lite. Tenemos constancia de múltiples Softphone Open Source, pero bajo este sistema operativo, no tienen demasiada estabilidad y pueden complicarnos la vida.

Configurando la extensión del Sr.Doe de ejemplo como podemos ver en la siguiente Figura, captura de pantalla de X-Lite.

Comprobamos que se registra correctamente:

```
-- Registered SIP '10' at 192.168.1.200:16992
CLI> sip show peers
Name/username Host Dyn Forcerport ACL Port Status
10/10 192.168.1.200 D N 16992 OK (3 ms)
```

Así que ya podemos proceder a registrar las 20 extensiones, en los 20 equipos de nuestros usuarios y ya estarían listos para empezar a operar con nuestra primera instancia de sistema Asterisk.

3.1.3. Configuración del Primer Plan de Marcación

Ahora los teléfonos no pueden hacer casi nada, excepto llamar a una extensión si saber la ruta exacta, y sin pasar por nuestro sistema Asterisk. Nos interesa que por lo menos puedan realizar llamadas entre ellos marcando el número de extensión según la estructura que definimos anteriormente. Para ello vamos a crear el fichero de configuración del Dial Plan según veremos a continuación:

/etc/asterisk/extensions.conf

```
; --- Plan General de Extensiones

[extensiones]

exten => 10,1,NoOp()
same => n,Dial(SIP/10)

exten => 11,1,NoOp()
same => n,Dial(SIP/11)

exten => 12,1,NoOp()
same => n,Dial(SIP/12)

; --- Contexto especifico de los Managers

[manager]

include => extensiones

; --- Contexto especifico del Gerente
```

SIP Account

Account | Voicemail | Topology | Presence | Transport | Advanced

Account name:

Protocol:

Allow this account for

☒ Call

☒ IM / Presence

User Details

* User ID:

* Domain:

Password:

Display name:

Authorization name:

Domain Proxy

☒ Register with domain and receive calls

Send outbound via:

☒ Domain

☐ Proxy Address:

Dial plan:

Figura 3.2: Configuración de una Extensión con el Programa X-Lite v4.1

```
[gerencia]

include => manager

; --- Otros Sistemas Adicionales de Prueba, solo puede acceder el Gerente.

exten => 1111,1,NoOp()
same => n,Progress()
same => n,MusicOnHold()

exten => 2222,1,NoOp()
```

```
same => n, Answer()  
same => n, Set (CHANNEL(tonezone)=starwars)  
same => n, Dial (Local/10,tT,30)
```

Es importante considerar, que no hemos creado extensiones de marcación dentro del plan para cada uno de los dispositivos SIP ya que como veremos adelante, este proceso se simplifica bastante utilizando Macros, como un sistema algo más avanzado del Dialplan de Asterisk.

Para recargar el Plan de Marcación y poder hacer uso de este desde la CLI de Asterisk:

```
CLI> dialplan reload
```

Ahora utilizando nuestro Softphone X-Lite podríamos marcar la extensión 11 y el teléfono empezaría a sonar. Al descolgar se establecería la conexión. En este fichero vemos algunas cosas peculiares:

La aplicación NoOp sirve para hacer debug en consola. Si pusiéramos mensajes dentro del paréntesis, se verían en el log del CLI. Se pueden poner incluso variables del sistema, la cuestión es que esto viene más explicando en el apartado correspondiente de la documentación.

Lo verdaderamente significativo es la estructura. Como podemos ver, existen tres contextos (se encuentran dentro de corchetes). Como recordaremos, en el fichero sip.conf anterior, estos contextos hacen referencia a los contextos a donde las marcaciones que realizan los dispositivos irían. En este caso, podemos ver como el contexto [gerente] tiene un parámetro **include** que sirve para incluir "todas las extensiones de los contextos a los que apunta el mismo. En este caso, incluiría por completo el contexto [manager], que este a su vez incluye el contexto [extensiones]. Esto significa que el gerente tiene "visibilidad" total sobre todo el plan de marcación del sistema, en cambio un dispositivo de un empleado normal, solo podría acceder al plan de marcación específico dentro de su contexto, pero no del contexto del gerente.

Cuando se lo expliquemos al gerente, seguro que se le ocurren aplicaciones prácticas como el hecho de evitar que sus empleados llamen a números internacionales, o de tarificación especial.

3.1.4. Introduciendo al proveedor de Telefonía

El último requisito que puso el gerente, es la posibilidad de tener 6 números de marcación directa, uno para cada departamento, y otro para él. Para poder conseguir esto, necesitamos contactar con un proveedor que nos ofrezca esto.

Como siempre, tenemos múltiples alternativas, pero si queremos evitar tener que comprar dispositivos, la más obvia es contactar con proveedores de Telefonía IP. Así que nos pusimos en contacto con una empresa muy conocida de la provincia, llamada **UCA Telecom**. Comparando los planes de precios que nos ofrecía "FonicaTele", el proveedor más conocido a nivel nacional, teníamos una oferta mucho más competitiva y encima, no teníamos la necesidad de andar contratando planes mensuales, ni comprando esos dispositivos que comentaba antes, para interconectar con la Red de Telefonía Conmutada clásica.

/etc/asterisk/sip.conf

```
[general]  
register => ucaautos:12345678@home-asterisk.local:35060  
  
[proveedorip]
```

```
defaultuser = ucaautos
fromuser = ucaautos
secret = 12345678
host = home-asterisk.local
port = 35060
type = friend
insecure = invite
qualify = yes
context = entrantes
```

Un dato muy relevante en la configuración de nuestro proveedor, es el hecho que el tipo es **friend** y no peer. Si nuestra intención fuera emitir llamadas a través de mismo, lo correcto sería esto. Pero como además, vamos a recibir llamadas entrantes dado que nos va a ofrecer un servicio de números Direct Dial-In (DDI), lo correcto es considerarlo, bidireccional con este tipo en concreto.

Pero de hecho, hay un parámetro que puede resultar aun más curioso de observar, que es el **insecure=invite**. Tenemos un problema si trabajamos con un dispositivo tipo friend: la autenticación bidireccional. Observamos que en la primera línea hay un mensaje register, para registrarnos en el proveedor IP. El tema es que el proveedor IP supuestamente, ha de autenticarse luego con nosotros cuando intenta enviarnos la llamada directa al DDI, pero nosotros al solo tener un dispositivo configurado, no podemos ofrecerle unas credenciales específicas (que probablemente sean las mismas que las de llamadas salientes aunque no tiene porque ser así). Al habernos autenticado primero con el mensaje register, al poner el parámetro insecure de esta forma, como la conexión ya está hecha (el primer mensaje REGISTER ya se envió), los sucesivos INVITE, no tendrían la necesidad de volver a autenticarse. Si quitamos el insecure, y activamos el debug de SIP, veremos como hay un error en la autenticación entrante, cuando tratamos de llamar a un número DDI.

Lo correcto en este caso, sería convertir este par friend, en un par tipo peer. Y por otro lado, crear un par tipo user, con el usuario de autenticación de entrada del proveedor, en caso que sea otro diferente (como es este caso). Con esto, ya no haría falta tener el parámetro insecure=invite activado, dado que todos los mensajes INVITE se autenticarían con el nuevo dispositivo tipo user, correctamente. Como este caso es relativamente raro (que los parámetros de autenticación de entrada y de salida sean diferentes para nuestro proveedor IP), entonces voy a dejarlo así, dando un voto de confianza a nuestro operador que a final de cuentas, es quien nos va a facturar a final de mes.

Introduciendo los siguientes cambios:

/etc/asterisk/sip.conf

```
[proveedorip]
defaultuser = ucaautos
fromuser = ucaautos
secret = 12345678
host = home-asterisk.local
port = 35060
+type = peer
-type = friend
-insecure = invite
qualify = yes
-context = entrantes
```

```
+[ucatelecom]
+secret = 12345678
+host = home-asterisk.local
+port = 35060
+type = user
+context = entrantes
```

Por otro lado, hemos creado un contexto nuevo en el Dialplan, llamado [entrantes] haciendo referencia al lugar donde apunta nuestro nuevo proveedor SIP. Ahí vamos a definir todos los números DDI que nos ha solicitado el gerente, y reenviarlos con la función GoTo al contexto, extensión y prioridad correspondientes de cada departamento.

/etc/asterisk/extensions.conf

```
[entrantes]

exten => 956001101,1,Goto(extensiones,10,1)
exten => 956001111,1,Goto(extensiones,11,1)
exten => 956001121,1,Goto(extensiones,21,1)
exten => 956001131,1,Goto(extensiones,31,1)
exten => 956001141,1,Goto(extensiones,41,1)
exten => 956001151,1,Goto(extensiones,51,1)
```

Y con esto, ya estarían satisfechas las necesidades del gerente, por lo que concertamos una nueva cita con el Sr. Zutano para mostrarle que todo esta correctamente funcionando.

Bibliografía

- [1] Manual de Asterisk y otras hierbas. Creación de ficheros \LaTeX con GNU Emacs. <https://forja.rediris.es/docman/view.php/555/854/Intro-asterisk-uca.pdf>.
- [2] Digium Inc. Pagina oficial de Asterisk. <http://www.asterisk.org>.
- [3] Jim Van Meggelen y Russell Bryant Leif Madsen. *Asterisk The Definitive Guide*. O'Reilly Media, 2011.
- [4] VOIP-Info LCC. Wiki sobre VoIP en Inglés. <http://www.voip-info.org>.

Software utilizado

Es usual en un PFC referenciar que software has usado para la realización del mismo. Aprovecharé este apartado para que conozcas alguna herramienta que puede serte de ayuda para realizar tus documentos en \LaTeX

Emacs + AucTeX

Emacs es uno de los programas de edición más usados por desarrolladores de software, ya que es bastante versátil admitiendo gran cantidad de “plugins” o extensiones que permiten ampliar aun más sus funcionalidades.

Uno de estos plugins es AucTeX[1], el cual incluye rutas para ciertos comandos, resaltado de sintaxis, previsualización del documento, menú matemático en el cual podemos acceder e insertar la gran mayoría de los símbolos matemáticos, para no tener que memorizarlos. Podemos ver un ejemplo de Emacs + AucTeX en la figura 3.3

Por ejemplo, para cerrar un entorno `\begin()`, con su respectivo `\end()`, utilizaremos el atajo `C-c M-]`, para añadir un `\item`, tenemos el atajo `C-c C-j`, y así unos cuantos, que una vez que nos habituamos a ellos, son bastante cómodos.

Además, es bastante configurable, con indentado automático, corrector ortográfico y demás. El fichero adjunto a este documento, *conf_emacs* incluye una configuración con varias de estas opciones.

Doxygen

Realmente, *Doxygen* [2] no es una herramienta que vayamos a utilizar para realizar documentos \LaTeX directamente. Sin embargo, para la documentación de código si es bastante útil.

Esta herramienta realiza una documentación automática de código fuente. Es decir, para nuestro PFC, podemos utilizar para generar la documentación de las APIs de nuestras librerías y demás. Puede generar esta documentación en varios formatos, y entre ellos, \LaTeX , de forma que podemos utilizar ese código generado en nuestra memoria de forma automática.

GNU Make

GNU Make es el programa de recompilación y de control de dependencias por excelencia. Se puede utilizar para compilar proyectos software en diversos códigos, o como en el caso de este documento, para compilar documentos \LaTeX con diversas opciones.

Para más información [4]

Dia

Dia es un editor de gráficos vectoriales el cual incluye distintas plantillas para distintos tipos de gráficos, como pueden ser UML, ERe, diagramas de flujo, esquemas Cisco de red y un larguísimo etcétera. Podemos ver el interfaz en la figura [3.4](#)

Estos diagramas podemos exportarlos a diversos formatos de imagen (`.png`, `.eps`, ...) o a formato `.tex`, como vimos anteriormente [\[3\]](#)

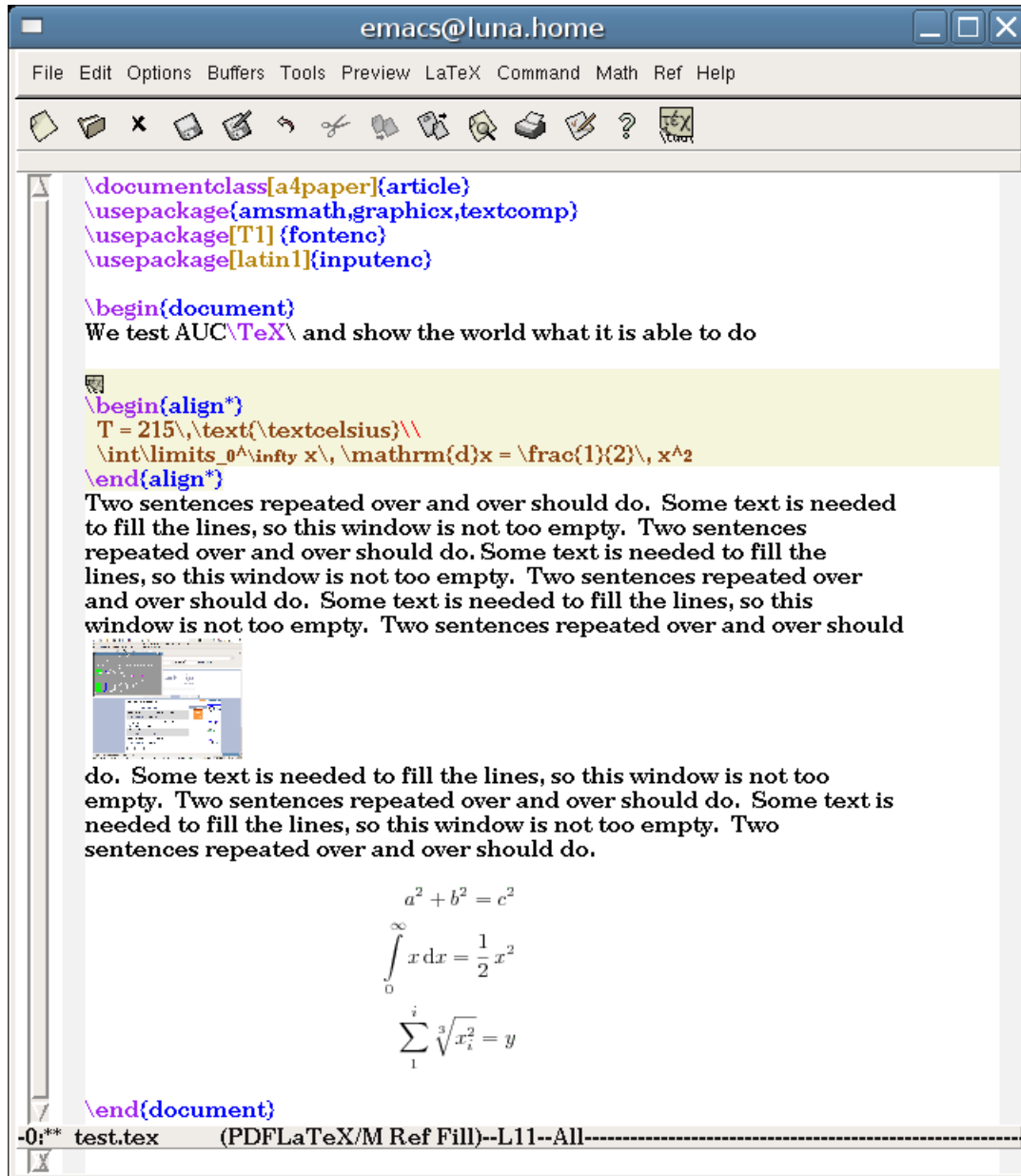


Figura 3.3: Emacs + Auc \TeX

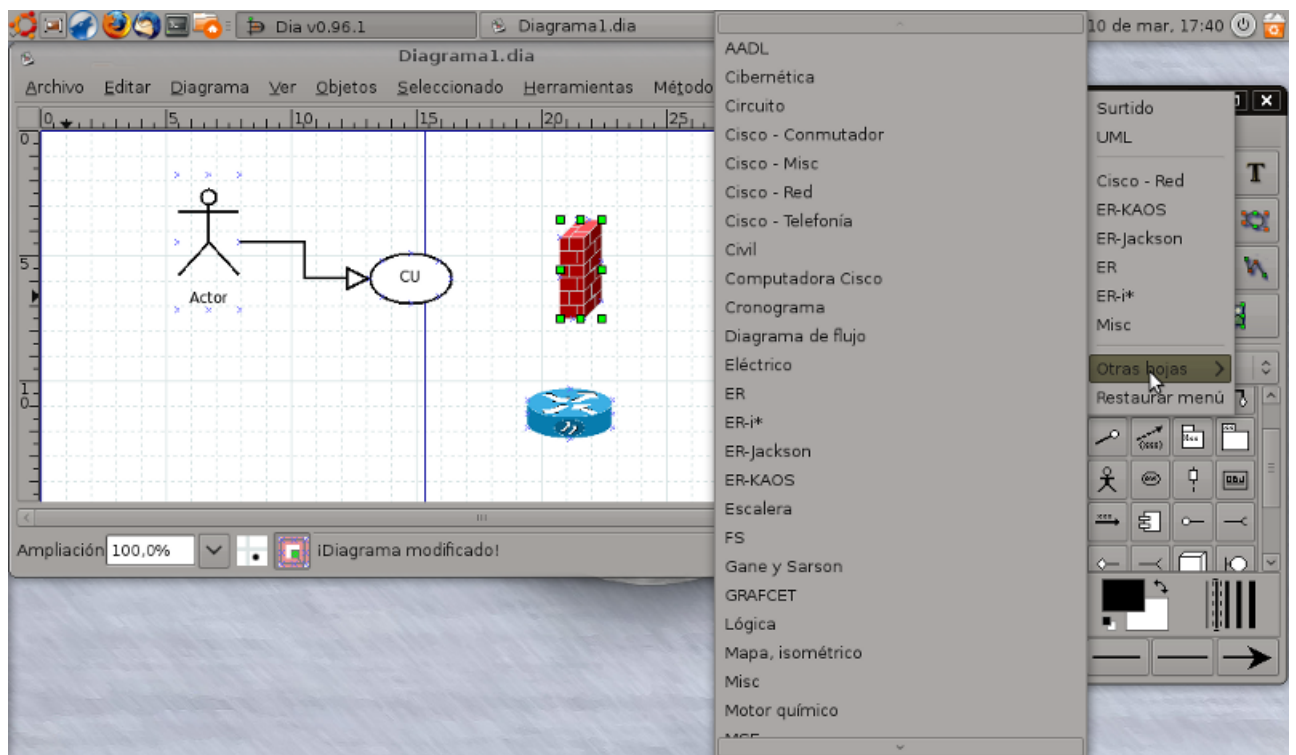


Figura 3.4: Interfaz de Dia

Instalación de L^AT_EX

Veamos que tenemos que hacer para instalar L^AT_EX con todas sus capacidades en un sistema basado en Debian, como Ubuntu. Primero hay que tener en cuenta que L^AT_EX es relativamente pesado con respecto a otros compiladores.

Nosotros vamos a utilizar la distribución de L^AT_EX incluida en los repositorios de Ubuntu llamada *texlive*. Si la buscas en tu gestor de paquetes, encontrarás infinidad de paquetes aparte del principal. Existen otras distribuciones como Te_X

Si instalas solo los básicos, es decir instalas *texlive* y los programas necesarios para él, no podrás compilar este documento, ya que faltarían paquetes tales como *supertabular* y varios. Por eso, si no tienes problema de espacio en el disco duro te recomiendo que instales el paquete *texlive-full*, que instala **todos** los paquetes de *texlive*, incluyendo documentación en todos los idiomas disponibles. Si buscas no tener problemas de dependencias, este es tu método.

```
sudo apt-get install texlive-full
```

En caso de querer ser un poco más concreto, en principio puedes trabajar con la más básica (*texlive* y sus dependencias) y en función de los paquetes que te vayan faltando, los instalas.

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”). To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.